z/VM and Linux on IBM System z
The Virtualization Cookbook

A cookbook for installing and customizing z/VM 5.2
and Linux SLES 10 on the mainframe

Marian Gasparovic, Michael MacIsaac, Carlos Ordonez, Jin Xiong
Contents

Preface ................................................................................................................. ix
Summary of changes to August 2006 version ........................................................ ix
Summary of changes to February 2007 version .................................................... ix
Conventions ....................................................................................................... x
The team that wrote this trilogy ....................................................................... xi
Comments welcome .......................................................................................... xi

Notices ............................................................................................................... xii
Trademarks ........................................................................................................ xiii

Part 1. z/VM ...................................................................................................... 1

Chapter 1. Introduction to z/VM and Linux ...................................................... 3
  1.1 What is virtualization? ............................................................................... 4
  1.2 A philosophy adopted in this book ............................................................ 4
  1.3 Choices and decisions made in this book ................................................ 4
  1.4 IBM Director and z/VM Center Extension ............................................. 5
  1.5 Infrastructure design ............................................................................. 5
  1.6 Usability tests performed for this book .................................................. 6
  1.7 The chapters in this book ..................................................................... 7

Chapter 2. Planning .......................................................................................... 9
  2.1 Bill of materials ....................................................................................... 9
  2.1.1 Hardware resources .......................................................................... 9
  2.1.2 Software resources ......................................................................... 10
  2.1.3 Networking resources .................................................................... 10
  2.2 z/VM conventions ............................................................................... 10
  2.2.1 Volume labeling convention ............................................................ 11
  2.2.2 Backup file naming convention ....................................................... 11
  2.2.3 The command retrieve convention ................................................ 12
  2.3 Disk planning ....................................................................................... 12
  2.4 Memory planning .............................................................................. 13
  2.5 Password planning ............................................................................... 13
  2.6 Planning worksheets ........................................................................... 14
  2.6.1 z/VM resources used in this book .................................................. 14
  2.6.2 z/VM DASD used in this book ......................................................... 15
  2.6.3 Linux resources used in this book ................................................... 16
  2.6.4 Linux user IDs used in this book ..................................................... 16
  2.7 Blank worksheets ............................................................................... 17
  2.7.1 z/VM resources worksheet ............................................................... 17
  2.7.2 z/VM DASD worksheet ................................................................. 18
  2.7.3 Linux resources worksheet ............................................................ 19
  2.7.4 Linux user ID worksheet ............................................................... 19

Chapter 3. Configuring a desktop machine .................................................... 21
  3.1 PuTTY: A free SSH client for Windows ............................................... 21
  3.2 Setting up a VNC client ..................................................................... 25
  3.2.1 Downloading and running RealVNC ......................................... 25
  3.2.2 Customizing RealVNC ................................................................. 26
Chapter 4. Installing and configuring z/VM

4.1 Installing z/VM from DVD
  4.1.1 Booting z/VM from DVD
  4.1.2 Copying a vanilla z/VM system to DASD
  4.1.3 IPL the vanilla z/VM from DASD
  4.1.4 Completing the z/VM installation
4.2 Configuring TCP/IP
  4.2.1 Use the IPWIZARD tool
  4.2.2 Use the IPWIZARD tool
4.3 Configuring the XEDIT profile
4.4 Customizing the SYSTEM CONFIG file
4.5 Configuring TCP/IP to start at IPL time
  4.5.1 Renaming the TCPIP configuration file
  4.5.2 Copy the PROFILE XEDIT file
  4.5.3 Configuring the FTP server
  4.5.4 Shuting down and reIPLing the system
4.6 Adding paging volumes
  4.6.1 Formatting the paging volumes
  4.6.2 Formatting DASD for minidisks
  4.6.3 Updating the SYSTEM CONFIG file
  4.6.4 Testing the changes
4.7 Creating a user ID for common files
  4.7.1 Define the user in the USER DIRECT file
  4.7.2 Logging and customizing the new user ID
  4.7.3 Copying a PROFILE XEDIT
  4.7.4 Creating a PROFILE EXEC
  4.7.5 Copying files associated with this book to LNXMAINT
4.8 Customizing system startup and shutdown
  4.8.1 Configuring the AUTOLOG1 PROFILE EXEC
  4.8.2 Testing the changes
4.9 Addressing z/VM security issues
  4.9.1 Changing passwords in USER DIRECT
4.10 Backing up your z/VM system to tape
4.11 Relabeling the system volumes
  4.11.1 Modifying labels in the SYSTEM CONFIG file
  4.11.2 Modifying labels in the USER DIRECT file
  4.11.3 Changing the labels on the five volumes
  4.11.4 Shuting down your system and restarting it
4.12 Restoring your z/VM system from tape

Chapter 5. Servicing z/VM

5.1 Applying a PTF
  5.1.1 Determining if a PTF has been applied
  5.1.2 Getting service via Internet FTP
  5.1.3 Downloading the service to z/VM
  5.1.4 Receiving, applying, and building service
  5.1.5 Putting the service into production
5.2 Applying a Recommended Service Upgrade or RSU
  5.2.1 Getting service from the Internet
  5.2.2 Downloading the service to z/VM
  5.2.3 Receiving, applying, and building the service
  5.2.4 Putting the service into production
# Contents

5.3 Determining z/VM's service level ........................................ 80

Part 2. SLES 10 ........................................................................... 81

Chapter 6. Configuring an NFS server for SLES 10. .......................... 83
6.1 Downloading files associated with this book .............................. 83
6.2 Setting up a SLES 10 install server ........................................ 84
   6.2.1 SLES 10 ISO image file names and sizes ............................... 84
   6.2.2 SLES 10 Service Pack CDs ............................................... 84
   6.2.3 Starting from physical CDs .............................................. 84
   6.2.4 Verifying the ISO images ................................................ 85
   6.2.5 Configuring the SLES 10 install server ................................. 85
6.3 Enabling the NFS server ..................................................... 86

Chapter 7. Installing and configuring SLES 10 ................................. 89
7.1 Creating the user ID SLES10 ................................................. 89
7.2 Adding SLES10 to AUTOLOG1’s PROFILE EXEC ......................... 92
7.3 Preparing SLES10 bootstrap files ........................................... 92
7.4 Installing the master image .................................................. 94
   7.4.1 Begin the SLES 10 installation ......................................... 96
   7.4.2 Beginning YaST installation ........................................... 98
   7.4.3 Continuing the YaST2 installation .................................... 100
   7.4.4 Booting your new Linux system from disk ......................... 101
   7.4.5 Completing YaST2 installation ........................................ 102
7.5 Configuring the master image .............................................. 104
   7.5.1 Create nightly.sh script ............................................... 104
   7.5.2 Adding additional RPMs .............................................. 104
   7.5.3 Configuring the VNC server .......................................... 105
   7.5.4 Preparing for Online Update ........................................ 107
   7.5.5 Removing unnecessary RPMs ...................................... 107
   7.5.6 Turning off unneeded services ..................................... 108
   7.5.7 Configuring rsyncd .................................................... 109
   7.5.8 Applying service - online update .................................. 109
   7.5.9 Configuring sitar ..................................................... 111
   7.5.10 Setting the software clock accurately ............................ 111
   7.5.11 Setting system to halt on SIGNAL SHUTDOWN ............... 112
   7.5.12 Turning off the hz_ .................................................. 112
   7.5.13 Modifying zipl.conf ............................................... 113
   7.5.14 Rebooting the system and verifying changes .................. 114
7.6 Installing the controller ..................................................... 114
   7.6.1 Verifying the installation ........................................... 120
7.7 Configuring the controller ................................................ 121
   7.7.1 Copying files to the controller ..................................... 121
   7.7.2 Adding additional RPMs ............................................ 122
   7.7.3 Configuring the VNC server ........................................ 122
   7.7.4 Preparing for Online Update ....................................... 122
   7.7.5 Removing unnecessary RPMs .................................... 122
   7.7.6 Turning off unneeded services ................................... 122
   7.7.7 Applying service if necessary - online update .................. 122
   7.7.8 Configuring sitar ................................................... 123
   7.7.9 Installing the cmsfs package ....................................... 123
   7.7.10 Turning on the NFS server ....................................... 124
   7.7.11 Cleaning temporary files .......................................... 124
   7.7.12 Prepare special files in master ................................. 124

7.7.13 Cleaning temporary files .......................................... 124
14.3 Disaster recovery of z/VM and virtual Linux servers ........................................... 228

Appendix A. References ...................................................................................................... 229
A.1 Related books ............................................................................................................... 229
A.2 Online resources ......................................................................................................... 230
A.3 Important z/VM files .................................................................................................. 230
A.4 Cheat sheets ............................................................................................................... 230
  A.4.1 XEDIT cheat sheet ................................................................................................. 231
  A.4.2 vi cheat sheet ........................................................................................................ 231

Appendix B. Common source code ..................................................................................... 233
B.1 z/VM REXX EXECs and XEDIT macros ...................................................................... 233
  B.1.1 The CPFORMAT EXEC ......................................................................................... 233
  B.1.2 The CHPW52 XEDIT macro .................................................................................. 237
  B.1.3 The LABEL520 EXEC ............................................................................................ 240
  B.1.4 The LABEL520 XEDIT macro .............................................................................. 242
B.2 Linux source code ....................................................................................................... 243
  B.2.1 Bash scripts on the controller ............................................................................... 243

Appendix C. Source code specific to Linux ....................................................................... 249
C.1 Obtaining and using the Web material ........................................................................ 249
C.2 The clone.sh script ..................................................................................................... 249

Index .................................................................................................................................. 261
Preface

“\textit{The search for truth is more precious than its possession.}”

— Albert Einstein

This book describes how to \textit{roll your own} Linux virtual servers on IBM System z hardware under z/VM. It adopts a cookbook format that provides a concise, repeatable set of procedures for installing and configuring z/VM in an LPAR and then installing and customizing Linux. You need an IBM System z logical partition (LPAR) with associated resources, z/VM 5.2 media, and a Linux distribution.

In February of 2007, two books were published, each book targeting a different distribution:

- \textit{z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES 10} (no upgrade path from SLES 9 to SLES 10 is described)
- \textit{z/VM and Linux on IBM System z: The Virtualization Cookbook for RHEL 5}

In August of 2006 a book was published that addresses both 31-bit and 64-bit SLES 9:

- \textit{z/VM and Linux on IBM System z: The Virtualization Cookbook 2}

In addition to these books, there are associated REXX EXECs and Linux scripts to help speed up the installation and configuration of both z/VM and Linux. These tools are not IBM products nor formally supported. However, they will be informally supported. They are available on the Web.

This book assumes that you have a general familiarity with System z technology and terminology. It does not assume an in-depth understanding of z/VM and Linux. It is written for those who want to get a quick start with z/VM and Linux on the mainframe.

Summary of changes to August 2006 version

The second release of the cookbook in August of 2006 was based on the IBM redbook \textit{z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES9}, SG24-6695-01. The content that was added to that book is as follows:

- Manual cloning: A new section in chapter 9, Cloning a virtual server manually was added so you can better understand the steps involved in the cloning process
- Cloning of IBM Middleware: A new chapter 11 \textit{Cloning IBM Middleware} was added that describes installing, sharing, customizing and cloning IBM WebSphere Application Server, DB2 and MQ Series.
- The \texttt{clone.sh} script has three new flags added: \texttt{--was}, \texttt{--db2} and \texttt{--mqs} to support the cloning of IBM middleware.

Summary of changes to February 2007 version

The release of two new books in February of 2007 added the following:

© Copyright IBM Corp. 2007. All rights reserved.
SLES 10 is described in the book *z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES 10*.

RHEL 5 is described in the book *z/VM and Linux on IBM System z: The Virtualization Cookbook for RHEL 5*.

Two VDISK swap spaces are used rather than one VDISK and one minidisk - this allows for both more potential swap space and a larger root file system.

Associated files for the controller are distributed as RPMs - one for SLES 10 and one for RHEL 5. This removes the need to recursively copy the content of the associated tar files.

A section is added to the chapter “Miscellaneous recipes” entitled “Centralizing home directories for LDAP users”. This ties together a virtual LDAP server, authentication (PAM, NSS), LVM, NFS and automount so that cloned virtual servers share a common user/group/password LDAP database and a common /home/ directory that is automatically mounted.

Another section is added to the chapter “Miscellaneous recipes” entitled “Rescuing a Linux system”. This is a quick reference on how to begin to repair a system that is no longer able to boot. Hopefully this section will not be referenced often (or at all :)).

A Web Application named Data About z/VM and Linux (DAZL) is described in Chapter 13, “Monitoring z/VM and Linux” on page 197. This allows the controller to run a password protected Web Server that makes much z/VM and Linux data quickly available via a browser. It relies on the /backup/ file system being populated correctly via the clone.sh script, rsync, key-based authentication and cron.

The clone.sh script was moved from /usr/local/sbin/ to /sbin/. The hard coded reference to the master image’s mount point has been removed. Rather, only the root file system minidisk (100) is cloned. That value was made into a variable to more easily allow for multiple master images to be cloned.

Installing z/VM onto 3390-9s is addressed - a LBL520-9 EXEC and XEDIT macro are supplied to allow for the relabeling of the system volumes.

The release of two new books also removed some sections:

- The chapter on Cloning of IBM Middleware has been removed - at the time of publishing not all the middleware was not supported on the new distributions.
- DCSS/XIP2 - it does not seem that this environment is being widely used.

## Conventions

The following font conventions are used in this book:

**Monospace and bold** Commands entered by the user on the command line

<value> Value inside angle brackets is to be replaced

monospace File, directory and user ID names

The following command conventions are used in this book:

- z/VM commands are prefixed with `==>
- z/VM XEDIT subcommands are prefixed with `====>
- Linux commands running as root are prefixed with `#
- Linux commands that will not fit on one line are suffixed with `
- Linux commands running as non-root are usually prefixed with `$
The team that wrote this trilogy

This book was originally written in 2005 by Michael Maclsaac, Jin Xiong and Curtis Gearhart. It was updated in 2006 by Michael Maclsaac, Carlos Ordonez and Jin Xiong. It was updated a third time in late 2006 and early 2007 by Marian Gasparovic taking the lead on the SLES 10 version and Brad Hinson of Red Hat working on the RHEL 5 version. The DCSS/XIP2 section was based on work by Carlos Ordonez, which was in turn based on work by Carsten Otte. Most of the section on the CP INDICATE command was written by Bill Bitner.

Sincere thanks goes out to the following people who contributed to this project in various ways:

Jerry Epker
Fidelity Information Services

Bill Bitner, Bruce Hayden, Denny Refsnider, Roger Lunsford, Jim Switzer and Romney White
IBM Endicott

Carsten Otte, Claudia Prawirakusumah
IBM Germany

Kim Betros, Duane Beyer, Michel Considine, Roy Costa, Greg Geiselhart, Dionne Graff, Susan Greenlee, Ernest Horn, Rong Lee, Bill Norton, Bruce Smilowitz, Kyle Smith, Paul Sutera, Helen Tsang, Donna Von Dehsen, Hossee Wakil and Dennis Wunder
IBM Poughkeepsie

Bernard Wu
NCCI

Mark Post
Novell

Matthew Eernisse
OpenSource Applications

David Boyes, Adam Thornton
Sine Nomine Associates

Chester Hood
State of Tennessee

Rob van der Heij
Velocity Software

Thanks the zBMC team in IBM Poughkeepsie and to the many who answered questions on the linux-390 list server.

Special thanks to Tony Giaccone from IBM Poughkeepsie for always finding another free LPAR and more DASD.

Comments welcome

Your comments are important. You can send your comments in e-mail directly to:
mikemac at us.ibm.com
marian.gasparovic at sk.ibm.com
bhinson at redhat.com // For the RHEL 5 book
Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious, like anyone will ever read this, and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.
Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States,

- developerWorks®
- eServer™
- ibm.com®
- iSeries™
- z/OS®
- z/VM®
- System z®
- AIX®
- DirMaint™
- Domino®
- ECKD™
- ETE™
- HiperSockets™
- Hummingbird®
- IBM®
- Lotus®
- Notes®
- Perform™
- Requisite®
- RACF®
- RMF™
- S/390®
- SLC™
- Tivoli®
- Workplace™

The following terms are trademarks of other companies:

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.
This part of the book has an introduction chapter and then discusses z/VM installation, configuration, and servicing.
Chapter 1. Introduction to z/VM and Linux

“Everything should be made as simple as possible, but not simpler.”
— Albert Einstein

Virtualization is hot. The mainframe and z/VM have been doing virtualization for decades. When Linux came to the mainframe in 2000, it was a natural fit to run under z/VM. You can run many tens of Linux images on the same System z logical partition (LPAR). Some customers are running hundreds.

With a z/VM and Linux infrastructure, you can reduce the time between deciding on new servers and implementing them because new servers can be deployed in a matter of minutes. This powerful capability can enable you to launch new products and services without planning, purchasing, installing and configuring new hardware and software. Development groups who need test environments built and rebuilt rapidly to efficiently handle change management can also benefit from this unique advantage.

Some of the mainframe's and z/VM's best strengths are:

- z/VM and the mainframe's virtualization capabilities are more mature and robust than any other hardware and virtualization combination.
- z/VM's virtual switch (VSWITCH) has made networking Linux much simpler.
- Full volume backup of systems allows for complete disaster recovery when another data center is available.
- z/VM is one of the easiest operating systems to customize. There is only a handful of configuration files. Once it is set up, z/VM will run for months with little maintenance and administration required.

z/VM 5.2, available in December of 2005, provides major improvements when operating on System z servers with large memory configurations. Scalability is improved with the Control Program (CP) now using memory above 2 GB for a much broader set of operations. Previously, guest pages had to be moved below 2GB for many reasons, for example in both standard I/O and Queued Direct I/O (QDIO). Now I/O can be done using buffers anywhere in real memory, and QDIO structures can reside above 2 GB, as can most CP control blocks. These improvements offer constraint relief for large-real-memory virtual server environments that are memory intensive.
1.1 What is virtualization?

Virtualization is the ability for a computer system to share resources so that one physical server can act as many virtual servers. z/VM allows the sharing of the mainframe's physical resources such as disk (DASD), memory (storage), network adapters (OSA cards) and CPU (CPs or IFLs). These resources are managed by a hypervisor. z/VM's hypervisor is called Control Program (CP). When the user logs onto z/VM, the hypervisor creates a virtual machine which can run one of many different operating systems. The two operating systems that are discussed in this book are the Conversational Monitoring System (CMS) and Linux. CMS can be thought of as a z/VM shell. Virtual machines running Linux become the virtual servers.

1.2 A philosophy adopted in this book

An important philosophy adopted in this book is to keep all solutions simple. Two common expressions used are “the KISS method” (Keep It Simple, Stupid) and the quote from Albert Einstein at the start of this chapter: Everything should be made as simple as possible, but not simpler. Because the former is somewhat condescending and because 2005 was both the first year this book was published and the centennial anniversary of Albert Einstein's many famous papers, this book will use the latter.

1.3 Choices and decisions made in this book

When deciding on installing, maintaining and provisioning (cloning) Linux virtual servers under z/VM, there are many basic choices to make. Here are some of the more important choices and assumptions made in writing this book:

- Cloning product versus “roll your own” cloning:
  Cloning products, such as Aduva’s Onstage, IBM Tivoli Provisioning Manager, IBM Director function ‘z/VM Center’ (briefly discussed in “IBM Director and z/VM Center Extension” on page 5) and Levanta, are outside the scope of this book. While these are all viable solutions, the cloning described in this book allows you to roll your own Linux images without requiring such products. However, these products are more sophisticated than the simple EXECs and scripts in this book.

- Directory Maintenance product versus the USER DIRECT file:
  The USER DIRECT file is chosen over a directory maintenance product such as IBM DirMaint or Computer Associates' VM:Direct. If you feel that DirMaint as a directory maintenance product is better for your enterprise, you can use the book Getting Started With Linux, SC24-6096, to configure z/VM. You can still use this book to configure Linux.

- Provisioning versus predefined user IDs:
  z/VM user IDs must be predefined to clone. There is no attempt to provision them (define and bring Linux user IDs online automatically) as part of the cloning process. The target Linux user ID must exist with the appropriate minidisks defined or the cloning script will fail.
Shared read-only Linux /usr/ file system versus read-write:

Many cloning solutions use an environment which shares the /usr/ file system. This choice often makes the solution more complex, especially when adding software to the virtual servers. A read-write /usr/ file system on the virtual servers is chosen to keep things as simple as possible.

Conventional 3390 ECKD DASD versus FBA disks accessed via SCSI over FCP:

The System z server has traditionally only supported 3390 (or older 3380) DASD. Support has been extended to include SCSI/FBA disks in storage area networks (SANs). The support of FBA disks is slightly more complicated than conventional DASD. In keeping things as simple as possible, only conventional DASD is described in this book.

Cloning script or EXEC versus manual installation:

It is easy to spend more time setting up an infrastructure for cloning Linux under z/VM than the time that it saves you over manually installing Linux, given the number of times you actually clone. When cloning works quickly, it can be an extremely useful tool in the system administrator’s toolbox. Therefore, this book discusses both cloning and manual installation. These two methods of provisioning Linux servers hinge on the existence of an installation parameter file for each Linux user ID. If you want a more complete solution, the products recommended in the first bullet point are recommended.

1.4 IBM Director and z/VM Center Extension

IBM Director 5.10 brings a comprehensive management functionality to Linux on IBM system z. The base IBM Director functions (e.g. monitoring, event action plans, software distribution, inventory, remote control, task scheduling) are now provided for any Linux end point on System z. In addition, the z/VM Center Extension provides further functionality for provisioning and configuration of z/VM Linux guests.

The z/VM Center extension includes the following tasks:

- **Virtual Server Deployment** - creation of virtual servers and deployment of operating systems into them by using virtual server and operating system templates, management of virtual servers (create/delete/activate/properties) and provisioning resources.
- **Server Complexes** - automatic fashion of controlling the configuration (and creation) of groups of Linux guests, handling both the z/VM side and Linux side aspects, supporting z/VM Resource Manager performance goals, virtual networking (based on VM Guest LAN, OSA and VSWITCH), z/VM minidisk attachments and configuration scripts.

The integration of the z/VM Center virtualization functionality with the full breadth of IBM Director on Linux managed end points provides a powerful tool for managing Linux guest colonies on z/VM systems.

1.5 Infrastructure design

To clone Linux, or provision virtual servers, there must be a certain infrastructure design in place. A System z server with associated resources and the z/VM operating system define much of this infrastructure. Figure 1-1 on page 6 shows a block diagram of a System z9 with many LPARs. z/VM 5.2 is installed in one of these LPARs. z/VM comes with many user IDs predefined. The most important six IDs are shown in the z/VM LPAR above the dashed line. Below the dashed line, you see the user IDs described in this book. Important z/VM minidisks and configuration files are shown next to each user ID.
Figure 1-1  System infrastructure and z/VM user IDs

The user IDs above the dashed line are those important user IDs defined in z/VM 5.2. The user IDs below the dashed line are described in this book and have the following functions:

- **LNXMAINT**: A user ID on which to store files that will be used by both CMS and Linux
- **<MAIN ID>**: The user ID that stores the master (or golden) image that is cloned and IPLed from minidisk 100 and the controller (or cloner) image that does the cloning and is IPLed from minidisk 200. In the RHEL 5 book, this user ID is named RHEL5 and in the SLES 10 book it is named SLES10.
- **LINUX01-04**: The Linux virtual servers described in the Cloning open source virtual servers chapter. In addition, creating LINUX05-07 is described so you can have additional user IDs to clone to. Each virtual server is configured with a single 3390-3 minidisk which is slightly more than 2 GB of space.

### 1.6 Usability tests performed for this book

During the writing of this book, many usability tests were conducted. The participants had a variety of skills, but none had both Linux and z/VM system administration skills. By the end of the first day in all of the formal tests, most participants had all completed up to and including Chapter 5, “Servicing z/VM” on page 69, so z/VM was installed, serviced and customized for TCP/IP communications with a highly available VSWITCH. By the end of the second day, most participants had cloned their first Linux virtual server. You should be able to complete most steps in the book in four solid days of work, if all goes well.
1.7 The chapters in this book

The remaining chapters and appendixes in this book are summarized in the following list:

- **Chapter 2, “Planning”** on page 9 describes how to plan hardware, software and networking resources. It discusses DASD labeling conventions used in the book and a password planning. Sample worksheets are provided for the examples used in the book, as are blank copies for your use.

- **Chapter 3, “Configuring a desktop machine”** on page 21 describes how to set up Windows desktops. Specifically, the following tools are discussed:
  - How to get and set up PuTTY: a commonly used SSH client
  - How to get and set up a VNC client: a tool for running graphical applications
  - 3270 emulator applications

- **Chapter 4, “Installing and configuring z/VM”** on page 29 shows how to install and configure z/VM. This is where you roll up your sleeves and start to work.

- **Chapter 5, “Servicing z/VM”** on page 69 describes how to apply service to z/VM both in the form of Programming Temporary Fixes (PTFs) and Recommended Service Upgrades (RSUs).

- **Chapter 6, “Configuring an NFS server for SLES 10”** on page 83, explains how to set up a temporary NFS server on a Linux PC for the purpose of installing the first two Linux images. After the System z controller Linux is installed, you can copy the Linux install tree to it and retire the Linux PC server.

- **Chapter 7, “Installing and configuring SLES 10”** on page 89, describes how to install and configure two Linux images onto the first Linux user ID: the master image, which it is cloned from, and the controller, which does the cloning among other tasks.

- **Chapter 8, “Configuring NFS on controller”** on page 131, illustrates how to move the Linux install tree from the Linux PC server to the controller under z/VM.

- **Chapter 9, “Configuring SLES 10 for cloning”** on page 139 explains how to prepare z/VM user IDs and clone your first virtual server.

- **Chapter 10, “Cloning open source virtual servers”** on page 155, shows how to configure cloned Linux images into the following virtual servers:
  - Web server virtual server
  - LDAP virtual server
  - File and print virtual server
  - Basic application development system

- **Chapter 11, “Cloning IBM middleware on SLES 10”** on page 177 describes how to install, share, customize and clone IBM WebSphere Application Server, DB2 and MQ Series.

- **Chapter 12, “Miscellaneous recipes”** on page 179 describes how to add a logical volume to a Linux system and how to set up a z/VM Discontiguous Saved Segment (DCSS) in conjunction with the Linux eXecute In Place 2 (xip2) file system.

- **Chapter 13, “Monitoring z/VM and Linux”** on page 197, describes basic steps to begin monitoring z/VM and your new Linux virtual servers.

- **Chapter 14, “Backup and restore”** on page 223, shows basic steps on how to back up these new systems.

- **Appendix A, “References”** on page 229, provides references Web sites, books and other pertinent information.

- **Appendix B, “Common source code”** on page 233 lists source code of the REXX EXECs and shell scripts common, regardless of Linux distribution.
Appendix C, “Source code specific to Linux” on page 249 lists source code of the REXX EXECs and shell scripts common, regardless of Linux distribution.
Planning

“The only reason for time is so that everything doesn’t happen at once.”
— Albert Einstein

This chapter covers the planning that should be done before installing z/VM. It begins by discussing *a bill of materials*, or all the resources that you need. Then it explains the labeling of 3390 volumes. Finally resource worksheets are presented for:

- z/VM resources other than direct access storage device (DASD)
- DASD resources
- Linux resources
- Linux user IDs

2.1 Bill of materials

The resources needed for a Linux on zSeries project can be divided into:

- Hardware
- Software
- Networking

2.1.1 Hardware resources

The following hardware is needed:

- A zSeries logical partition (LPAR); z800, z900, z890 or z990, or System z9
  - Processors or CPUs: One IFL (or CP) minimum, two or more recommended
  - Memory: 3 GB central/1 GB expanded minimum, 6 GB/2 GB or more recommended. This 3:1 ratio of central to expanded storage is a good starting point.
    See the following Web site for a discussion of how to apportion memory:
  - DASD: 24 3390-3s or 9 3390-9s at an absolute minimum
– Open Systems Adapter (OSA) network cards: One card minimum with 12 device numbers (technically 9, but OSA “triplets” usually start on an even address). Two OSA Express cards with eight device numbers on one and four on the other is recommended for high availability.

» A network-attached computer that will act as an Network File System (NFS) server temporarily with at least 6 GB of disk space, but more may be needed. A Linux PC or UNIX server is recommended.

» A workstation or desktop that has network access to the mainframe

2.1.2 Software resources

The following software resources are needed:

» z/VM 5.2 install media with documentation (DVD install is described in this book.)

» Linux install media (SLES or RHEL)

» An operating system for the NFS server

» The code associated with this book

» Tools on the workstation and desktop:
  – A 3270 Emulator such as Attachmate Extra, Hummingbird Host Explorer, or IBM Personal Communications for Windows desktops (for Linux desktops, a 3270 emulator named x3270 is available)
  – A Linux SSH client such as PuTTY (recommended) or TeraTerm (for Linux desktops the ssh client is built in)
  – A VNC viewer

These resources are described in more detail in the chapters that follow.

2.1.3 Networking resources

The following network resources are needed:

» A TCP/IP address for z/VM

» One TCP/IP address for each Linux virtual server

» Associated TCP/IP information:
  – DNS host name
  – DNS domain
  – DNS server TCP/IP address
  – TCP/IP gateway
  – TCP/IP subnet mask
  – TCP/IP broadcast address (usually calculated from address and subnet mask)
  – TCP/IP MTU size

The TCP/IP addresses should be routed to the OSA card(s).

2.2 z/VM conventions

It is good to use conventions so that you and others can recognize z/VM resources by their names. This section discusses conventions for DASD volume names and backup file names.
2.2.1 Volume labeling convention

You should have a convention for labeling DASD. Your shop may already have a labeling convention which will largely determine the labels to be given to the DASD used by your z/VM and Linux LPAR.

Each zSeries DASD is addressed with a device number consisting of four hexadecimal digits. Each zSeries DASD has a six character label. It is convenient to include the four-digit address in the label so that you can easily tell the address of each DASD from its label. When followed, this convention guarantees that no two DASD will have the same label. This can be an important issue especially when z/OS has access to the DASD.

Sometimes DASD is shared among LPARs in which case your z/VM LPAR can see DASD owned by other LPARs. In this situation, it is convenient to identify the LPAR that owns the DASD. Therefore the volume labeling convention used in this book identifies the LPAR via the first character. That leaves the second character in the label to identify the basic function of the DASD.

The LPAR used in this book is identified by the character M. The following characters are used for the types of DASD in the second character of the label:

- **M**: Minidisk space (PERM)
- **P**: Paging space (PAGE)
- **S**: Spool space (SPOL)
- **T**: Temporary disk space (TDISK)
- **V**: z/VM operating system volumes

For example, Figure 2-1 shows the labeling convention for the DASD in LPAR M, of type minidisk at real address A700.

![Figure 2-1 DASD labeling convention](image)

The letter M is hard-coded into two EXECs that adopt this convention. If you want a different LPAR identifier character, they can easily be changed.

2.2.2 Backup file naming convention

It is recommend that you keep copies of important z/VM and Linux configuration files. You should always keep copies of original configuration files in case you need to go back to them. Since z/VM file names are limited to 16 characters (eight for the file name and eight for the file type), only the last four characters of the file type are used. This often requires some characters to be overwritten. For the original file, the suffix ORIG is used, and for the most recent working copy, the suffix WRKS (for “it WoRKS!”) is used. For example, the original USER DIRECT file is copied to the file USER DIREORIG before it is modified the first time.
2.2.3 The command retrieve convention

The ability to retrieve past commands is a common tool. Often it is nice to retrieve in both
directions in case you “pass” the command you’re looking for. The default Linux shell, bash,
does this by default with the up arrow and down arrow keys.

There is a convention in z/VM to use the F12 function key (labeled PF12 on physical 3270
devices) to retrieve the last command, though it is not defined to all user IDs. There is no
convention retrieve commands in the other direction but it is possible to set another key to that
function. Therefore, F11 is used to retrieve forward since it is right next to F12. Also, the same
function is useful in the editor, XEDIT. The ? subcommand retrieves past commands, so it is
recommended that you assign it to F12.

2.3 Disk planning

There are different aspects to consider when planning how to choose and allocate disk
storage. Some aspects include the following

▶ Conventional ECKD DASD vs. FBA disks over SCSI/FCP
▶ 3390-3s vs. 3390-9s or large disk support
▶ Amount of disk storage per Linux image and how to allocate file systems

DASD vs SCSI/FCP
This book describes how to use conventional ECKD DASD and does not discuss FBA disks
accessed over SCSI/FCP. This is not because either technology is superior, but simply
because DASD seems to be much more common than SCSI/FCP disks. If you were to use
SCSI/FCP disks, cloning via the clone.sh script would have to be modified to account for
World Wide Port Names and Numbers. Sometimes a combination of these two types of disk
storage is used - when that is the case the ECKD emulated DASD is often used for the root
file system and SCSI/FCP disks are used for large data storage areas.

3390-3s vs. 3390-9s
Emulated 3390-3s format to about 2.3GB, while 3390-9s are three times the size or about
6.9MB. Either size will work, though 3390-3s have been recommended over 3390-9s by some
performance analysts. This book describes mainly using 3390-3s, however, comments are
added where using 3390-9s differs - especially with installing z/VM.

Disk storage per Linux image
A single 3390-3, or 3339 cylinders, is allocated to the root file system (/) for each Linux image.
Often recommendations are made to create many more file systems to be mounted over
other directories, such as /usr/, /var/, /tmp/, /home/, /opt/, etc. This model is not used for
two reasons:

▶ It is difficult to come up with a default set of file system sizes given that the Linux virtual
  servers will be running many different applications and workloads after being cloned.
▶ Under the philosophy of keep things as simple as possible.

One common argument for having many mounted file systems is so that the root file system
will not fill up. To help prevent this from happening, it is recommended to add mounted file
systems after cloning under the directory where data and applications will be added. For
example, in the chapters that follow, a /nfs/ file system is added to the controller to store the
install tree and a /home/ file system is added to LINUX02 to store user’s data.
2.4 Memory planning

Planning memory may be the most difficult issue with System z Linux and z/VM, yet the most important to ensure adequate performance. The simplest solution may be to have enough central memory (storage) in the LPAR so that z/VM never pages and Linux never swaps. However, such resource may not be realistically available.

Discussing memory planning further is outside the scope of this book (and the authors’ knowledge :)). The following resources are recommended:

- The Redbook *Linux on IBM zSeries and S/390: Performance Measurement and Tuning*, 2003, on the Web at:
- The IBM z/VM Performance Resource pages in general, on the Web at:
- The IBM z/VM page specifically discussing memory allocation:

One simple rule that can be recommended is to have as few virtual machines logged on (or disconnected) as possible. Every virtual machine that can be logged off will mean more memory for the other virtual servers.

2.5 Password planning

Good passwords are critical to good security. However, requiring many different passwords leads to people writing them down, which detracts from good security. Sometimes it is difficult to balance these two extremes.

This book considers different system administration roles:

- The z/VM system administrator
- The Linux system administrator
- The Linux virtual server end users

The z/VM and Linux system administrator may well be the same person.

The method of backing up z/VM data onto the Linux controller means that the Linux administrator will have access to all z/VM passwords. Therefore, the examples in this book set all z/VM and Linux system administration passwords to the same value, `lnx4vm`. If the z/VM and Linux system administrator roles must be kept separate and the Linux administrator is not to have access to the z/VM passwords, then a different method of backing up z/VM data must be chosen.

You may want to define a finer granularity for passwords based on the following system administration roles:

- The main z/VM system administrator (MAINT)
- The z/VM network administrator (TCPMAINT)
- The z/VM Linux administrator (LNXMAINT, Linux controller, Linux virtual server user IDs)
- The Linux virtual server end users (with or without access to 3270 sessions, with or without the root passwords)

The sets of passwords that you define will depend on the roles that your organization will adopt.
2.6 Planning worksheets

Four worksheets are included in this section. They are populated with the resources used in writing this book. There are also four corresponding blank worksheets in 2.7, “Blank worksheets” on page 17.

2.6.1 z/VM resources used in this book

Table 2-1 lists the z/VM resource values used in the examples in this book. You can use these values as a reference for completing the blank worksheets that follow.

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPAR name</td>
<td>P21</td>
<td>3 GB central storage/1 GB expanded, 4 shared IFLs</td>
</tr>
<tr>
<td>CPC name</td>
<td>PELCP01</td>
<td>Name of CPC on which the LPAR is located</td>
</tr>
<tr>
<td>z/VM system name</td>
<td>LNXVM52</td>
<td>Name to be assigned to z/VM system</td>
</tr>
<tr>
<td>TCP/IP host name</td>
<td>lat120</td>
<td>Assigned by a network administrator; helpful to set in DNS beforehand, but not necessary</td>
</tr>
<tr>
<td>TCP/IP domain name</td>
<td>pbm.ihost.com</td>
<td>Helpful to set in DNS beforehand</td>
</tr>
<tr>
<td>TCP/IP gateway</td>
<td>129.40.178.254</td>
<td>The router to and from the local subnet</td>
</tr>
<tr>
<td>DNS server 1</td>
<td>129.40.106.1</td>
<td>Assigned by the network administrator</td>
</tr>
<tr>
<td>DNS server 2/3 (optional)</td>
<td></td>
<td>Not used</td>
</tr>
<tr>
<td>OSA device name</td>
<td>eth0</td>
<td>Name of the interface to be assigned by IPWIZARD</td>
</tr>
<tr>
<td>OSA starting device number</td>
<td>3000</td>
<td>Start of OSA triplet for the z/VM TCP/IP stack</td>
</tr>
<tr>
<td>TCP/IP address</td>
<td>129.40.178.120</td>
<td>The TCP/IP address of the z/VM system</td>
</tr>
<tr>
<td>Subnet mask</td>
<td>255.255.255.0</td>
<td>Assigned by network administrator</td>
</tr>
<tr>
<td>OSA device type</td>
<td>QDIO</td>
<td>Often “QDIO” for OSA/Express cards</td>
</tr>
<tr>
<td>Network type</td>
<td>Ethernet</td>
<td>Usually “Ethernet”</td>
</tr>
<tr>
<td>Port name (optional)</td>
<td></td>
<td>Not required by z/VM</td>
</tr>
<tr>
<td>Router type</td>
<td>None</td>
<td>Usually “None”</td>
</tr>
<tr>
<td>Primary OSA device number for VSWITCH</td>
<td>3004</td>
<td>Specify the first device number (must be even number) and the next two device numbers will also be used</td>
</tr>
<tr>
<td>Secondary OSA device number for VSWITCH</td>
<td>3008</td>
<td>Should be on a different CHPID/OSA card</td>
</tr>
</tbody>
</table>
2.6.2 z/VM DASD used in this book

Table 2-2 lists the z/VM DASD resource values used in the examples in this book.

<table>
<thead>
<tr>
<th>Device number</th>
<th>Label</th>
<th>Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>A700</td>
<td>520RES</td>
<td>CP owned</td>
<td>z/VM System residence volume</td>
</tr>
<tr>
<td>A701</td>
<td>520SPL</td>
<td>System spool</td>
<td>Spool volume 1 from z/VM installation</td>
</tr>
<tr>
<td>A702</td>
<td>520PAG</td>
<td>System page</td>
<td>Paging volume 1 from z/VM installation</td>
</tr>
<tr>
<td>A703</td>
<td>520W01</td>
<td>CP owned</td>
<td>z/VM first work volume</td>
</tr>
<tr>
<td>A704</td>
<td>520W02</td>
<td>CP owned</td>
<td>z/VM second work volume</td>
</tr>
<tr>
<td>A705</td>
<td>MPA705</td>
<td>System page</td>
<td>Paging volume 2</td>
</tr>
<tr>
<td>A706</td>
<td>MPA706</td>
<td>System page</td>
<td>Paging volume 3</td>
</tr>
<tr>
<td>A707</td>
<td>MPA707</td>
<td>System page</td>
<td>Paging Volume 4</td>
</tr>
<tr>
<td>A708</td>
<td>MPA708</td>
<td>System page</td>
<td>Paging Volume 5</td>
</tr>
<tr>
<td>A709</td>
<td>MPA709</td>
<td>System page</td>
<td>Paging volume 6</td>
</tr>
<tr>
<td>A70A</td>
<td>MMA70A</td>
<td>System minidisk</td>
<td>SLES10 100 - the master image</td>
</tr>
<tr>
<td>A70B</td>
<td>MMA70B</td>
<td>System minidisk</td>
<td>SLES10 200 - the controller</td>
</tr>
<tr>
<td>A70C</td>
<td>MMA70C</td>
<td>System minidisk</td>
<td>SLES10 203 and LNXMAINT 191 and 192</td>
</tr>
<tr>
<td>A70D</td>
<td>MMA70D</td>
<td>System minidisk</td>
<td>SLES10 204, part of the /nfs/ logical volume</td>
</tr>
<tr>
<td>A70E</td>
<td>MMA70E</td>
<td>System minidisk</td>
<td>SLES10 205, part of the /nfs/ logical volume</td>
</tr>
<tr>
<td>A70F</td>
<td>MMA70F</td>
<td>System minidisk</td>
<td>SLES10 206, part of the /nfs/ logical volume</td>
</tr>
<tr>
<td>A710</td>
<td>MMA710</td>
<td>System minidisk</td>
<td>SLES10 207, part of the /nfs/ logical volume</td>
</tr>
<tr>
<td>A711</td>
<td>MMA711</td>
<td>System minidisk</td>
<td>LINUX01 100</td>
</tr>
<tr>
<td>A712</td>
<td>MMA712</td>
<td>System minidisk</td>
<td>LINUX02 100</td>
</tr>
<tr>
<td>A713</td>
<td>MMA713</td>
<td>System minidisk</td>
<td>LINUX03 100</td>
</tr>
<tr>
<td>A714</td>
<td>MMA714</td>
<td>System minidisk</td>
<td>LINUX04 100</td>
</tr>
<tr>
<td>A715</td>
<td>MMA715</td>
<td>System minidisk</td>
<td>LINUX05 100</td>
</tr>
<tr>
<td>A716</td>
<td>MMA716</td>
<td>System minidisk</td>
<td>LINUX06 100</td>
</tr>
<tr>
<td>A717</td>
<td>MMA717</td>
<td>System minidisk</td>
<td>LINUX07 100</td>
</tr>
</tbody>
</table>
2.6.3 Linux resources used in this book

Table 2-3 lists the Linux resources used in the examples in this book.

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux install password</td>
<td>lnx4vm</td>
<td></td>
</tr>
<tr>
<td>Linux TCP/IP gateway</td>
<td>129.40.178.254</td>
<td></td>
</tr>
<tr>
<td>Linux TCP/IP broadcast</td>
<td>129.40.178.255</td>
<td></td>
</tr>
<tr>
<td>Linux DNS server</td>
<td>129.40.106.1</td>
<td>Often the same as z/VM’s</td>
</tr>
<tr>
<td>NFS server TCP/IP address</td>
<td>129.40.46.206</td>
<td></td>
</tr>
<tr>
<td>VNC installation password</td>
<td>lnx4vm</td>
<td></td>
</tr>
</tbody>
</table>

2.6.4 Linux user IDs used in this book

Table 2-4 lists the Linux user IDs used in the examples in this book.

<table>
<thead>
<tr>
<th>Linux user ID</th>
<th>IP address</th>
<th>DNS name</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLES10</td>
<td>129.40.178.130</td>
<td>lat130.pbm.ihost.com</td>
<td>A 64-bit controller and master image</td>
</tr>
<tr>
<td>LINUX01</td>
<td>129.40.178.121</td>
<td>lat121.pbm.ihost.com</td>
<td>A Web virtual server</td>
</tr>
<tr>
<td>LINUX02</td>
<td>129.40.178.122</td>
<td>lat122.pbm.ihost.com</td>
<td>An LDAP virtual server</td>
</tr>
<tr>
<td>LINUX03</td>
<td>129.40.178.123</td>
<td>lat123.pbm.ihost.com</td>
<td>A file and print virtual server</td>
</tr>
<tr>
<td>LINUX04</td>
<td>129.40.178.124</td>
<td>lat124.pbm.ihost.com</td>
<td>An application development virtual server</td>
</tr>
<tr>
<td>LINUX05</td>
<td>129.40.178.125</td>
<td>lat125.pbm.ihost.com</td>
<td>A WebSphere Application Server</td>
</tr>
<tr>
<td>LINUX06</td>
<td>129.40.178.126</td>
<td>lat126.pbm.ihost.com</td>
<td>A DB2 Server</td>
</tr>
<tr>
<td>LINUX07</td>
<td>129.40.178.127</td>
<td>lat127.pbm.ihost.com</td>
<td>An MQ Series server</td>
</tr>
</tbody>
</table>
2.7 Blank worksheets

Blank copies of the same four worksheets are provided for your use.

2.7.1 z/VM resources worksheet

Use the worksheet in Table 2-5 to document the z/VM resources that you will use.

Table 2-5  z/VM resources blank worksheet

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPAR name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPC name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>System name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCP/IP host name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCP/IP domain name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCP/IP gateway</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DNS server 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DNS server 2/3 (optional)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OSA device name</td>
<td></td>
<td>Often “eth0”</td>
</tr>
<tr>
<td>OSA starting device number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCP/IP address</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subnet mask</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OSA device type</td>
<td></td>
<td>Often “QDIO”</td>
</tr>
<tr>
<td>Network Type</td>
<td></td>
<td>Often “Ethernet”</td>
</tr>
<tr>
<td>Port name (optional)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Router Type</td>
<td></td>
<td>Often “None”</td>
</tr>
<tr>
<td>Primary OSA device number for VSWITCH</td>
<td></td>
<td>Should be on a different CHPID/OSA card than primary</td>
</tr>
<tr>
<td>Secondary OSA device number for VSWITCH</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2.7.2 z/VM DASD worksheet

Use the worksheet in Table 2-6 to document the z/VM DASD that you will use.

Table 2-6 z/VM DASD blank worksheet

<table>
<thead>
<tr>
<th>Device number</th>
<th>Label</th>
<th>Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2.7.3 Linux resources worksheet

Use the worksheet in Table 2-7 to document your Linux resources.

Table 2-7 Linux resources blank worksheet

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFS server TCP/IP address</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linux install password</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linux root password</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Apache user ID and password</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linux TCP/IP gateway</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linux TCP/IP broadcast</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linux DNS server</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VNC Installation password</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.7.4 Linux user ID worksheet

Use the worksheet in Table 2-8 to document the Linux user IDs that you will create.

Table 2-8 Linux user ID blank worksheet

<table>
<thead>
<tr>
<th>Linux user ID</th>
<th>IP address</th>
<th>DNS name</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Configuring a desktop machine

“Technological progress is like an axe in the hands of a pathological criminal.”
— Albert Einstein

Many people use Microsoft Windows as a desktop operating system. This chapter addresses the following tools that are recommended for accessing z/VM and Linux from a Windows desktop:

- An SSH client: PuTTY is recommended
- A VNC client: RealVNC is recommended
- A 3270 emulator: Many choices are available

### 3.1 PuTTY: A free SSH client for Windows

Throughout this book, SSH is used to log into Linux systems. It is easy to use and cryptographically secure. If you are using a Linux desktop system, an SSH client is built in. But if you are using a Windows desktop, you will need a good SSH client.

PuTTY is probably the most commonly used. You can find a PuTTY client for Windows on CD1 of a SLES 9 or 10 distribution in the /dosutils/putty directory. You can download PuTTY from the Web at:

http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

To download from this page, click on the putty.exe link for your architecture. Save the file in a directory path such as C:\WINNT. PuTTY is a stand-alone executable (no installation needed other than copying the file). You may also want to create a shortcut on your desktop or task bar.

Open PuTTY and the configuration window shown in Figure 3-3 should open. If you spend a few minutes to configure PuTTY it may pay off in time savings.
1. In the PuTTY Configuration window, in the left Category panel, click SSH.
2. Under the Protocol heading on the right, click the SSH radio button. This specifies to use the SSH protocol.

![PuTTY Configuration window](image1)

*Figure 3-1 PuTTY Configuration window*

3. Click on Logging in the left panel. Click the radio button Log printable output only. This will allow you to go back and check on the output of certain commands. Set the Log file name to &H&M&D&T.log so it will include a timestamp in the file name.

![Setting Logging](image2)

*Figure 3-2 Setting Logging*
4. In the left panel, click **SSH** near the bottom.

5. On the right side, under *Preferred SSH protocol version*, click the **2 only** radio button.

6. In the left Category panel, click **Terminal**.

7. Select the **Use background colour to erase screen** check box, which results in a better job of painting the screen for applications that uses curses (block graphics).
8. Click **Window** in the left pane.

9. You see the window in Figure 3-5. You may choose more rows and columns than the default of 24x80. In this example, 43 rows and 100 columns are set.

![PuTTY Configuration](image)

**Figure 3-5 Customizing PuTTY window settings (Part 3 of 4)**

**Saving sessions**

1. Click **Session** to get back to the initial window when PuTTY is invoked (see Figure 3-6).

2. In the **Save Sessions** area, select **Default Settings** and click **Save**. This makes all future sessions that you define inherit the preferences you just set.

![PuTTY Configuration](image)

**Figure 3-6 Customizing PuTTY window settings (Part 4 of 4)**
Now to save a session for each virtual server, perform the following:

1. In the Host Name (or IP address) field, enter the TCP/IP address (or DNS name).
2. Under Saved Sessions, choose a name that you will remember. In this example, the name `lat40 (LINUX00)` is used. This is the DNS name and the z/VM user ID.
3. Again click **Save** and you should see the name added to the **Saved Session** list.

Now whenever you start PuTTY, you can simply double-click any saved session name, and an SSH session to the desired Linux system is invoked. The final window should appear similar to that shown in Figure 3-7.

![PuTTY Configuration](image)

**Figure 3-7** Saving default PuTTY settings and creating new sessions

### 3.2 Setting up a VNC client

A VNC client allows access to a graphical environment with System z Linux.

If you are using a Linux desktop you probably have, or at least have access to a VNC client, named `vncviewer`. It is part of the tightvnc package.

#### 3.2.1 Downloading and running RealVNC

If you have a Windows desktop, the VNC client from RealVNC is a popular choice (you can also find a TightVNC client for Windows on CD1 of a SLES 9 or 10 distribution in the `/dosutils/tightvnc` directory).

You can purchase a full function RealVNC client, or there is a free version. The RealVNC home page is:

http://www.realvnc.com

The download page is:
Fill out the Web form and download the executable. When you have downloaded it, run it and an install program will start. At the time of writing of this book, RealVNC 4.1.1 was the current version.

Accept all defaults, however, you probably do not need a VNC server on your desktop. So you can deselect VNC Server from the Select Components panel as shown in Figure 3-8.

Complete the screens and the installation process should go quickly.

### 3.2.2 Customizing RealVNC

The latest VNC protocol is version 4, which is the default with the VNC client. This version will work with the VNC servers shipped with SLES 10 or RHEL5, however, version 3.3 is required to communicate with the VNC server shipped with SLES 9.

If you need to use protocol version 3.3 for SLES 9, open the VNC client and click on the Options button as shown in the left side of Figure 3-9. Click on the Misc tab. Click the checkbox named Use only protocol version 3.3 as shown in the center of the figure. Finally, click the Load/Save tab and click Save the changes.
Your VNC client should now be ready to connect to the VNC server that your Linux systems will have.

### 3.3 3270 emulators

To access a logon session with z/VM, it is common to use a 3270 emulator that runs on Windows. Many commercial products are available. Some of the more common ones are:

- Attachmate Extra!
- Hummingbird Host Explorer
- IBM Personal Communications
- Quick3270

It is beyond the scope of this book to explain the details of configuring all the various emulators. However, it is recommended that you investigate the following settings for your emulator:

- Set the **Enter** and **Clear** function keys to be where you would expect them. On some emulators, the default Enter key action is set to the right Ctrl key of modern keyboards. Likewise the Clear key action is sometimes set to the Esc key in the upper left corner of modern keyboards or the **Pause** key in the upper right.
- Set a larger screen. Often the default number of lines in an emulator session is 24. You will probably be more productive with a 32 or 43 lines if that can easily fit in a window given your desktop display size and resolution.
- Have the session automatically reconnect after logoff. Having a new logon screen come back immediately after you log off can also save you time in the long run. This is often not the default behavior.
- Save your connection sessions. Rather than continually typing in the IP address or DNS name of the z/VM system to which you want to connect, spend a few minutes to define and save a session for each system to which you may connect, as was described for PuTTY. Then you can usually double-click the saved connection to quickly access a new 3270 session.
Installing and configuring z/VM

“Example isn't another way to teach. It is the only way to teach.”
— Albert Einstein

To complete this chapter, you must complete the majority of chapter 6 Configuring an NFS server. However, it is recommended that you start here, because there is a step when installing z/VM (instdvd) that takes over two hours or possibly much longer near the end of section 4.1.2, “Copying a vanilla z/VM system to DASD” on page 32. While that process is running, you can configure the Network File System (NFS) server. Alternatively, if you have other personnel who can work on the project, you can start both chapters at the same time on the different systems.

4.1 Installing z/VM from DVD

The section that follows assumes a first level installation of z/VM from DVD onto DASD. If you have not already done so, complete the worksheet in 2.7.1, “z/VM resources worksheet” on page 17. You will need access to the Hardware Management Console (HMC) with a user ID that has authority to go into single object operations mode.

z/VM 5.2 is shipped on tape and DVD. z/VM should install faster from tape due to faster I/O speeds, however, installing from tape may require more trips between and the HMC and the tape drive.

If you are familiar with the HMC, you can use the two page z/VM Summary for Automated Installation and Service (DVD Installation) to replace or augment the description that follows.

If you are not familiar with the HMC and z/VM, you may want to use the complete installation manual z/VM Guide for Automated Installation and Service Version 5 Release 2.0, GC24-6099. If you are installing z/VM at the second level (z/VM under z/VM) or onto SCSI disk, you will want to use the z/VM manual as the sections that follow don’t address these options.
4.1.1 Booting z/VM from DVD

This section explains how to install z/VM 5.2 from an HMC with a DVD-ROM onto 3390-3 DASD. Some words are included for installing onto 3390-9s. For alternative configurations such as installing from tape or onto SCSI disks, refer to the z/VM documentation.

1. On the Hardware Management Console, select the LPAR on which you want to install z/VM.

2. If necessary, click the racetrack buttons (two buttons that are circular arrows on the bottom right corner) to traverse to the CPC Recovery (sometimes named just Recovery) menu.

3. On the CPC Recovery menu, double-click the Integrated 3270 Console as shown at the bottom of Figure 4-1. A window entitled Integrated 3270 Console for <yourCPC> will open (on older HMC levels, the window may be entitled Personal Communications).

   **Hint:** It is convenient to use the Alt-Tab key sequence to move between the HMC window and 3270 console.

4. Place the z/VM Product Package Version 5 Release 2.0 DVD in the HMC DVD drive.

5. Get into Single Object Operations mode. To get into this mode, perform the following steps:
   a. Double-click the Groups icon in the Views Area
   b. Double-click Defined CPCs in the Groups Work Area.
   c. Select your CPC.
   d. If necessary, go around the racetrack (the buttons with circular arrows on the bottom right corner) to the CPC Recovery menu.
   e. Double-click the Single Object Operations icon. Click yes to confirm. Now a new window Primary Support Element Workplace should appear (on older HMC levels it will be a “window within a window”). A window about a certificate. If so, click OK.
   f. Double-click Groups near the top of this window.
   g. Double-click Images in the Groups Work Area.

   **Important:** If you are unable to get into Single Object Operations mode, it may be because you do not have sufficient permission. Check with the system administrator.

6. Select the LPAR that z/VM will be installed into.

7. Go around the racetrack in this window to the CPC Recovery menu. Double-click the Load from CD-ROM or Server icon when you see it (see Figure 4-2 on page 31).
8. On the Load CD-ROM or Server window as shown in Figure 4-3, the radio button Hardware Management Console CD-ROM/DVD should be selected.

9. In the same Load CD-ROM or Server window, fill in File Location with /cpdvd. This is the directory on the DVD with the z/VM 5.2 installation code.

10. Click OK.
11. Load the RAMDISK:
   a. From the Load from CD-ROM or Server panel, the software 520vm.ins should be selected as shown in Figure 4-4. Click OK.

   ![Figure 4-4 Selecting z/VM 5.2 RAMdisk system](image)

   b. From the Confirm the action window, click Yes. You should see the Load from CD-ROM, DVD or Server Progress window. The green light on the DVD drive should light up.

   c. When you see the message Completed successfully, click OK to close. This should normally take about four to six minutes.

   **Attention:** Normally, the z/VM RAMdisk (IBMVMRAM) loads in about four minutes. However, slow load times have been observed (15-18 minutes). When the green light on the DVD drive is solid, the load time will be acceptable. When it is intermittently dark more than it is green, long load times can result. This will also balloon the estimated run time of 2 - 2.5 hours of the INSTVDVD process where it may consume 5-6 hours. PMR 58909,180 was opened to address this issue.

   You should now have an in-memory z/VM 5.2 system running.

4.1.2 Copying a vanilla z/VM system to DASD

   This section describes the steps to copy z/VM to DASD.

   1. You can now get out of Single object operations mode. To do so, log off the primary SE window by closing that window.

   2. Use the Alt-Tab sequence, move to the Integrated 3270 Console window. The RAMdisk IPLs and the system comes up with the MAINT user ID logged on. You should see z/VM boot as shown in Figure 4-5:
3. Invoke the `instplan` command. This will allow you to choose associated z/VM products to install, the language to use and the type of DASD on which to install:

   `=> instplan`
4. You should see the display as shown in Figure 4-6. It is recommended that you leave the M's in the top section alone.

5. Type the letter X next to AMENG (or select your language) and 3390 Mod 3 (or the type of DASD you will use) as shown above.

6. Press F5. You should the message HCPINP8392I INSTPLAN EXEC ENDED SUCCESSFULLY.

7. Attach the DASD devices onto which z/VM will be installed defined in your planning worksheet in 2.7.2, “z/VM DASD worksheet” on page 18. In this example, the devices are a700-a704.

   ```
   ==> att <a700-a704> *
   a700-a704 ATTACHED TO MAINT
   ```

**Important:** The angle brackets, <>, in the above example should not be typed. They are used throughout the book to signify that you should replace the example value with the correct value for your site. For example, if you are installing z/VM onto DASD 1200-1204, you would type the following:

   ```
   ==> att 1200-1204 *
   ```

**Running INSTDVD**

The INSTDVD EXEC copies the z/VM system from DVD to disk.

1. Execute the `INSTDVD EXEC`:
===> instdvd

2. If you are using 3390-3s, you see a panel asking for the five volumes as shown in Figure 4-7.

![Image of INSTDVD DASD address panel]

**Figure 4-7   INSTDVD DASD address panel**

a. Enter the addresses of the five volumes that z/VM will be installed on.

b. Do *not* select the DO NOT FORMAT DASD check box on the right side of the panel.

c. Press F5 to start the installation.

3. When you see the question **DO YOU WANT TO CONTINUE?**, type Y. You should see the message **NOW FORMATTING DASD <A700>**.

4. You are asked to place the system RSU in the drive. Insert the z/VM Stacked Recommended Service Upgrade 5201 DVD into the HMC DVD-ROM drive and type 60. You should see a message of the form **DVLOAD: LOADING FILE CKD5000x IMAGE *.***. This step takes two to four minutes.

5. Finally, you should see the message **HCPIDV8329I INSTDVD EXEC ENDED SUCCESSFULLY**.

### 4.1.3 IPL the vanilla z/VM from DASD

IPL your initial z/VM system now on DASD.

1. From the HMC, **select your LPAR** by clicking it. You may have to first double-click **Groups**.

2. You should see the **CPC Recovery** (sometimes just **Recovery**) menu. Double-click the **Load** icon in the menu at the right side.
3. The Load window opens as shown in Figure 4-8. Follow these steps:
   a. Check the radio button **Clear**.
   b. Set the load address to the new system residence (520RES) volume which is **<A700>** in this example.
   c. Set the load parameter to **SYSG**. This specifies to use the Integrated 3270 console.
   d. Click **OK** to IPL.

![Load window](https://example.com/load_window.png)

*Figure 4-8  Load window*

4. When you see the **Load Task Confirmation** window, click **Yes**.

5. After 1-3 minutes you should see **Success** in the **Load Program** window. Click **OK**.

6. Use the **Alt-Tab** sequence to move back to the Integrated 3270 console window. You should see the **Standalone Program Loader** panel as shown in the following diagram.
   a. Press the **Tab** key to traverse to the **IPL Parameters** section and enter the value **cons=sysg**. This specifies to use the Integrated 3270 console.
b. Press the F10 key to continue the IPL of your z/VM system. This should take around 2-3 minutes.

7. At the Start (Warm|Force|COLD|CLEAN) prompt, enter the following:
   ```
   ==> cold drain noautolog
   ```

8. At the Change TOD clock prompt enter:
   ```
   ==> no
   ```

9. The last message should be HCPCRC8082I EREP records are accumulating for userID EREP. Disconnect from the OPERATOR user ID via the DISCONNECT command:
   ```
   ==> disc
   ```
   Press Enter to get a new logon screen.

### 4.1.4 Completing the z/VM installation

Follow these steps to complete the z/VM installation

1. On the z/VM login screen, logon as MAINT. The password is MAINT. You may receive messages HCPLMN108E or DMSACP113S about disks not linked or attached. This is not a problem. Press Enter when you see the VM Read prompt in the lower right corner.

   **Important:** When logging onto a z/VM user ID that runs CMS, you should usually press Enter at the VM READ prompt. This will result in a prompt of the form:

   ```
   Ready; T=0.01/0.01 11:14:20
   ```

2. IPL CMS and press Enter. You should see the Ready; prompt.
   ```
   ==> ipl cms
   ```
3. Run the `instvm dvd` command:

```plaintext
===> instvm dvd
... 
HCPPLD8329I POSTLOAD EXEC ENDED SUCCESSFULLY 
... 
HCPIVM8392I INSTVM ENDED SUCCESSFULLY 
```

This EXEC continues the installation process. This step should take about 4-8 minutes. The last message should be HCPIVM8392I INSTVM ENDED SUCCESSFULLY.

4. Load the recommended service. For z/VM 5.2, the service name is 5201RSU1. Run the following commands:

```plaintext
===> ipl cms
===> Press Enter
Ready;
===> acc 500 c
DMSACC724I 500 replaces C (2CC)
===> listfile * * c
5201RSU1 SERVLINK C1
===> service all 5201rsu1
```

This step should take about 3-6 minutes. The last message should be VMFSRV2760I SERVICE processing completed successfully.

5. Now IPL CMS and run the `put2prod` command. This puts the service into production:

```plaintext
===> ipl cms
===> Press Enter
Ready;
===> put2prod
```

This step should take about 2-4 minutes. The last message should be VMFP2P2760I PUT2PROD processing completed successfully.

A return code of 0 is ideal. You may get a return code of 4 and the message:

VMFP2P2760I PUT2PROD process completed with warnings.

In general on z/VM, a return code of 4 is acceptable. That means that only warnings were issued. A return code of 8 or greater generally means that errors were encountered.

6. Enter the following command to shutdown and rel IPL your system:

```plaintext
===> shutdown reipl 
SYSTEM SHUTDOWN STARTED 
```

7. You will lose your 3270 session. The system should come back in about 2-4 minutes. After it comes back, the last message should be "Press enter or clear key to continue".

Press Enter and you should see a z/VM logon screen.

Congratulations! You should now have a vanilla z/VM system installed.

### 4.2 Configuring TCP/IP

It is recommended that you initially configure TCP/IP via the `IPWIZARD` command which is generally used just once. After `IPWIZARD` creates the initial configuration files, they are typically maintained manually.

From the z/VM logon panel Logon to MAINT. The default password for all z/VM user IDs is the same as the user ID. So enter a password of `maint` which will not be echoed on the screen.
USERID   ==> maint
PASSWORD  ==> 

After entering the user ID and password, press Enter when the status area in the lower right reads “VM READ”.

### 4.2.1 Use the IPWIZARD tool

The IPWIZARD command is on the MAINT 193 disk. You will need to access it file mode G via the ACCESS command so you will pick up IPWIZARD from that minidisk.

1. Access the MAINT 193 disk:

   ```
   => acc 193 g
   ```

2. Invoke IPWIZARD:

   ```
   => ipwizard
   ```

### z/VM TCP/IP Configuration Wizard

The items that follow describe your z/VM host

**User ID of VM TCP/IP Stack Virtual Machine:** TCPIP

**Host Name:** <lat120>

**Domain Name:** <pbm.ihost.com>

**Gateway IP Address:** <129.40.178.254>

**DNS Addresses:**
1) <129.40.106.1>
2) 
3) 

### General Interface Configuration Panel

**Interface Name:** ETH0

**Device Number:** <3000>

**IP Address:** <129.40.178.120>

**Subnet Mask:** <255.255.255.0>

**Interface Type (Select one):**

- QDIO
- LCS
- HiperSockets
- CLAW
- CTC
4. An *Interface Name* of **ETH0** is arbitrary but recommended. The *Device Number* will be the starting address of the OSA triplet that the z/VM stack will use. The *IP address* which must be routed to the OSA card will become the TCP/IP address of the z/VM system. The *Interface Type* will typically be **QDIO** with modern OSA devices. When completed, press **F8**.

```
*** QDIO Interface Configuration Panel ***

Network Type (Select one):
  x Ethernet     _ Token Ring

Port Name (optional):   ________

Router Type (Select one):
  _ Primary       _ Secondary    x None

Maximum Transmission Unit (MTU) size:  **1500**
```

5. In general a value for the *Port Name* is no longer necessary and a *Router Type of None* is recommended. Press **F5** to complete the wizard.

```
DTCIPW2508I DTCIPWIZ EXEC is attempting to create the necessary DTCIPW2508I configuration files
```

6. Enter **1** to restart the TCP/IP stack:

```
The TCP/IP stack (TCPIP) must be restarted as part of this procedure
Would you like to restart and continue?
Enter 0 (No), 1 (Yes) 1
USER DSC   LOGOFF AS  TCPIP    USERS = 2     FORCED BY MAINT

Successfully PINGed Interface (129.40.178.120)
Successfully PINGed Gateway (129.40.178.254)
Ping Level 520: Pinging host 129.40.106.1.
   Enter 'HX' followed by 'BEGIN' to interrupt.

**Important:** If the DNS server cannot be pinged, enter **1** to try it again.

```
PING: Ping #1 timed out
Not all of the PINGS were successful. Would you like to try them again?
Enter 0 (No), 1 (Yes) 1

*** QDIO Interface Configuration Panel ***

Network Type (Select one):
  x Ethernet     _ Token Ring

Port Name (optional):   ________

Router Type (Select one):
  _ Primary       _ Secondary    x None

Maximum Transmission Unit (MTU) size:  **1500**
```

```
Successfully PINGed DNS (129.40.106.1)
DTCIPW2519I Configuration complete; connectivity has been verified
```
Chapter 4. Installing and configuring z/VM

4.3 Configuring the XEDIT profile

Logon to MAINT if you are not already.

The **XEDIT** command looks for the file **XEDIT PROFILE** configuration file when it is invoked. Many z/VM user IDs do not have such a file, so all **XEDIT** default values are in effect. The **MAINT 191 (A)** disk has a **PROFILE XEDIT** so when you are editing files on **MAINT**, the values in this profile are usually in effect.

One setting that can be dangerous, especially if you use F12 to retrieve commands, is that PF12 is set to the **FILE** subcommand. Sometimes you may not want to save your changes with the stroke of one key. It is recommended that you set PF12 to the **?** subcommand which has the effect of a retrieve key:

```
=> copy profile xedit a profile xediorig a (oldd
=> x profile xedit a
```

**Before:**

```
SET PF12 FILE
```

**After:**

```
SET PF12 ?
```

Save your changes with the **FILE** subcommand.
4.4 Customizing the SYSTEM CONFIG file

The first configuration file read when z/VM IPLs is the SYSTEM CONFIG file. The following changes are recommended:

- Change the system name
- Increase retrieve key capacity
- Allow virtual disks (VDISKs) to be created
- Turn off the Disconnect Timeout. This will prevent idle disconnected users from being forced off the system
- Define a virtual switch (VSWITCH) that will be used for Linux networking

To make these changes, perform the following steps:

1. To edit the SYSTEM CONFIG file, the MAINT CF1 minidisk must be released as a CP disk via the CPRELEASE command. The CP disks are queried via the QUERY CPDISK command. Note the MAINT CF1 disk is accessed as CP disk A before it is released but not after.

   => q cpdisk

<table>
<thead>
<tr>
<th>Label</th>
<th>Userid</th>
<th>Vdev</th>
<th>Mode</th>
<th>Stat</th>
<th>Vol-ID</th>
<th>Rdev</th>
<th>Type</th>
<th>StartLoc</th>
<th>EndLoc</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNTCF1</td>
<td>MAINT</td>
<td>CF1</td>
<td>R/O</td>
<td>MVA740</td>
<td>A740</td>
<td>CKD</td>
<td></td>
<td>39</td>
<td>158</td>
</tr>
<tr>
<td>MNTCF2</td>
<td>MAINT</td>
<td>CF2</td>
<td>R/O</td>
<td>MVA740</td>
<td>A740</td>
<td>CKD</td>
<td></td>
<td>159</td>
<td>278</td>
</tr>
<tr>
<td>MNTCF3</td>
<td>MAINT</td>
<td>CF3</td>
<td>R/O</td>
<td>MVA740</td>
<td>A740</td>
<td>CKD</td>
<td></td>
<td>279</td>
<td>398</td>
</tr>
</tbody>
</table>

   => cprel a

   CPRELEASE request for disk A scheduled.
   HCPZAC6730I CPRELEASE request for disk A completed.

   => q cpdisk

<table>
<thead>
<tr>
<th>Label</th>
<th>Userid</th>
<th>Vdev</th>
<th>Mode</th>
<th>Stat</th>
<th>Vol-ID</th>
<th>Rdev</th>
<th>Type</th>
<th>StartLoc</th>
<th>EndLoc</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNTCF2</td>
<td>MAINT</td>
<td>CF2</td>
<td>R/O</td>
<td>MVA740</td>
<td>A740</td>
<td>CKD</td>
<td></td>
<td>159</td>
<td>278</td>
</tr>
<tr>
<td>MNTCF3</td>
<td>MAINT</td>
<td>CF3</td>
<td>R/O</td>
<td>MVA740</td>
<td>A740</td>
<td>CKD</td>
<td></td>
<td>279</td>
<td>398</td>
</tr>
</tbody>
</table>

2. Once it is released you are able to access the MAINT CF1 disk read-write. Use the LINK command with multi-read (MR) parameter and ACCESS command to get read-write access to the minidisk.

   => link * cf1 cf1 mr
   => acc cf1 f

3. Now the MAINT CF1 disk is accessed read-write as your F disk. First make a backup copy of the vanilla SYSTEM CONFIG file using the COPYFILE command with the OLDDATE parameter so the file’s timestamp is not modified, then edit the original copy:

   => copy system config f system config f (oldd
   => x system config f

4. The system name is set to ZVMV5R20 by default in the System_Identifier_Default statement. You can search for it using the / XEDIT subcommand:

   ======/System_Identifier_D

   Modify this to the new name of your system:

   System_Identifier_Default <LINUXVM52>

5. Next look for the Features statement. You can search for it again or you can use F8 to page down. The following changes and additions are recommended:

   - Increase the number of commands that can be retrieved from 20 to 99.
   - Set the Disconnect_Timeout to off so disconnected users do not get forced off.
   - Allow unlimited VDISKs to be created by users by changing Userlim to infinite and by adding the Syslim infinite clause:

     Features,
     Disable ,    /* Disable the following features */
     Set_Privclass , /* Disallow SET PRIVCLASS command */
     Auto_Warm_IPL , /* Prompt at IPL always */
Clear_TDisk ,               /* Don't clear TDisks at IPL time */
Retrieve ,                      /* Retrieve options               */
Default  99 ,               /* Default.... default is 20 */
Maximum  255 ,               /* Maximum.... default is 255 */
MaxUsers noLimit ,              /* No limit on number of users */
Passwords_on_Cmds ,             /* What commands allow passwords? */
Autolog yes ,                  /* ... AUTOLOG does */
Link yes ,                     /* ... LINK does */
Logon yes ,                    /* ... and LOGON does, too */
Disconnect_Timeout off ,       /* Don't force disconnected users */
Vdisk ,                        /* Allow VDISKS for Linux swaps */
Syslim  infinite ,             /* Allow VDISKS for Linux swaps */
Userlim  infinite

6. Define a VSWITCH:

Use the BOTTOM subcommand to go to the bottom of the file. Add some lines (you can use the XEDIT add subcommand a3). Define a VSWITCH and set the MAC address prefix. If you have multiple z/VM systems, each should have a unique prefix. Modify the two starting addresses of the OSA triplets (3004 and 3008 in this example) to those you specified in 2.7.1, “z/VM resources worksheet” on page 17.

====> bot
====> a3
/* define vswitch named vsw1 and set MAC address prefixes to 02-00-01 */
define vswitch vsw1 rdev <3004> <3008>
vmlan macprefix 020001

7. Save your changes with the XEDIT FILE subcommand:

====> file

8. Test your changes with the CPSYNTAX command which is on the MAINT 193 disk:

==> acc 193 g
==> cpsyntax system config f
CONFIGURATION FILE PROCESSING COMPLETE -- NO ERRORS ENCOUNTERED.

Pay attention to the output. If you get any syntax errors, fix them before proceeding.

9. Release and detach the MAINT CF1 disk with the RELEASE command and DETACH parameter. Then put it back online with the CPACCESS command:

==> rel f (det
DASD OCF1 DETACHED
==> cpacc * cfi a
CPACCESS request for mode A scheduled.
HCPZAC6732I CPACCESS request for MAINT's OCF1 in mode A completed.
==> q cpdisk

<table>
<thead>
<tr>
<th>Label</th>
<th>Userid</th>
<th>Vdev</th>
<th>Mode</th>
<th>Stat</th>
<th>Vol-ID</th>
<th>Rdev</th>
<th>Type</th>
<th>StartLoc</th>
<th>EndLoc</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNTCF1</td>
<td>MAINT</td>
<td>OCFI</td>
<td>A</td>
<td>R/O</td>
<td>MVA740</td>
<td>A740</td>
<td>CKD</td>
<td>39</td>
<td>158</td>
</tr>
<tr>
<td>MNTCF2</td>
<td>MAINT</td>
<td>OCF2</td>
<td>B</td>
<td>R/O</td>
<td>MVA740</td>
<td>A740</td>
<td>CKD</td>
<td>159</td>
<td>278</td>
</tr>
<tr>
<td>MNTCF3</td>
<td>MAINT</td>
<td>OCF3</td>
<td>C</td>
<td>R/O</td>
<td>MVA740</td>
<td>A740</td>
<td>CKD</td>
<td>279</td>
<td>398</td>
</tr>
</tbody>
</table>

Note that all three CP disks are now accessed.

4.5 Configuring TCP/IP to start at IPL time

Configure the TCP/IP service machine to be started when z/VM IPLs. This is commonly accomplished from AUTOLOG1's PROFILE EXEC. If the noautolog parameter is not specified when z/VM is IPLed, the AUTOLOG1 virtual machine is started. Because this virtual machine IPLs CMS, the PROFILE EXEC that is found on its A disk is run. This is analogous to the /etc/profile file on Linux and the autoexec.bat on DOS systems.
1. Logoff of MAINT.
   ==> log

2. You should see a new logon panel. Logon to AUTOLOG1. Again the password is the same as the user ID.

3. At the VM READ prompt enter the command ACCESS (NOPROF) so that the PROFILE EXEC is not run.

   LOGON AUTOLOG1
   z/VM Version 5 Release 2.0, Service Level 0501 (64-bit),
   built on IBM Virtualization Technology
   There is no logmsg data
   FILES:   NO RDR,   NO PRT,   NO PUN
   LOGON AT 13:30:12 EST THURSDAY 01/19/06
   DMSIND2015W Unable to access the Y-disk. Filemode Y (19E) not accessed
   z/VM V5.2.0    2005-12-22 09:36
   acc (noprof

4. Copy the PROFILE XEDIT from the MAINT 191 disk so XEDIT sessions will have a common interface among user IDs.
   a. Use theVMLINK command to both link to the disk read-only and to access it as the highest available file mode. The default read password is read:
      ==> vmlink maint 191
      ENTER READ PASSWORD:
      read
      DMSVML2060I MAINT 191 linked as 0120 file mode Z
   b. Copy the PROFILE XEDIT to your A disk:
      ==> copy profile xedit z = a

5. Make a backup copy of the PROFILE EXEC and edit it:
   ==> copy profile exec a = execorig =
   ==> x profile exec

6. You should see the text in the top half of the following example. Modify it as follows.
   a. The z/VM Shared File System (SFS), is not required to run Linux so you can safely delete the three lines that XAUTOLOG the user IDs VMSERVS, VMSERVU and VMSERVR.
   b. You can also safely delete the Address Command line.
   c. Add a line to start the TCPIP user ID via the XAUTOLOG command and keep two statements that start the VSWITCH controllers.
   d. Add a line to logoff of AUTOLOG1 when the EXEC is complete. There is no need to keep that virtual machine running as its sole purpose is to run the PROFILE EXEC.

Before:
/*********************/
/* Autolog1 Profile Exec */
/*********************/

Address Command
'CP XAUTOLOG VMSERVS'
'CP XAUTOLOG VMSERVU'
'CP XAUTOLOG VMSERVR'
'CP XAUTOLOG DTCVSW1'
'CP XAUTOLOG DTCVSW2'

After:
7. Save your changes with the FILE subcommand and logoff of AUTOLOG1:

   ===> file
   ==> log

When your z/VM system IPLs, the TCP/IP stack should now come up automatically (as long as you do not specify the notautolog parameter at IPL time).

### 4.5.1 Renaming the TCPIP configuration file

It is recommended that you change the name of the main TCPIP configuration file from PROFILE TCPIP to `<system_ID> TCPIP`, where `<system_ID>` is the name of your new z/VM system. This is to avoid the possibility that the PROFILE TCPIP file will be overwritten when applying maintenance.

Logon to TCPMAINT. The PROFILE TCPIP file is on the TCPMAINT 198 disk which is accessed as the D disk.

Make a backup copy the original PROFILE TCPIP, then rename it to `<SYSTEM_ID> TCPIP` (where `<SYSTEM_ID>` is LNXVM52 in this example). When the TCPIP service machine starts, it will search for this file before the file PROFILE TCPIP.

   ==> copy profile tcpip d = tcpiorig = (oldd
   ==> rename profile tcpip d <lnxvm52> = =

You have now renamed your TCP/IP profile.

### 4.5.2 Copy the PROFILE XEDIT file

Again copy the PROFILE XEDIT from the MAINT 191 disk so XEDIT sessions will have a common interface among user IDs.

a. Use theVMLINK command to both link to the disk read-only and to access it as the highest available file mode. The default read password is read:

   ==> vmlink maint 191
   ENTER READ PASSWORD:
   read
   DMSVML2060I MAINT 191 linked as 0120 file mode Z

b. Copy the PROFILE XEDIT to your A disk:

   ==> copy profile xedit z = = a

Now, XEDIT sessions on TCPMAINT will have the same configuration as on MAINT.

### 4.5.3 Configuring the FTP server

It is recommend that you turn on the FTP server. To do so, edit the newly renamed configuration file and add an AUTOLOG statement near the top of the file with FTPSERVE as the only entry. In the PORT statement, remove the semicolons to uncomment the lines with FTPSERVE on them (ports 20 and 21). These changes will cause the FTP server to start when TCPIP is started. The important lines before the file is edited and after are shown:
==# x <lnxvm52> tcpip d

Before:

; --------------------------
OBEY
OPERATOR TCPMAINT MAINT MPROUTE ROUTED DHCPD REXECD SNMPD SNMPQE
ENDOBEY

PORT
  ; 20 TCP FTPSERVE NOAUTOLOG ; FTP Server
  ; 21 TCP FTPSERVE            ; FTP Server
  ; 23 TCP INTCLIEN            ; TELNET Server
  ; 25 TCP SMTP                ; SMTP Server
  ...

After:

; --------------------------
OBEY
OPERATOR TCPMAINT MAINT MPROUTE ROUTED DHCPD REXECD SNMPD SNMPQE
ENDOBEY

AUTOLOG
  FTPSERVE 0
ENDAUTOLOG

PORT
  20 TCP FTPSERVE NOAUTOLOG ; FTP Server
  21 TCP FTPSERVE            ; FTP Server
  23 TCP INTCLIEN            ; TELNET Server
  25 TCP SMTP                ; SMTP Server
  ...

====> file

Save your changes with the FILE subcommand. You could continue to configure the system, but at this time it is recommended that you test your changes by shutting down and reIPLing the system.

4.5.4 Shutting down and reIPLing the system

It is now time to shutdown and reIPL the system. You will also want to be able to shutdown and reIPL z/VM without having to access the HMC. Often, the HMC will be logged off and thus the Integrated 3270 console (SYSG) will not be available. Because of these factors it is useful to use the System Console (SYSC - which has a title of Operating System Messages on the HMC) in order to shut down z/VM and reIPL it without needing to use the console. This console is always accessible whether you are logged on to the HMC or not. z/VM messages during both the shutdown and reIPL process will be written to the system console, but often you will be able to ignore them - you just want your system back in a few minutes over the network.

Pass the parameter IPLPARMS CONS=SYSC to the SHUTDOWN REIPL command:

  ==> shutdown reipl iplparms cons=sysc

You will lose your session, but it should come back in a few minutes as described above. You may want to watch the system console as z/VM shuts down and comes back up. When your system is back up, perform the following commands

1. Start a 3270 session and Logon as MAINT. This shows that there is TCP/IP access to z/VM.
2. Query the new VSWITCH:

```bash
q vswitch
```

```
VSWITCH SYSTEM VSW1 Type: VSWITCH Connected: 0 Maxconn: INFINITE
PERSISTENT RESTRICTED NONROUTER Accounting: OFF
VLAN Unaware
State: Ready
IPTimeout: 5 QueueStorage: 8
Portname: UNASSIGNED RDEV: 3004 Controller: DTCVSW1 VDEV: 3004
Portname: UNASSIGNED RDEV: 3008 Controller: DTCVSW2 VDEV: 3008 BACKUP
```

You should see that the VSWITCH exists and that there are two built-in VSWITCH controllers, DTCVSW1 and DTCVSW2. Before z/VM 5.2, these user IDs had to be created manually.

3. Use the `QUERY VDISK` and `QUERY RETRIEVE` commands to see the changes made to the Features statement in the SYSTEM CONFIG file:

```bash
q retrieve
99 buffers available. Maximum of 255 buffers may be selected.
```

```bash
q vdisk userlim
VDISK USER LIMIT IS INFINITE
```

```bash
q vdisk syslim
VDISK SYSTEM LIMIT IS INFINITE, 0 BLK IN USE
```

This shows that the changes to the SYSTEM CONFIG file have taken effect.

### 4.6 Adding paging volumes

The z/VM operating system resides on the first three CP volumes (or one volume if installing onto 3390-9s). z/VM 5.2 now also installs with one full paging volume and one full spool volume. A single spool volume is probably adequate for Linux needs, however, a single paging volume is probably not.

It is recommended that you add five paging volumes so you will have a total of six (or one more 3390-9 if installing onto 3390-9s which will give the same total of 20034 cylinders). If you do not have sufficient DASD, this number can be reduced. Having adequate paging space will give you plenty of headroom to add more Linux virtual machines. A rule of thumb for the amount of paging space is to have twice as much as the total of all memory for all running Linux user IDs combined.

#### 4.6.1 Formatting the paging volumes

Before adding paging volumes to the system, the DASD volumes to be used for minidisk space (PERM) and paging space (PAGE) must be formatted. Normally this is done one volume at a time via the `CPFMTXA` command. If you have just a few volumes, that is fine, but when you have many volumes to format, the process of running `CPFMTXA` can become time consuming and tedious which can lead to errors.

Therefore, a REXX EXEC named `CPFORMAT` has been provided to allow you to format many volumes with a single command. The source code for this EXEC is in the section B.1.1, "The CPFORMAT EXEC" on page 233. It is a wrapper around `CPFMTXA`. To use this EXEC, each DASD to be formatted must first be attached with the virtual device address the same real device address (using `ATTACH <realDev> *`).

**Note:** This EXEC will label the volumes according to the convention described in 2.2.1, “Volume labeling convention” on page 11. If you want different volume labels, you can use
the CPHTX A command and manually specify each volume label, or you can modify the REXX EXEC.

Important: At this point, you will need access to the NFS server described in chapter 6, in order to get the files CPFORMAT EXEC. If you did not complete that chapter, do it now.

Getting the CPFORMAT EXEC to z/VM

Logoff of MAINT so you will be able to get the MAINT 191 disk in read-write mode via FTP.

Start an SSH (putty) session to the NFS server and change to the vm/ directory which was created when you untarred the files associated with this book. Verify that the file CPFORMAT.EXEC exists. Note that the directory name will be one of the following two depending on the Linux distribution you are working with:

```
# cd /nfs/virt-cookbook-S10/vm // if you are working with SLES 10
# cd /nfs/virt-cookbook-RH5/vm // if you are working with RHEL 5
# ls CPFORMAT*
```

CPFORMAT.EXEC

Now start an FTP session to z/VM. If you get a reply from the FTP server it shows that you correctly configured it on the z/VM TCPMAINT user ID.

```
# ftp <129.40.178.120>
220-FTPSERVE IBM VM Level 520 at LAT120.PBM.IHOST.COM, 14:53:44 EST WEDNESDAY 2004-12-08
Name (129.40.178.120:root): maint
Password: maint
...
ftp> put CPFORMAT.EXEC
...
ftp> quit
```

Using the CPFORMAT EXEC

Log back into MAINT. You should now have access to the CPFORMAT EXEC. You can get brief help on CPFORMAT by using a parameter of ?:

```
=>> cpformat ?
```

Synopsis:

Format one or a range of DASD as page, perm, spool or temp disk space
The label written to each DASD is V<t><xxxx> where:
  <t> is type - P (page), M (perm), S (spool) or T (Temp disk)
  <xxxx> is the 4 digit address

Syntax is:

```
>>--CPFORMAT--.-rdev--------------.--AS---+-PERM-+---------><
| <---------------< |       |-SPOL-|
'-rdev1-rdev2-------'       '-TDSK-
```

The following example shows how to attach five 3390-3 volumes and use CPFORMAT to format them as paging space. Refer to the planning work sheets that you filled out in 2.7.2, “z/VM DASD worksheet” on page 18:
Rather than using five consecutive DASD addresses, you may consider using DASD from different address ranges in an attempt to locate the paging volumes on different ranks in your disk array. This should enable z/VM to page more efficiently:

```plaintext
===> att <a705-a709> *
A705-A709 ATTACHED TO MAINT
===> cpformat <a705-a709> as page
```

Format the following DASD:

```
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A705 MAINT    A705 3390  MPA705 A705          0       3339
MAINT    A706 MAINT    A706 3390  MPA706 A706          0       3339
MAINT    A707 MAINT    A707 3390  MPA707 A707          0       3339
MAINT    A708 MAINT    A708 3390  MPA708 A708  0       3339
MAINT    A709 MAINT    A709 3390  MPA709 A709          0       3339
```

WARNING - this will destroy data!
ARE YOU SURE you want to format the DASD as PAGE space (y/n)?
y

... DASD status after:

```
TargetID Tdev OwnerID  Odev Dtype Vol-ID Rdev   StartLoc       Size
MAINT    A705 MAINT    A705 3390  MPA705 A705          0       3339
MAINT    A706 MAINT    A706 3390  MPA706 A706          0       3339
MAINT    A707 MAINT    A707 3390  MPA707 A707          0       3339
MAINT    A708 MAINT    A708 3390  MPA708 A708  0       3339
MAINT    A709 MAINT    A709 3390  MPA709 A709          0       3339
```

This formatting job should run for about 10-50 minutes depending on many factors. But don’t take a break yet! You can format more volumes for PERM (minidisk) space in the next section.

### 4.6.2 Formatting DASD for minidisks

You could wait until `CPFORMAT` of the five paging volumes completes on MAINT, and then format more volumes for PERM or minidisk space. However, you can also get more format jobs going by using a different user ID.

1. Start a new 3270 session and `logon as SYSMAINT`. Press `Enter` when you get the VM READ prompt:

```
LOGON SYSMAINT
```

z/VM Version 5 Release 2.0, Service Level 0601 (64-bit),
DMSACP113S A(191) not attached or invalid device address  
DMSACP723I D (192) R/O

2. Link to the MAINT 191 disk read-only to pick up the CPFORMAT EXEC. This can be done with the VMLINK command (VMLINK performs the LINK and ACCESS commands, with a read-only link and accessing the highest free file mode letter).

  ==>> vmlink maint 191
  DMSVML2060I MAINT 191 linked as 0192 file mode D

3. Attach the seven volumes that you will use for the controller user ID and the eighth that will be used for the first Linux clone. In this example it is the DASD at addresses A70A-A711.

   Note: For RHEL5, the example would use DASD addresses A728-A72E.

  ==>> att <a70a-a711> *
  A70A-A711 ATTACHED TO MAINT
  ==>> cpformat <a70a-a711> as perm

Format the following DASD:

  TargetID Tdev OwnerID Odev Dtype Vol-ID Rdev StartLoc Size
  MAINT    A70A MAINT    A70A 3390  MMA70A A70A  0       3339
  MAINT    A70B MAINT    A70B 3390  MMA70B A70B  0       3339
  MAINT    A70C MAINT    A70C 3390  MMA70C A70C  0       3339
  MAINT    A70D MAINT    A70D 3390  MMA70D A70D  0       3339
  MAINT    A70E MAINT    A70E 3390  MMA70E A70E  0       3339
  MAINT    A70F MAINT    A70F 3390  MMA70F A70F  0       3339
  MAINT    A710 MAINT    A710 3390  MMA710 A710  0       3339
  MAINT    A711 MAINT    A711 3390  MMA711 A711  0       3339

WARNING - this will destroy data! 
ARE YOU SURE you want to format the DASD as PERM space (y/n)?

  y ...

   DASD status after:

  TargetID Tdev OwnerID Odev Dtype Vol-ID Rdev StartLoc Size
  MAINT    A70A MAINT    A70A 3390  MMA70A A70A  0       3339
  MAINT    A70B MAINT    A70B 3390  MMA70B A70B  0       3339
  MAINT    A70C MAINT    A70C 3390  MMA70C A70C  0       3339
  MAINT    A70D MAINT    A70D 3390  MMA70D A70D  0       3339
  MAINT    A70E MAINT    A70E 3390  MMA70E A70E  0       3339
  MAINT    A70F MAINT    A70F 3390  MMA70F A70F  0       3339
  MAINT    A710 MAINT    A710 3390  MMA710 A710  0       3339
  MAINT    A711 MAINT    A711 3390  MMA711 A711  0       3339

Now you can take a break! You should now have page volumes being formatted on MAINT and PERM or minidisk volumes being formatted on SYSMAINT.
When completed, you should have eight newly formatted volumes that can be used as minidisks.

4.6.3 Updating the SYSTEM CONFIG file

Follow these steps to update the SYSTEM CONFIG file:

1. Now that the PAGE and PERM volumes are ready for use, they must be added to the SYSTEM CONFIG file so z/VM knows about them. The following example uses the same steps to access the MAINT CF1 disk read-write that you used earlier:

   ```
   ===> q cpdisk
   Label  Userid   Vdev Mode Stat Vol-ID Rdev Type   StartLoc     EndLoc
   MNTCF1 MAINT    0CF1  A   R/O  520RES 0200 CKD          39         83
   MNTCF2 MAINT    0CF2  B   R/O  520RES 0200 CKD          84        128
   MNTCF3 MAINT    0CF3  C   R/O  520RES 0200 CKD         129        188
   ===> cprel a
   CPRELEASE request for disk A scheduled.
   HCPZAC6730I CPRELEASE request for disk A completed.
   ===> link * cf1 cf1 mr
   ===> acc cf1 f
   ```

   It is good to remember this sequence of steps.

2. Edit the SYSTEM CONFIG file and specify each of the new page volumes (PAGE) by name as CP_Owned. When you system IPLs it will pick up these as paging volumes.

   ```
   ===> x system config f
   ...
   /*********************************************************************************/
   /*                     CP_Owned Volume Statements */
   /*********************************************************************************/
   CP_Owned   Slot   1  520RES
   CP_Owned   Slot   2  520W01
   CP_Owned   Slot   3  520W02
   CP_Owned   Slot   4  520SPL
   CP_Owned   Slot   5  520PAG
   CP_Owned   Slot   6  <MPA705>
   CP_Owned   Slot   7  <MPA706>
   CP_Owned   Slot   8  <MPA707>
   CP_Owned   Slot   9  <MPA708>
   CP_Owned   Slot   10  <MPA709>
   CP_Owned   Slot   11  RESERVED
   CP_Owned   Slot   12  RESERVED
   CP_Owned   Slot   13  RESERVED
   ...
   /*********************************************************************************/
   /*                          User_Volume_List                          */
   /* These statements are not active at the present time. They are     */
   /* examples, and can be activated by removing the comment delimeters */
   /*********************************************************************************/
   User_Volume_List USRP01
   User_Volume_List USRP02
   ```

3. Move down to the User_Volume_List section. User volumes (PERM) can be specified individually with the User_Volume_List statement, or with wild cards via the User_Volume_Include statement. If you are using the labelling convention enforced by the CPFORMAT EXEC, then add the following single line to include all PERM space as volume labels all begin with "MM" (the labeling convention used the by the CPFORMAT EXEC):

   ```
   User_Volume_Include MM*
   User_Volume_List USRP01
   User_Volume_List USRP02
   ```
4. Save your changes with the `FILE` subcommand. Verify the integrity of the changes with the `CPSYNTAX` command then put the MAINT CF1 disk back online. The following example shows how you did this previously:

```plaintext
=> acc 193 g
=> cpsyntax system config f
CONFIGURATION FILE PROCESSING COMPLETE -- NO ERRORS ENCOUNTERED.
=> rel f (det
DASD OCF1 DETACHED
=> cpacc * cfi a
CPACCESS request for mode A scheduled.
HCPZAC6732I CPACCESS request for MAINT's OCF1 in mode A completed.
=> q cpdisk
<table>
<thead>
<tr>
<th>Label</th>
<th>Userid</th>
<th>Vdev</th>
<th>Mode</th>
<th>Stat</th>
<th>Vol-ID</th>
<th>Rdev</th>
<th>Type</th>
<th>StartLoc</th>
<th>EndLoc</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNTCF1</td>
<td>MAINT</td>
<td>0CF1</td>
<td>A</td>
<td>R/O</td>
<td>520RES</td>
<td>0200</td>
<td>CKD</td>
<td>39</td>
<td>83</td>
</tr>
<tr>
<td>MNTCF2</td>
<td>MAINT</td>
<td>0CF2</td>
<td>B</td>
<td>R/O</td>
<td>520RES</td>
<td>0200</td>
<td>CKD</td>
<td>84</td>
<td>128</td>
</tr>
<tr>
<td>MNTCF3</td>
<td>MAINT</td>
<td>0CF3</td>
<td>C</td>
<td>R/O</td>
<td>520RES</td>
<td>0200</td>
<td>CKD</td>
<td>129</td>
<td>188</td>
</tr>
</tbody>
</table>
```

4.6.4 Testing the changes

It is recommended that you again shutdown and reIPL to test the changes. Before you shut down, note that you have only one page volume (520PAG) via the `QUERY ALLOC PAGE` command. Your output should look similar to the following:

```plaintext
=> q alloc page
<table>
<thead>
<tr>
<th>EXTENT</th>
<th>EXTENT</th>
<th>TOTAL</th>
<th>PAGES</th>
<th>HIGH</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOLID</td>
<td>RDEV</td>
<td>START</td>
<td>END</td>
<td>PAGES IN USE</td>
<td>PAGE USED</td>
</tr>
<tr>
<td>-------</td>
<td>--------</td>
<td>-------</td>
<td>-------</td>
<td>----------------</td>
<td>-----------</td>
</tr>
<tr>
<td>520PAG</td>
<td>A702</td>
<td>1</td>
<td>3338</td>
<td>600840</td>
<td>12 1%</td>
</tr>
<tr>
<td>SUMMARY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USABLE</td>
<td>601020</td>
<td>12</td>
<td>1%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Now shut the system down again with the command `SHUTDOWN REIPL IPLPARMS CONS=SYSC`. This is analogous to the Linux `reboot` command in that the system attempts to come back up after it shuts down. If you are connected via a 3270 emulator, you will lose your session, but if all goes well, your system will be available again in a couple of minutes.

```plaintext
=> shutdown reipl iplparsms cons=sysc
```

After the system comes back, `logon as MAINT` and look at the page space again. If you are using 3390-3s, you should see output similar to the following:

```plaintext
=> q alloc page
<table>
<thead>
<tr>
<th>EXTENT</th>
<th>EXTENT</th>
<th>TOTAL</th>
<th>PAGES</th>
<th>HIGH</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOLID</td>
<td>RDEV</td>
<td>START</td>
<td>END</td>
<td>PAGES IN USE</td>
<td>PAGE USED</td>
</tr>
<tr>
<td>-------</td>
<td>--------</td>
<td>-------</td>
<td>-------</td>
<td>----------------</td>
<td>-----------</td>
</tr>
<tr>
<td>520PAG</td>
<td>A702</td>
<td>1</td>
<td>3338</td>
<td>600840</td>
<td>0 0%</td>
</tr>
<tr>
<td>MPA705</td>
<td>A705</td>
<td>0</td>
<td>3338</td>
<td>601020</td>
<td>0 0%</td>
</tr>
<tr>
<td>MPA706</td>
<td>A706</td>
<td>0</td>
<td>3338</td>
<td>601020</td>
<td>0 0%</td>
</tr>
<tr>
<td>MPA707</td>
<td>A707</td>
<td>0</td>
<td>3338</td>
<td>601020</td>
<td>0 0%</td>
</tr>
<tr>
<td>MPA708</td>
<td>A708</td>
<td>0</td>
<td>3338</td>
<td>601020</td>
<td>0 0%</td>
</tr>
<tr>
<td>MPA709</td>
<td>A709</td>
<td>0</td>
<td>3338</td>
<td>601020</td>
<td>12 1%</td>
</tr>
<tr>
<td>SUMMARY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USABLE</td>
<td>3521K</td>
<td>12</td>
<td>1%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

The output shows there are six paging volumes constituting 3521 KB pages, or about 14 GB of page space (there are 4KB/page).
4.7 Creating a user ID for common files

Now it is time to define your first z/VM user ID, LNXMAINT. It will be used to store files that will be shared by Linux user IDs. Before starting, make a copy of the original USER DIRECT file:

```bash
=> copy user direct c = direorig = (oldd
```

4.7.1 Define the user in the USER DIRECT file

A small 20 cylinder minidisk is allocated at virtual address 191 and a larger 300 cylinder minidisk (approximately 225MB), to be shared by many guests, is defined at virtual address 192. Use the next free DASD designated as PERM space on your worksheet (2.7.2, “z/VM DASD worksheet” on page 18). Cylinder 0 should always be reserved for the label therefore you should start minidisks at cylinder 1.

1. Edit the USER DIRECT file and add the following user ID definition to the bottom of the file:

   - **User ID**: LNXMAINT, same password, default size of 64MB, with class B, E and G privileges
   - **Include**: the profile named TCPCMSU (defined earlier in the USER DIRECT file)
   - **Link**: to the TCPMAINT 592 disk read-only for access to FTP and other TCP/IP commands
   - **Define**: a 191 minidisk of size 20 cylinders from volume MMA70C
   - **Define**: 192 minidisk of size 300 cylinders (approximately 225MB) from volume MMA70C with the special read password of ALL which allows read access from any user ID without a disk password

   1. User ID LNXMAINT, same password, default size of 64MB, with class B, E and G privileges
   2. Include the profile named TCPCMSU (defined earlier in the USER DIRECT file)
   3. Link to the TCPMAINT 592 disk read-only for access to FTP and other TCP/IP commands
   4. Define a 191 minidisk of size 20 cylinders from volume MMA70C
   5. Define 192 minidisk of size 300 cylinders (approximately 225MB) from volume MMA70C with the special read password of ALL which allows read access from any user ID without a disk password
   6. An empty comment line for better readability.

2. Whenever an MDISK statement is added or modified in the USER DIRECT file you should always check for overlapping cylinders and gaps (gaps will only leave empty disk space, however z/VM will allow you to shoot yourself in the foot by defining multiple minidisks over the same disk space). This is done with the DISKMAP command:

   ```bash
   => diskmap user
   ```

   The minidisks with the END option specified in this directory will not be included in the following DISKMAP file.

   File USER DISKMAP A has been created.

3. The file created, USER DISKMAP, contains a mapping of all minidisk volumes defined in the USER DIRECT file. It will list any overlaps or gaps found on the volumes. Edit the file and turn off the prefix area with the XEDIT PREFIX OFF subcommand to view 80 columns:
4. Search for the text overlap with the / subcommand:

```bash
====> /overlap
```

You should see the error message: DMSXDC546E Target not found. This means that no minidisks are overlapping each other.

Now search for gaps. You should also see some gaps:

```bash
====> /gap
```

```
VOLUME   USERID      CUU   DEVTYPE      START         END        SIZE
0         500    501    GAP
$$$$$$   DATAMOVE    5F0     3380       00501       00501       00001
DATAMOVE    5FF     3380       00502       00502       00001
```

```
VOLUME   USERID      CUU   DEVTYPE      START         END        SIZE
MMA70C  $ALLOC$     A04  3390       00000       00000       00001
LNXMAINT   0191     3390       00001       00020       00020
LNXMAINT   0192     3390       00021       00320       00300
```

The two gaps should be listed on the right side: a gap of 501 cylinders on the $$$$$$$ volume and a new gap of 1 cylinder exists on the volume that was just used to create disk space for the LNXMAINT user ID; in this case the MMA70C volume.

Don't worry about the 501 cylinder gap, but to avoid a 1 cylinder gap being reported on each user volume, it is recommended to use the user ID $ALLOC$. This user is set to NOLOG which means it can never be logged onto. Thus it is not a conventional user ID, rather, it is a convenient place to put dummy minidisk definitions for cylinder 0 of all PERM volumes.

Look at the rest of the file. You should see the three volumes that z/VM installs onto are already there (520RES, 520W01, 520W02).

5. Get out of the file USER DISKMAP with the QUIT command or by pressing F3.

6. Edit the USER DIRECT file again and add a new minidisk definition:

```bash
==> x user direct
====> /user $alloc
```

```bash
USER $ALLOC$ NOLOG
MDISK A01 3390 000 001 520RES R
MDISK A02 3390 000 001 520W01 R
MDISK A03 3390 000 001 520W02 R
MDISK A04 3390 000 001 <MMA70C> R
```

7. Save your changes and run DISKMAP again. Edit the USER DISKMAP file. This time you should see just the single 501 cylinder gap and cylinder 0 of the first user volume allocated to the $ALLOC$ user ID. When you are done you can quit without saving changes by pressing F3.

```bash
==> diskmap user
==> x user diskmap
====> prefix off
====> /$ALLOC
```

```
VOLUME   USERID      CUU   DEVTYPE      START         END        SIZE
MMA70C  $ALLOC$     A04  3390       00000       00000       00001
LNXMAINT   0191     3390       00001       00020       00020
LNXMAINT   0192     3390       00021       00320       00300
```

...
8. Now that you are sure the minidisk layout is correct, the changes to the USER DIRECT file can be brought online via the DIRECTXA command:

```text
=> directxa user
```

**z/VM USER DIRECTORY CREATION PROGRAM - VERSION 5 RELEASE 2.0**

EOJ DIRECTORY UPDATED AND ON LINE

**HCPDIR494I User directory occupies 39 disk pages**

If the DIRECTXA command fails, you must correct the problem before proceeding.

You have now defined your first z/VM user ID named LNXMAINT.

### 4.7.2 Logging and customizing the new user ID

Now you should be able to logon to the new user ID and format its two minidisks.

1. **Logoff of MAINT and logon to LNXMAINT.**

   ```text
   LOGON LNXMAINT
   z/VM Version 5 Release 2.0, Service Level 0601 (64-bit),
   built on IBM Virtualization Technology
   There is no logmsg data
   FILES: 0003 RDR, NO PRT, NO PUN
   LOGON AT 05:41:34 EST THURSDAY 01/04/07
   z/VM V5.2.0 2006-07-18 08:48
   DMSACP112S A(191) device error
   
   You should see an error message ending in “device error”. When CMS is started, it tries to access the user’s 191 minidisk as file mode A. The 191 minidisk has been defined to this user ID, however, it has never been formatted as a CMS file system.
   
2. **To format this disk for CMS use the FORMAT command. It requires a parameter specifying the file mode to access the disk as, mode A in the following example:**

   ```text
   => format 191 a
   DMSFOR603R FORMAT will erase all files on disk A(191). Do you wish to continue?
   Enter 1 (YES) or 0 (NO).
   1
   DMSFOR605R Enter disk label:
   lxm191
   DMSFOR733I Formatting disk A
   DMSFOR732I 20 cylinders formatted on A(191)
   
   3. Format the larger 192 disk as the D minidisk which should take a minute or two:

   ```text
   => format 192 d
   DMSFOR603R FORMAT will erase all files on disk D(192). Do you wish to continue?
   Enter 1 (YES) or 0 (NO).
   1
   DMSFOR605R Enter disk label:
   lxm192
   DMSFOR733I Formatting disk D
   DMSFOR732I 300 cylinders formatted on D(192)
   
   You have now formatted the two minidisks and accessed them as file modes A and D.

### 4.7.3 Copying a PROFILE XEDIT

Copy the PROFILE XEDIT from the MAINT 191 disk so XEDIT sessions will have a common interface among user IDs.
1. Use the **VMLINK** command to both link to the disk read-only and to access it as the highest available file mode. The default read password is **read**:

```plaintext
=> vmlink maint 191
ENTER READ PASSWORD:
=> read
DMSVML2060I MAINT 191 linked as 0120 file mode Z
```

2. Copy the **PROFILE XEDIT** to your A disk:

```plaintext
=> copy profile xedit z = = a
```

### 4.7.4 Creating a PROFILE EXEC

Create a simple **PROFILE EXEC** that will be run each time this user ID is logged on.

1. Create the new file and add the following lines. REXX EXECs must always begin with a C language-style comment.

```plaintext
=> x profile exec a 
====> a 5
/* PROFILE EXEC */
'acc 592 e'
'cp set run on'
'cp set pf11 retrieve forward'
'cp set pf12 retrieve'
====> file
```

This PROFILE EXEC access the TCPMAINT 592 disk as file mode E, sets CP run on, and sets the retrieve keys per the convention.

2. You could test your changes by logging off and logging back on. However, typing the command **PROFILE** will do the same. By default CMS tries to access the 191 disk as A and the 192 disk as D. Also you should have the TCPMAINT 592 disk accessed as E. To see your minidisks, use the **QUERY DISK** command:

```plaintext
=> profile
DMSACP723I E (592) R/O
=> q disk
```

<table>
<thead>
<tr>
<th>LABEL</th>
<th>VDEV</th>
<th>M</th>
<th>STAT</th>
<th>CYL</th>
<th>TYPE</th>
<th>BLKSZ</th>
<th>FILES</th>
<th>BLKS</th>
<th>USED-(%)</th>
<th>BLKS LEFT</th>
<th>BLK TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>LXM191</td>
<td>191</td>
<td>A</td>
<td>R/W</td>
<td>20</td>
<td>3390</td>
<td>4096</td>
<td>2</td>
<td>9-01</td>
<td></td>
<td>3591</td>
<td>3600</td>
</tr>
<tr>
<td>LXM192</td>
<td>192</td>
<td>D</td>
<td>R/W</td>
<td>300</td>
<td>3390</td>
<td>4096</td>
<td>0</td>
<td>11-00</td>
<td>53989</td>
<td>54000</td>
<td></td>
</tr>
<tr>
<td>TCM592</td>
<td>592</td>
<td>E</td>
<td>R/O</td>
<td>67</td>
<td>3390</td>
<td>4096</td>
<td>877</td>
<td>8167-68</td>
<td>3893</td>
<td>12060</td>
<td></td>
</tr>
<tr>
<td>MNT190</td>
<td>190</td>
<td>S</td>
<td>R/O</td>
<td>100</td>
<td>3390</td>
<td>4096</td>
<td>689</td>
<td>14325-80</td>
<td>3675</td>
<td>18000</td>
<td></td>
</tr>
<tr>
<td>MNT19E</td>
<td>19E</td>
<td>Y/S</td>
<td>R/O</td>
<td>250</td>
<td>3390</td>
<td>4096</td>
<td>1010</td>
<td>26665-59</td>
<td>18335</td>
<td>45000</td>
<td></td>
</tr>
<tr>
<td>MNT191</td>
<td>120</td>
<td>Z</td>
<td>R/O</td>
<td>175</td>
<td>3390</td>
<td>4096</td>
<td>36</td>
<td>224-01</td>
<td>31276</td>
<td>31500</td>
<td></td>
</tr>
</tbody>
</table>

3. Verify that your F11 and F12 keys are set to the **RETRIEVE** command:

```plaintext
=> q pf11
PF11 RETRIEVE FORWARD
=> q pf12
PF12 RETRIEVE BACKWARD
```

### 4.7.5 Copying files associated with this book to LNXMAINT

The z/VM files associated with this book are in the **vm/** subdirectory of the NFS server you set up earlier. These files should be stored on the larger 192 disk which is accessed as your D disk. **Log off of LNXMAINT** so that the 192 disk can be accessed read-write.

**Start an SSH session on the NFS server** and change directory to the VM files associated with this book. Note that the directory name will be one of the following two depending on the distribution you are working with:
FTP to z/VM. By default FTP copies files to your 191 disk, so first change directory to the LNXMAINT 192 disk. Then use the mput * subcommand to copy all the files from the vm/ subdirectory to LNXMAINT. The files are all in ASCII so the default transfer type of ASCII will cause the files to be converted to EBCDIC.

```
# ftp <129.40.178.120>
220-FTPSERVE IBM VM Level 520 at LAT120.PBM.IHOST.COM, 15:20:32 EDT THURSDAY 2006-11-23
220 Connection will close if idle for more than 5 minutes.
User (129.40.178.120:(none)): lnxmaint
331 Send password please.
Password:
230 LNXMAINT logged in; working directory = LNXMAINT 191
ftp> cd lnxmaint.192
250 Working directory is LNXMAINT 192
ftp> prompt
Interactive mode off
ftp> mput *
...
ftp> quit
```

Logon to LNXMAINT. You should see the following files on your D disk:

```
== filel * * d
LNXMAINT FILELIST A0  V 169  Trunc=169 Size=8 Line=1 Col=1 Alt=0
Cmd  Filename Filetype Fm Format Lrecl    Records     Blocks   Date     Time
    CHPW52     XEDIT    D1 V         70        180          3  1/04/07  5:50:42
    CPFORMAT    EXEC     D1 V         79        231          3  1/04/07  5:50:42
    LABEL520    EXEC     D1 V         75        108          2  1/04/07  5:50:42
    LABEL520     XEDIT    D1 V         71         41          1  1/04/07  5:50:42
    LBL520-9    EXEC     D1 V         75         99          1  1/04/07  5:50:42
    LBL520-9     XEDIT    D1 V         71         43          1  1/04/07  5:50:42
    PROFILE    EXEC     D1 V         63         26          1  1/04/07  5:50:42
    RHEL5       CONF-RH5 D1 V         38         13          1  1/04/07  5:50:42
    RHEL5       EXEC     D1 V         67          9          1  1/04/07  5:50:42
    RHEL5       PARM-RH5 D1 V         80          3          1  1/04/07  5:50:42
    SLES10      EXEC     D1 V         68          9          1  1/04/07  5:50:42
    SLES10      PARM-S10 D1 V         65          9          1  1/04/07  5:50:42
    SLES9X      EXEC     D1 V         74          9          1  1/04/07  5:50:42
    SLES9X      PARMFILE D1 V         73          8          1  1/04/07  5:50:42
    SWAPGEN     EXEC     D1 V         72        358          5  1/04/07  5:50:42
```

### 4.8 Customizing system startup and shutdown

When your z/VM system is IPLed, it is often desirable to have important Linux systems also start. Conversely, when you shut down z/VM, it is desirable to have all Linux systems shut down first.

#### 4.8.1 Configuring the AUTOLOG1 PROFILE EXEC

It is recommended that the following tasks be accomplished by using AUTOLOG1’s PROFILE EXEC.

- Configure Linux to shut down gracefully via the SET SIGNAL command
- Overcommit memory via the SET SRM STORBUF command
- Grant access to the VSWITCH for each Linux user
- Start user IDs that should be started via the XAUTOLOG command
1. **Logoff of LNXMAINT and logon to AUTOLOG1.** At the VM READ prompt you have usually been pressing Enter which causes the PROFILE EXEC to be run. If you do not want this EXEC to run, enter the command **ACCESS (NOPROF):**

```
LOGON AUTOLOG1
```

z/VM Version 5 Release 2.0, Service Level 0501 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES: NO RDR, NO PRT, NO PUN
LOGON AT 13:39:10 EST WEDNESDAY 01/18/06
DMSIND2015W Unable to access the Y-disk. Filemode Y (19E) not accessed
z/VM V5.2.0 2005-12-22 09:36

```
=> acc (nopref)
```

2. **Make a copy of the working PROFILE EXEC**

```
=> copy profile exec a = execwrks =
```

3. **Edit the file and add the emboldened text.**

```
=> x profile exec
/****************************/
/* Autolog1 Profile Exec */
/****************************/
'cp xautolog tcpip'      /* start up TCPIP */
'CP XAUTOLOG DTCVS1W1'  /* start VSWITCH controller 1 */
'CP XAUTOLOG DTCVS2W1'  /* start VSWITCH controller 2 */
'cp set pf12 ret'       /* set the retrieve key */
'cp set mdc stor 0m 128m' /* Limit minidisk cache in CSTOR */
'cp set mdc xstore 0m 0m' /* Disable minidisk cache in XSTOR */
'cp set srm storbuf 300%; 250%; 200%;' /* Overcommit memory */
'cp set signal shutdown 180' /* Allow guests 3 min to shut down */

/* Grant access to VSWITCH for each Linux user */
/* XAUTOLOG each Linux user that should be started */
'cp logoff' /* logoff when done */
```

Save your changes with the **FILE** subcommand.

---

**Important:** The **set mdc** and **set srm** lines are z/VM tuning values. It is believed that these are good starts for Linux systems, but may not be optimal. For more reading on these values see the following Web sites:


You may choose to modify or omit some of these settings. Your system should now be configured to start up and send a signal to shut down Linux user IDs.

### 4.8.2 Testing the changes

To test your changes you must relIPL z/VM again. Perform the following steps:

1. Shutdown and relIPL your system.
2. When your system comes back logon as MAINT.

3. Use the QUERY NAMES command to see that TCP/IP, the FTP server and the two VSWITCH controllers have been logged on:

```bash
=> q n
FTPSERVE - DSC, DTCVSW2 - DSC, DTCVSW1 - DSC, TCP/IP - DSC
OPERSYMP - DSC, DISKACNT - DSC, EREP - DSC, OPERATOR - DSC
MAINT -L0004
VSM - TCPIP
```

4. Query the SRM values to see that the new STORBUF settings is in effect and the SIGNAL SHUTDOWN value is set to 180 seconds:

```bash
=> q srm
IABIAS : INTENSITY=90%; DURATION=2
LDUBUF : Q1=100% Q2=75% Q3=60%
STORBUF: Q1=300% Q2=250% Q3=200%
DSPBUF : Q1=32767 Q2=32767 Q3=32767
...
=> q signal shutdown
System default shutdown signal timeout: 180 seconds
```

This output shows that your changes have taken effect.

### 4.9 Addressing z/VM security issues

This section briefly discusses the following security issues.

- **z/VM security products**
- **High level z/VM security**
- **Linux user ID privilege classes**
- **z/VM user ID and minidisk passwords**

**VM security products**

You might want to use a z/VM security product such as IBM RACF or Computer Associates VM:Secure. They allow you to address more security issues such as password aging and the auditing of users/ access attempts.

**High level z/VM security**

The paper *z/VM Security and Integrity* by Cliff Laking and Alan Altmark discusses the isolation and integrity of virtual servers under z/VM. It is on the Web at:


**Linux user ID privilege classes**

Another security issue is the privilege class that Linux user IDs are assigned. The IBM redpaper *Running Linux Guests with less than CP Class G Privilege* by Rob van der Heij addresses this issue. It is on the Web at:


**z/VM user ID and minidisk passwords**

All passwords in a vanilla z/VM system are the same as the user ID. This is a large security hole. The *minimum* you should do is to address this issue.

There are two types of passwords in the USER DIRECT file:
User IDs       The password required to logon with
Minidisks     Separate passwords for read access, write access and multi-write access

Both types of passwords should be modified. This can be done using the CHPW52 XEDIT macro defined in the next section.

4.9.1 Changing passwords in USER DIRECT

Changing the passwords can be done manually in XEDIT. However, this is both tedious and error-prone. So a profile named CHPW52 XEDIT has been included with this book. The source code is in Appendix B.1.2, “The CHPW52 XEDIT macro” on page 237.

This macro will change all z/VM passwords to the same value, which may still not be adequate security given the different function of the various user IDs. If you want different passwords, you have to modify the USER DIRECT file manually, either with or without using the CHPW52 XEDIT macro.

To modify all user ID and minidisk passwords to the same value, perform the following steps.

1. **Logon to MAINT.**

2. **Link and access the LNXMAINT 192 disk to pick up the CHPW52 EXEC:**

   => vmlink lnxmaint 192
   DMSVML2060I LNXMAINT 192 linked as 0120 file mode Z

3. **Make a backup copy of the USER DIRECT file and first be sure the password that you want to use is not a string in the file. For example if you want to change all passwords to lnx4vm, then do the following:**

   => copy user direct c = direwrks = (oldd
   => x user direct c
   ===> /lnx4vm
   DMSXDC546E Target not found
   ===> quit

   The Target not found message shows that the string LNX4VM is not used in the USER DIRECT file, so it is a good candidate for a password.

4. **Edit the USER DIRECT file with a parameter of (profile chpw52) followed by the new password. Rather than invoking the default profile of PROFILE XEDIT, this command will invoke the XEDIT macro named CHPW52 XEDIT and pass it the new password. For example, to change all passwords to lnx4vm, enter the following command:**

   => x user direct c (profile chpw52) lnx4vm

   Changing all passwords to: LNX4VM

   DMSXCG517I 1 occurrence(s) changed on 1 line(s)
   DMSXCG517I 1 occurrence(s) changed on 1 line(s)
   DMSXCG517I 1 occurrence(s) changed on 1 line(s)
   ...

5. **When the profile finishes you are left in the XEDIT session with all passwords modified. You may wish to first examine the changes. Then save the changes with the FILE subcommand:**

   ===> file

6. **Bring the changes online with the DIRECTXA command:**

   => directxa user
   z/VM USER DIRECTORY CREATION PROGRAM - VERSION 5 RELEASE 2.0
   EOJ DIRECTORY UPDATED AND ON LINE
Your new directory is online. Do not forget the new password!

Note that this XEDIT macro will only work on a vanilla USER DIRECT file because it searches for the original user IDs next to passwords. If you want to change your password again, it should be much easier as you can use the XEDIT CHANGE subcommand. For example to change all passwords from lnx4vm to vm5lnx, invoke the following commands:

```
==> x user direct c
====> /lnx4vm/vm5lnx/* *
```

DMSXC6517I 773 occurrence(s) changed on 328 line(s)

Congratulations, your z/VM system is now customized and ready for Linux. It is recommended that you back up your system to tape.

### 4.10 Backing up your z/VM system to tape

Your system is now customized with a running TCP/IP stack, a highly available VSWITCH, a startup and shutdown process and with a user ID for shared files. You have changed the passwords. This would be a good time to back up the system to tape.

There are five system volumes that should be backed up 520RES, 520W01, 520W02, 520SPL and 520PAG (or just the first three if you are using 3390-9s) You also have configured a sixth volume that is important to Linux: that is the first 320 cylinders of the volume with LNXMAINT on it. Back up that entire volume, because the remainder of it will be used for Linux backup data also.

To backup these volumes to tape, refer to chapter 8. Load the System Image, Step 11. Store a Backup Copy of the z/VM System on Tape in the manual The z/VM Guide for Automated Installation and Service, GC204-6099.

### 4.11 Relabeling the system volumes

This step is optional, however, it is recommended. There are times when you will want to change the volume labels of the five z/VM system volumes (or three if you installed onto 3390-3s). If there is a possibility that another vanilla z/VM system with the same labels is installed onto volumes accessible by your z/VM system, one of the systems will not IPL correctly.

When installing z/VM it is possible to modify all but one volume label, that of the 520RES volume. This alleviates the problem that is described next, but it does not alleviate the problem of duplicate volume names.

To understand this possibility, refer to Figure 4-9 on page 62. The z/VM system with the lower device addresses starting at E340 should IPL fine (though you may see a warning at system startup time about duplicate volume labels). However, if the z/VM system starting at device address F000 is IPLed, the 520RES volume will be used, but the remaining volumes in the system are searched for by volume label, not by device address. Because z/VM system 1’s addresses are lower than z/VM system 2’s, system 2 will be using system 1’s volumes. This is not good for either system!
If there is a possibility of another z/VM system being installed on DASD that this system will have access to, it is recommended that you perform the following steps. You will need access to the HMC to perform them:

- “Modifying labels in the SYSTEM CONFIG file” on page 62
- “Modifying labels in the USER DIRECT file” on page 64
- “Changing the labels on the five volumes” on page 65
- “Shutting down your system and restarting it” on page 66

Important: This process must be done as documented. Making a mistake in one of the steps can easily result in an unusable system. Check your steps carefully and your system will come back with no problems. Try to do all steps in succession in a short amount of time. Close your door, don’t answer your phone or e-mail, turn off instant messaging :))

4.11.1 Modifying labels in the SYSTEM CONFIG file

An HMC 3270 session is needed because z/VM will have to be restarted with a FORCE option.

1. Start an Integrated 3270 Console session on the HMC from the CPC Recovery (or just Recovery) menu.

2. If you have not already done so, Logon to MAINT and link and access the LNXMAINT 192 disk to pick up the LABEL520 EXEC and XEDIT macro:

```bash
vmlink lnxmaint 192
DMSVML2060I LNXMAINT 192 linked as 0120 file mode Z
```

3. Note the first five CP-owned volumes via the QUERY CPOWNED command:

```bash
q cpowned
```

<table>
<thead>
<tr>
<th>Slot</th>
<th>Vol-ID</th>
<th>Rdev</th>
<th>Type</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>520RES</td>
<td>A700</td>
<td>Own</td>
<td>Online and attached</td>
</tr>
<tr>
<td>2</td>
<td>520SPL</td>
<td>A701</td>
<td>Own</td>
<td>Online and attached</td>
</tr>
<tr>
<td>3</td>
<td>520PAG</td>
<td>A702</td>
<td>Own</td>
<td>Online and attached</td>
</tr>
<tr>
<td>4</td>
<td>520W01</td>
<td>A703</td>
<td>Own</td>
<td>Online and attached</td>
</tr>
<tr>
<td>5</td>
<td>520W02</td>
<td>A704</td>
<td>Own</td>
<td>Online and attached</td>
</tr>
<tr>
<td>6</td>
<td>MPA705</td>
<td>A705</td>
<td>Own</td>
<td>Online and attached</td>
</tr>
<tr>
<td>7</td>
<td>MPA706</td>
<td>A706</td>
<td>Own</td>
<td>Online and attached</td>
</tr>
<tr>
<td>8</td>
<td>MPA707</td>
<td>A707</td>
<td>Own</td>
<td>Online and attached</td>
</tr>
<tr>
<td>9</td>
<td>MPA708</td>
<td>A708</td>
<td>Own</td>
<td>Online and attached</td>
</tr>
<tr>
<td>10</td>
<td>MPA709</td>
<td>A709</td>
<td>Own</td>
<td>Online and attached</td>
</tr>
<tr>
<td>11</td>
<td>------</td>
<td>----</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>------</td>
<td>----</td>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>

...
4. The labeling convention described in 2.2.1, “Volume labeling convention” on page 11 suggests using ‘M’ in the second character of the label. An XEDIT macro, LABEL520 XEDIT, is supplied to help make this process more reliable. It can be used on both the SYSTEM CONFIG and USER DIRECT files. To modify the labels in the SYSTEM CONFIG file, release the A CP-disk and access it read-write. Back up the SYSTEM CONFIG file:

```plaintext
===> cp rel a
CPRELEASE request for disk A scheduled.
HCPZAC6730I CPRELEASE request for disk A completed.
===> link * cfl cfl mr
===> acc cfl f
===> copy system config f = confwrks = (oldd rep
```

5. Edit the SYSTEM CONFIG file with the LABEL520 XEDIT macro passing the five addresses of the z/VM system volumes (a700-a704 in this example):

```plaintext
===> x system config f (profile label520) <a700> <a701> <a702> <a703> <a704>
DMSXC5171 3 occurrence(s) changed on 3 line(s)
DMSXC5171 1 occurrence(s) changed on 1 line(s)
DMSXC5171 1 occurrence(s) changed on 1 line(s)
DMSXC5171 1 occurrence(s) changed on 1 line(s)
DMSXC5171 1 occurrence(s) changed on 1 line(s)
```

For 3390-9s: If z/VM is installed onto 3390-9s, there should only be three system volumes:

```plaintext
For 3390-9s: If z/VM is installed onto 3390-9s, there should only be three system volumes:
===> q cpowned
Slot Vol-ID Rdev  Type   Status
 1 520RES 9300  Own    Online and attached
 2 520SPL 9301  Own    Online and attached
 3 520PAG 9302  Own    Online and attached
... 
```

6. Clear the screen and you will be left in XEDIT editing the file. Search for the string cp_owned and you should see the new labels. Be sure they are correct before saving the file with the FILE subcommand:

```plaintext
****> /cp_owned
/*                     CP_Owned Volume Statements                     */
*************************************************************************/
CP_Owned Slot 1 MVA700
CP_Owned Slot 2 MVA701
CP_Owned Slot 3 MVA702
CP_Owned Slot 4 MVA703
CP_Owned Slot 5 MVA704
CP_Owned Slot 6 MPA705
```
7. Verify there are no syntax errors:

```shell
cpacc * cf1 a
```

CPACCESS request for mode A scheduled.
Ready; T=0.01/0.01 09:19:57
HCPZAC6732I CPACCESS request for MAINT's 0CF1 in mode A completed.

```shell
q cpdisk
```

<table>
<thead>
<tr>
<th>Label</th>
<th>Userid</th>
<th>Vdev</th>
<th>Mode</th>
<th>Stat</th>
<th>Vol-ID</th>
<th>Rdev</th>
<th>Type</th>
<th>StartLoc</th>
<th>EndLoc</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNTCF1 MAINT</td>
<td>0CF1</td>
<td>A</td>
<td>R/O</td>
<td>520RES</td>
<td>A700</td>
<td>CKD</td>
<td>39</td>
<td>83</td>
<td></td>
</tr>
<tr>
<td>MNTCF2 MAINT</td>
<td>0CF2</td>
<td>B</td>
<td>R/O</td>
<td>520RES</td>
<td>A700</td>
<td>CKD</td>
<td>84</td>
<td>128</td>
<td></td>
</tr>
<tr>
<td>MNTCF3 MAINT</td>
<td>0CF3</td>
<td>C</td>
<td>R/O</td>
<td>520RES</td>
<td>A700</td>
<td>CKD</td>
<td>129</td>
<td>188</td>
<td></td>
</tr>
</tbody>
</table>

You have now changed the labels of the system volumes in the SYSTEM CONFIG file. It is critical that you proceed as your system is now in a state where it will not IPL cleanly.

### 4.11.2 Modifying labels in the USER DIRECT file

In this section you will modify the system volume labels in the USER DIRECT file.

1. Modify the USER DIRECT file again using the `LABEL520 XEDIT` macro. You should see many more occurrences of the labels being changed:

```shell
copy user direct c = direwrks = (oldd rep
```

```shell
x user direct c (profile label520) <a700> <a701> <a702> <a703> <a704>
```

DMSXCG5171 84 occurrence(s) changed on 84 line(s)
DMSXCG5171 134 occurrence(s) changed on 134 line(s)
DMSXCG5171 69 occurrence(s) changed on 69 line(s)
DMSXCG5171 2 occurrence(s) changed on 2 line(s)
DMSXCG5171 1 occurrence(s) changed on 1 line(s)

**For 3390-9s:** Again the macro named `LBL520-9 XEDIT` should be used for systems installed onto 3390-9s. For example, if the system volumes are 9300-9302, the command to edit the SYSTEM CONFIG file would be as follows:

```shell
x user direct c (profile lbl520-9) <9300> <9301> <9302>
```

DMSXCG5171 283 occurrence(s) changed on 283 line(s)
DMSXCG5171 2 occurrence(s) changed on 2 line(s)
DMSXCG5171 1 occurrence(s) changed on 1 line(s)

You may choose to traverse the file before saving the changes with the FILE subcommand:

```shell
file
```
You have now changed the labels of the system volumes in the USER DIRECT and SYSTEM CONFIG files. Again, it is critical that you proceed with the remaining steps.

### 4.11.3 Changing the labels on the five volumes

In this section you will change the labels on the 5 volumes via the CPFMTXA command. You could do this one volume at a time with the CPFMTXA LABEL command. However, the LABEL520 EXEC has been written to make this step easier.

1. Use LABEL520 with the physical addresses of the 5 system volumes:

   ```
   ==> label520 <a700> <a701> <a702> <a703> <a704>
   The volumes are:
   DASD A700 CP OWNED 520RES  50
   DASD A701 CP OWNED 520SPL  1
   DASD A702 CP OWNED 520PAG  0
   DASD A703 CP OWNED 520W01  93
   DASD A704 CP OWNED 520W02  1
   
   The system volume labels will become:
   MVA700 MVA701 MVA702 MVA703 MVA704
   
   ARE YOU SURE you want to relabel the DASD (y/n)?
   y
   
   HCPCCF6209I INVOKING ICKDSF.
   ...
   ICK03000I CPVOL REPORT FOR 0123 FOLLOWS:
   ...
   VOLUME SERIAL NUMBER IS NOW = MVA700
   ...
   VOLUME SERIAL NUMBER IS NOW = MVA701
   ...
   VOLUME SERIAL NUMBER IS NOW = MVA702
   ...
   VOLUME SERIAL NUMBER IS NOW = MVA703
   ...
   VOLUME SERIAL NUMBER IS NOW = MVA704
   ...
   ICK00000I ICKDSF PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
   DASD 0A03 DETACHED
   ```

For 3390-9s: The LBL520-9 EXEC should be used for systems installed onto 3390-9s. For example, if the system volumes are 9300-9302, the command to relabel the system volumes would be as follows:

```
==> 1b1520-9 <9300> <9301> <9302>
The volumes are:
DASD 9300 CP OWNED 520RES  142
DASD 9301 CP OWNED 520SPL  2
DASD 9302 CP OWNED 520PAG  0

The system volume labels will become:
MV9300 MV9301 MV9302

ARE YOU SURE you want to relabel the DASD (y/n)?

y
```
2. Now that the five volumes have been relabeled (sometimes called clipping the volumes), you can run the `DIRECTXA` command to update the directory:

```bash
=> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 5 RELEASE 2.0
EOJ DIRECTORY UPDATED
HCPDIR494I User directory occupies 41 disk pages
Ready(00005); T=0.01/0.01 14:30:37
```

A return code of 5 is expected because the labels in the `USER DIRECT` file are different from the spool data in the currently running system.

Finally, you are ready to issue a `SHUTDOWN` command.

### 4.11.4 Shutting down your system and restarting it

You will need an HMC console session for this step, if you are not already running from there.

To test the changes you must shut your system down and then restart it. You cannot do a `SHUTDOWN REIPL` in this situation because you will have to do a `FORCE` start

```bash
=> shutdown
SYSTEM SHUTDOWN STARTED
HCPSSH601 System shutdown may be delayed for up to 210 seconds
```

Perform the following steps to bring the system back up:

1. From the HMC click the `LOAD` icon in the CPC Recovery (or just Recovery) menu.
2. Select the Clear radio button. All the other parameters should be correct from the previous IPL. Click OK
3. Click Yes on the Load Task Confirmation panel.
4. Go back to the Integrated 3270 console. After a few minutes the Standalone Program Loader panel should appear. Use the TAB key to traverse to the section IPL Parameters and enter the value cons=sysg
5. Press the F10 key to continue the IPL of your z/VM system. This takes around 3 minutes.
6. At the Start prompt you have to specify a FORCE start, again because the spool volume label has changed. Enter the following:

```bash
=> force drain
```

7. Do not change the time of day clock:

```bash
=> no
```

8. When the IPL completes, DISCONNECT from the OPERATOR user ID and logon to MAINT:

```bash
=> disc
```

You should now be able to get a 3270 emulator session as the TCPIP service machine should be up. Get a 3270 session as MAINT and verify the volume labels have changed with the `QUERY CPOWNED` command:

```bash
=> q cpowned
```

<table>
<thead>
<tr>
<th>Slot</th>
<th>Vol-ID</th>
<th>Rdev</th>
<th>Type</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MVA700</td>
<td>A700</td>
<td>Own</td>
<td>Online and attached</td>
</tr>
<tr>
<td>2</td>
<td>MVA701</td>
<td>A701</td>
<td>Own</td>
<td>Online and attached</td>
</tr>
<tr>
<td>3</td>
<td>MVA702</td>
<td>A702</td>
<td>Own</td>
<td>Online and attached</td>
</tr>
<tr>
<td>4</td>
<td>MVA703</td>
<td>A703</td>
<td>Own</td>
<td>Online and attached</td>
</tr>
<tr>
<td>5</td>
<td>MVA704</td>
<td>A704</td>
<td>Own</td>
<td>Online and attached</td>
</tr>
</tbody>
</table>
In the event that you IPLed a system with duplicate system volumes, it is possible that you may have destroyed your saved segments. You will know this is the case when you cannot IPL CMS. Rather, you will have to IPL 190.

**Important:** Only do this if your saved segments have been destroyed! To rebuild saved segments, try the following commands:

```
=> vmfsetup zvm cms
=> sampnss cms
=> i 190 cl parm savesys cms
=> vmfbld ppf segbld esasegs segblist ( all
```

### 4.12 Restoring your z/VM system from tape

It is good to practice to restore a system. You don’t want to be doing your first restore when the pressure is on.

Restoring a z/VM system from tape that has the same set of volume labels as the system that is running is problematic. If there are two z/VM systems on the same LPAR with the same volume labels, both systems cannot be IPLed cleanly. IPLing one of the two will probably find the correct W01, W02, PAG and SPL volumes, but IPLing the other one will probably find the wrong set.

Perform this step only if you successfully completed 4.10, “Backing up your z/VM system to tape” on page 61, and 4.11, “Relabeling the system volumes” on page 61. If you have done both, then the system on tape has volume labels of 520xxx and the system on DASD has volume labels MVyyyy. You can restore this system to five other 3390-3s.

Refer to the Appendix E “Restore the z/VM System Backup Copy from Tape” in the manual *The z/VM Guide for Automated Installation and Service*, GC204-6099.
Servicing z/VM

You cannot solve a problem with the same kind of thinking that created it.
--Albert Einstein

This section describes how to apply:

- A Programming Temporary Fix (PTF)
- A Recommended Service Upgrade (RSU) from “envelope files”

Both processes are basically the same.

Important: When applying service, there is always a chance that you may want to back it out. It is recommended that you have a backup of your system before starting this section.

The application of corrective service to z/VM is covered in the z/VM V5R1 Service Guide and VMSES/E Introduction and Reference. Both of these documents can be downloaded in PDF format from the following URL:
http://www.vm.ibm.com/library

VMSES/E is a component of z/VM that provides the SERVICE and PUT2PROD EXECs. The SERVICE EXEC:

- Installs an RSU or applies CORrective service for z/VM components, features, or products.
- Displays either the RSU level of the component specified or whether a particular PTF or APAR has been applied (when used with STATUS).
- Creates PTF bitmap files (when used with BITMAP).

When SERVICE is successfully complete, the PUT2PROD EXEC places the z/VM components, features, or products that are installed on the z/VM System DDR into production.
5.1 Applying a PTF

You may determine that you need to apply a specific fix or PTF to your system.

For example, an Authorized Program Analysis Report (APAR), VM63895, was opened to address the problems reported with virtual NIC support. There are three known symptoms addressed by this APAR:

► Linux guests may lose connectivity after `shutdown -r now` (or any device reset).
► Using an External Security Manager (ESM) to authorize a VLAN list may lead to an FRF002 abend.
► Virtual Hipersockets NIC configured with "VLAN nnn" (exploiting Set Global VLAN ID) did not really filter inbound frames.

The APAR was assigned the following PTF numbers:

► z/VM 5.1.0 VM63895 UM31612
► z/VM 5.2.0 VM63895 UM31613

There are more details on:

http://www-1.ibm.com/support/docview.wss?uid=isg1VM63895

So for z/VM 5.2, you will want to apply PTF UM31613. Following is an example of how to do so.

5.1.1 Determining if a PTF has been applied

Check to make sure the PTF has not previously been applied.

1. Logon to MAINT and issue the `VMFSETUP` command to set up minidisks for TCPIIP and link to them:

   ```
   ==> vmfsetup zvm cp (link
   VMFSETUP processing started for ZVM CP
   VMFUTL2205I Minidisk|Directory Assignments:
   String    Mode  Stat  Vdev  Label/Directory
   VMFUTL2205I LOCALMOD  E      R/W  2C4   MNT2C4
   VMFUTL2205I LOCALSAM  F      R/W  2C2   MNT2C2
   ...
   VMFSET2760I VMFSETUP processing completed successfully
   ```

2. Use the `VMINFO` command to query the Software Inventory files. Move the Tab key to ZVM and type s to select it on the PPF Fileid panel:

   ```
   ==> vminfo
   PPF Fileid - Help
   
   Product parameter files (PPFs) define the environment and key variables required to process the queries. The following is a list of all PPFs found on all accessed disks. Select one to continue. The View function can be used to examine one or more PPFs.

   Type a "V" next to one or more PPFs to view their contents, or type an "S" next to one PPF to select.

   Options: S - select  V - view
   
   Option    PPF Fileid
   _  $5654260 PPF    D1
   ```
3. Because the description of the PTF cites a component name of “VM CP”, select CP on the Component Name panel.

4. Select PTFs/APARs on the VMFINFO Main Panel.

5. Type in the PTF number UM31613 in the PTF number field then select Status of PTF on the PTF/APAR Queries panel:

PTF/APAR Queries

Enter a PTF or APAR number and type an option code. Then press Enter.

PPF fileid ....... ZVM       PPF  D
Component name .. CP               Setup ... NO
Product ID ......: 5VMCPR20         System .. VM
PTF number ...... UM31613
APAR number ..... 

Options: S - select
Option   Query
s  Status of PTF
-  Requisites/supersedes of PTF
-  Dependencies/superseding of PTF
-  User memo of PTF
-  Serviceable parts included by PTF
-  Abstract of APAR(s)

6. If the PTF has not been successfully applied, you should see the message No data found:

Query Output - PTF Status

PPF fileid ......: ZVM       PPF  D
Component name ..: CP               Setup ..: NO
Product ID ......: 5VMCPR10         System .. VM
---------------------------------------------------------------------
WN:VMFSIP2481W No entries match search arguments
WN:            TDATA :PTF UM31198
WN:            in table 5VMCPR10 SRVRECS J
No data found

This shows that PTF UM31613 has not been successfully applied. The sections that follow describe how to obtain and apply it.

5.1.2 Getting service via Internet FTP

You may get service for z/VM via tapes. However, you may also wish to get service over the Internet. If so, point a Web browser to:

https://techsupport.services.ibm.com/server/login

If you have an IBM user ID and password, use that. If you do not, you can fill out the form to create an IBM ID and password. You should then be at the following Web site:

https://www.ibm.com/account/profile/us

1. Click on Support and Downloads at the top menu.
2. Click on **Downloads and Drivers** on the left frame.

3. Under **Category**, select **zSeries (mainframe)**

4. Under **Operating Systems**, select **z/VM** and click **Software Only**. This should take you to a page entitled **Support for VM**.

5. Click on **Download selective fixes by PTF**. You may be prompted for your IBM ID and password.

6. In the text box **Enter PTF numbers below [e.g: U412345, U467890]**, enter **UM31613**. All other defaults should be correct. Click **Continue**.

7. In the Verify Order page, click **Submit**. You should get a message similar to the following:

   Your order has been submitted for processing. Email will be sent to nospam@us.ibm.com.

   COER NUMBER is <390473266>. This number is used to submit your request. You will receive a confirmation email that contains your ORDER NUMBER.

   ...

### 5.1.3 Downloading the service to z/VM

You should receive two e-mails. The first e-mail has your order number. The second e-mail has instructions on how to download the service files. Make sure you have access to these. Following is an example.

```plaintext
TEXT = Data sent via "INET". To retrieve your service:
TEXT = FTP to: ptf.boulder.ibm.com
TEXT = Log on using userid "owte8a" and password "h2q9nep9"
TEXT = Enter the following FTP commands:
TEXT = cd /390268476/c568411202
TEXT = ascii
TEXT = get ftp1585.txt
TEXT = binary f 1024
TEXT = get rlst1585.bin
TEXT = get rptf1585.bin
```

1. **Logon to MAINT.**

2. The MAINT 500 disk should have a lot of free space, so it is a good minidisk on which to download the files. By default the FTP client saves files on the A disk, so access the 500 disk as A:

   ```
   ==> vmlink tcpmaint 592
   DMSVML2060I TCPMAINT 592 linked as 0120 file mode Z
   ==> acc 500 a
   DMSACC724I 500 replaces A (191)
   ```

3. Now use the FTP client to get the PTF **envelope files** off the Internet. The envelope files can be large so this may take some time. As you are downloading the files, note the file sizes. Following is an example.

   ```
   ==> ftp ptf.boulder.ibm.com
   ftp> <owte8a>
   ftp> <h2q9nep9>
   ftp> cd /390268476/c568411202>
   ftp> ascii
   ftp> get ftp1585.txt
   ...
   ftp> binary f 1024
   ftp> get rlst1585.bin
   ...

   150 Opening BINARY mode data connection for rlst1585.bin (7168 bytes).
   7168 bytes transferred in 0.231 seconds. Transfer rate 31.03 Kbytes/sec.
   ```
ftp> get vptf1585.bin
...
551936 bytes transferred in 22.272 seconds. Transfer rate 24.78 Kbytes/sec.
ftp> quit

4. Use the BROWSE command to view the first text file and verify that the correct number of bytes were downloaded for each file. Press the F3 key to quit.

```shell
===> browse ftp1585.txt
VM PTF Package Information
--------------------------
This file contains byte counts of files to receive and instructions for preparing the files for installation. The byte counts listed below should match the byte counts of the files when they are received using FTP.

FILE BYTE COUNTS
----------------

The vptf1585.bin byte count is: 551936.
The vlst1585.bin byte count is: 7168.

Match these byte counts to that reported during the FTP get.
...
===> F3

5. You should now have the service or envelope files on your z/VM system. Rename the file type from BIN to SERVLINK as this is the file type that the SERVICE command expects.

```shell
===> rename vlst1585 bin a = servlink =
===> rename vptf1585 bin a = servlink =

6. The envelope files arrive in a compressed format to speed downloads. In order to use them they must first be uncompressed with the DETERSE command. Use the (REPLACE parameter to uncompress them in place and save disk space:

```shell
===> deterse vlst1585 servlink a = = = (replace
===> deterse vptf1585 servlink a = = = (replace
```

5.1.4 Receiving, applying, and building service

You must receive, apply, and build the PTF. Then it can be put into production. This can be done in a process that is much easier now with the SERVICE command.

To prepare to use the SERVICE command, you must have a 256MB virtual machine and you must have the minidisk with a lot of free space - that is what the MAINT 500 minidisk is for.

1. Increase the size of the MAINT virtual machine with the DEFINE STORAGE command:

```shell
===> def stor 256M
STORAGE = 256M
Storage cleared - system reset.
```

2. ReIPL CMS:

```shell
===> ipl cms
IPL CMS
z/VM V5.2.0    2006-01-24 13:26
===> Press Enter
```

3. The SERVICE command will write to the current A disk. Again access minidisk 500 as A:

```shell
===> acc 500 a
DMSACC724I 500 replaces A (191)
4. Now use the **SERVICE ALL** command specifying the envelope files you downloaded. Many, many screens of output will scroll by and the screens will automatically be cleared. Important messages will be saved to the A (500) disk. This process may take many minutes. Following is an example:

```bash
==> service all vptf1585
...
VMFSUT2760I VMFSUFTB processing started
VMFSUT2760I VMFSUFTB processing completed successfully
VMFSRV2760I SERVICE processing completed successfully
```

A return code of 0 is ideal. In general a return code of 4 is acceptable. That means that only warnings were issued. A return code of 8 or greater generally means that errors were encountered.

5. The output files written to the A disk are of the form `$VMF* MSGNUM`. You may wish to inspect these files.

```bash
==> file $VMF* MSGLOG
MAINT FILELIST A0 V 169 Trunc=169 Size=5 Line=1 Col=1 Alt=0
Cmd Filename Filetype Fm Format Lrecl Records Blocks Date Time
$VMFSRV $MSGLOG A1 V 80 132 3 1/31/06 12:58:09
$VMFBLD $MSGLOG A1 V 80 76 2 1/31/06 12:57:34
$VMFAPP $MSGLOG A1 V 80 70 1 1/31/06 12:57:13
$VMFREC $MSGLOG A1 V 80 55 1 1/31/06 12:57:12
$VMFMRD $MSGLOG A1 V 80 30 1 1/31/06 12:57:10
```

6. Invoke the **VMFVIEW SERVICE** command to review the results of the previous **SERVICE** command. Press the **F3** key to quit. Following is an example:

```bash
==> vmfview service
************************************************************************
****             SERVICE               USERID: MAINT                ****
************************************************************************
****            Date: 01/31/06            Time: 12:57:09            ****
************************************************************************
====>
```

Ideally there will be no output which as there is in this example. That means the service applied perfectly.

### 5.1.5 Putting the service into production

Use the **PUT2PROD** command to put the service into production.

**Important:** If you run **PUT2PROD** from a 3270 emulator session, you may lose your connection as the TCPIP service machine may be recycled. Therefore you may want to run this command from a console.

In this example, applying PTF UM31613 *did not* affect the emulator session.

```bash
==> put2prod
RDR FILE 0016 SENT FROM MAINT CON WAS 0016 RECS 0004 CPY 001 T NOHOLD NOKEEP
VMFP2P2760I PUT2PROD processing started
VMFP2P2760I PUT2PROD processing started for VMSES
VMFSET2760I VMFSETUP processing started for SERVP2P VMSESP2P
...
Your PTF should now be “put into production”. You may or may not have to reIPL the system, depending on the nature of the PTF applied. It is safest to reIPL via the SHUTDOWN REIPL command in order to completely test the changes:

```bash
=> shutdown reipl ipparms cons=sysc
SYSTEM SHUTDOWN STARTED
...
```

Your z/VM system should come back in a few minutes. You may wish to perform the steps in section 5.1.1, “Determining if a PTF has been applied” on page 70 to verify that this PTF has been applied.

### 5.2 Applying a Recommended Service Upgrade or RSU

Applying an RSU is very similar to applying a PTF described in the previous section. z/VM service can be preventive (RSU) or corrective (COR). Part 4, Service Procedure, in the manual Guide for Automated Installation and Service gives a complete description of applying service to z/VM, however it assumes you are starting with the RSU tape. Following is an example of upgrading to a z/VM 5.2 RSU with the medium being files, not tape.

The section that follows is a summary of applying service and also describes how to obtain the service via envelope files over the Internet.

You must first determine if your system needs service. Use the QUERY CPLEVEL command:

```bash
=> q cplevel
z/VM Version 5 Release 2.0, service level 0501 (64-bit)
Generated at 11/18/05 23:44:09 EST
IPL at 12/19/06 13:44:37 EST
```

The service level four digits are split in half, where the first half is the last two digits of the year and the second half is the service level for that year. Therefore, a service level of “0501” means the first service level for the year 2005 has been applied. If you installed from DVD, that was on the second RSU DVD disc.

The overall steps in applying a service level are as follow:

- Getting service from the Internet
- Download the service to z/VM
- Receive, apply and build the service
- Put the service into production

#### 5.2.1 Getting service from the Internet

The PTF number for the most current RSU for z/VM 5.2.0 is UM97520 The PTF number for the most current RSU for z/VM 5.1.0 is UM97510. Point a Web browser to:

https://techsupport.services.ibm.com/server/login

1. If you have an IBM user ID and password, use that. If you do not, you can fill out the form to create an IBM ID and password. You should then be at the following Web site:

   https://www.ibm.com/account/profile/us

2. Click on Support and Downloads at the top menu.
3. Click on **Downloads and Drivers** on the left frame.

4. Under **Category**, select **zSeries (mainframe)**

5. Under **Operating Systems**, select **z/VM** This should take you to a page entitled Support for VM.

6. Click on **Download selective fixes by PTF**. You may be prompted for your IBM ID and password.

7. In the text box **Enter PTF numbers below [e.g: U412345, U467890]**, enter **UM97520** for the latest z/VM 5.2 service level, **UM97510** for the latest z/VM 5.1 service level, or the appropriate PTF number. All other defaults should be correct.

8. Click **Continue**

9. In the Verify Order page, click **Submit**. You should get a message similar to the following

   Your order has been submitted for processing. Email will be sent to nospam@us.ibm.com.

   COER NUMBER is <390473266>. This number is used to submit your request. You will receive a confirmation email that contains your ORDER NUMBER.

### 5.2.2 Downloading the service to z/VM

You should receive two e-mails. The first e-mail has your order number. The second e-mail has instructions on how to download the service files. Make sure you have access to these. Following is an example.

```
TEXT = Data sent via "INET". To retrieve your service:
TEXT = FTP to: ptf.boulder.ibm.com
TEXT = Log on using userid "owte8a" and password "h2q9nep9"
TEXT = Enter the following FTP commands:
TEXT = cd /390268476/c568411202
TEXT = ascii
TEXT = get ftp8476.txt
TEXT = binary f 1024
TEXT = get rlst8476.bin
TEXT = get rptf0176.bin
TEXT = get rptf0276.bin
TEXT = get rptf0376.bin
```

1. Logon to MAINT.

2. The MAINT 500 disk should have a lot of free space, so it is a good minidisk on which to download the files. By default the FTP client saves files on the A disk, so access the 500 disk as A:

   ```
   ==> acc 500 a
   DMSACC7241 500 replaces A (191)
   ```

3. Link to the TCPMAINT 592 disk to get access to the FTP client command

   ```
   ==> vmlink tcpmaint 592
   DMSVML2060I TCPMAINT 592 linked as 0120 file mode Z
   ```

4. Use the FTP client to get the RSU envelopes off the Internet. The envelope files can be large so this may take some time. It is recommended that you rename the file type from BIN to SERVLINK via FTP because this is the file type that the SERVICE command expects. As you are downloading the files, note the file sizes. Following is an example.

   ```
   ==> ftp ptf.boulder.ibm.com
   ftp> <owte8a>
   ftp> <h2q9nep9>
   ftp> cd /390268476/c568411202>
   ```
ftp> ascii
ftp> get ftp8476.txt
ftp> binary f 1024
ftp> get rlst8476.bin rlst8476.servlink
...
10240 bytes transferred in 0.523 seconds. Transfer rate 19.58 Kbytes/sec.
ftp> get rptf0176.bin rptf0176.servlink
...
36944896 bytes transferred in 191.632 seconds. Transfer rate 192.79 Kbytes/sec.
ftp> get rptf0276.bin rptf0276.servlink
...
26028032 bytes transferred in 132.353 seconds. Transfer rate 196.66 Kbytes/sec.
ftp> get rptf0376.bin rptf0376.servlink
...
52193280 bytes transferred in 269.094 seconds. Transfer rate 193.96 Kbytes/sec.
ftp> quit

You should now have the service or envelope files on your z/VM system.

5. The envelope files arrive in a compressed format to speed downloads. In order to use them they must first be uncompressed with the DETERSE command. Use the (REPLACE parameter to uncompress them in place and save disk space:

```shell
==> deterse rlst8476 servlink a = = = (replace
==> deterse rptf0176 servlink a = = = (replace
==> deterse rptf0276 servlink a = = = (replace
==> deterse rptf0376 servlink a = = = (replace
```

Note: if you did not rename the file names during the FTP session you can use the CMS RENAME command. Type HELP RENAME for help.

6. Use the BROWSE command to read the RSU information. Compare the byte count that you recorded earlier with the values in this file.

```shell
==> browse ftp8476 txt
VM RSU Package Information
--------------------------
This file contains byte counts of files to receive and instructions for preparing the files for installation. The byte counts listed below should match the byte counts of the files when they are received using FTP.

FILE BYTE COUNTS
----------------

The rlst8476.bin byte count is: 10240.
The rptf176.bin byte count is: 36944896.
The rptf276.bin byte count is: 26028032.
The rptf376.bin byte count is: 52193280.
...
```

### 5.2.3 Receiving, applying, and building the service

You must receive, apply, and build the service. Then it can be put into production.

In the past, this was a cumbersome procedure. For example, to receive, apply and build the CP component, the following steps were needed:

```shell
vmfmrdsk zvm cp apply (setup
vmfsetup zvm cp
vmfpsu zvm cp
vmfins install ppf zvm cp {nomemo env \{filename\} nolink override no
vmfapply ppf zvm cp (setup
```
Then the same steps were needed for many other components. The process is much easier now with the SERVICE command. On the other hand, the previous method is more granular and better enables the system administrator to know which pieces of service have been applied.

1. To prepare to use the SERVICE command, you must have a 256MB virtual machine and you must have the minidisk with a lot of free space - that is what the MAINT 500 minidisk is for. Increase the size of the MAINT virtual machine with the DEFINE STORAGE command:

   ```
   ==> def stor 256M
   STORAGE = 256M
   Storage cleared - system reset.
   ==> ipl cms
   IPL CMS
   z/VM V5.2.0 2006-01-24 13:26
   ==> Press Enter
   ```

2. The SERVICE command will write to the current A disk, so you again want to access 500 as A:

   ```
   ==> acc 500 a
   DMSACC724I 500 replaces A (191)
   ```

3. Use the SERVICE ALL command specifying the envelope files you downloaded. Many, many screens of output will scroll by and the screens will automatically be cleared. Important messages will be saved to the A (500) disk. This process may take many minutes or tens or tens of minutes. Following is an example:

   ```
   ==> service all rptf0176 rptf0276 rptf0376
   ...
   VMFSUT2760I VMFSUFTB processing completed successfully
   VMFSRV2760I SERVICE processing completed with warnings
   Ready(00004); T=185.66/191.79 11:56:47
   ```

   A return code of 0 is ideal. Note in the last Ready line that this command returned a code of 4. In general a return code of 4 is acceptable. That means that only warnings were issued. A return code of 8 or greater generally means that errors were encountered.

4. The output files written to the A disk are of the form $VMF* $MSGNUM:

   ```
   ==> file1 $VMF* $MSGLOG
   $VMFP2P $MSGLOG A1 V 80 1520 28 11/10/05 13:35:43
   $VMFBLD $MSGLOG A1 V 80 639 9 11/10/05 13:28:42
   $VMFMRD $MSGLOG A1 V 80 499 7 11/10/05 13:28:39
   $VMFSRV $MSGLOG A1 V 80 1369 25 11/10/05 11:56:47
   $VMFAPP $MSGLOG A1 V 80 682 9 11/10/05 11:54:07
   $VMFINS $MSGLOG A1 V 80 381 6 11/10/05 11:54:05
   ```

5. Invoke the VMFVIEW SERVICE command to review the results of the previous SERVICE command. Following is an example:

   ```
   ==> vmfview service
   ********************************************
   **** SERVICE USERID: MAINT ****
   ********************************************
   **** Date: 11/10/05 Time: 11:43:15 ****
   ********************************************
   CK:VMFSUI2104I PTF UM30896 contains user information. Review the :UMEMO section in file UM30896 $PTFPART
   CK:VMFSUI2104I PTF UM31044 contains user information. Review the :UMEMO section in file UM31044 $PTFPART
   CK:VMFSUI2104I PTF UM31233 contains user information. Review the :UMEMO section in file UM31233 $PTFPART
   ```
CK:VMFSUI2104I PTF UM31275 contains user information. Review the :UMEMO
section in file UM31275 $PTFPART
WN:VMFBDC2250W The following VMHCD objects have been built on BUILD0 300
WN: (I) and should be copied to your workstation:
WN:VMFBDC2250W EEQINSTX EXEBIN
WN:VMFSRV1221W The CP Stand-Alone Dump Utility must be rebuilt. Follow
WN: the instructions in the z/VM Service Guide.

Ideally there will be no output which means the service applied perfectly. In this example,
the above the following messages are generated. The first four VMFSUI2104I messages are
informational. The VMFBDC2250W message is pertinent if you are using the VM HCD tool.
The VMFSRV1221W is pertinent if you are using the CP Stand-alone Dump Utility.

You should see that the service was installed successfully.

5.2.4 Putting the service into production

This section describes how to use the PUT2PROD command to put the service into production.

Important: If you run PUT2PROD from a 3270 emulator session, you may lose your
connection as the TCPIP service machine may be recycled. Therefore you may want to run
this command from a console. If you do switch to a new session, remember to set the
machine size to 256M and access the 500 disk as A.

1. Use the PUT2PROD command to put the service into production. Many screens will scroll by.
   This command can take quite a number of minutes to complete:
   ```
   ==> put2prod
   RDR FILE 0016 SENT FROM MAINT CON WAS 0016 RECS 0004 CPY 001 T NOHOLD NOKEEP
   VMFP2P2760I PUT2PROD processing started
   ...
   USER DSC LOGOFF AS BLDCMS USERS = 7 FORCED BY MAINT
   VMFP2P2760I PUT2PROD processing completed successfully for SAVECMS
   VMFP2P2760I PUT2PROD processing completed successfully
   ```

Even though the service has been “put into production”, the QUERY CPLEVEL command should
still return the current service level; in this example 0501. This is because the new CP load
module (nucleus) has not been invoked:

```
==> q cplevel
z/VM Version 5 Release 2.0, service level 0501 (64-bit)
Generated at 11/18/05 23:44:09 EST
IPL at 12/19/06 13:44:37 EST
```

To invoke the new CP load module, use the SHUTDOWN REIPL command. When your system
comes back up, it should be at the new CP service level, in this example 0602:

```
==> shutdown reipl iplparms cons=sysc
...
==> q cplevel
z/VM Version 5 Release 2.0, service level 0602 (64-bit)
Generated at 09/18/06 18:39:52 EST
IPL at 12/19/06 14:53:28 EST
```

This shows that the new CP load module is now being used.
5.3 Determining z/VM’s service level

Often you will want to be able to query more than just the service level. The following steps were taken from the links CP Maintenance Levels and Virtual Switch TCP/IP Maintenance Levels starting at the Web site:

http://www.vm.ibm.com/virtualnetwork/

Logon as TCPMAINT. Use the QUERY VMLAN command to determine the latest APAR applied:

```bash
==> cp query vmlan
VMLAN maintenance level:
  Latest Service: VM63895
VMLAN MAC address assignment:
  MACADDR Prefix: 020001
  MACIDRANGE SYSTEM: 000001-FFFFFF
  USER: 000000-000000
VMLAN default accounting status:
  SYSTEM Accounting: OFF   USER Accounting: OFF
VMLAN general activity:
  PERSISTENT Limit: INFINITE   Current: 1
  TRANSIENT Limit: INFINITE   Current: 0
```

This shows that the latest APAR applied is VM63895.

The maintenance level of the TCP/IP stack is important to virtual networking. To determine this, first get the active VSWITCH controller:

```bash
==> q vswitch
VSWITCH SYSTEM VSW1     Type: VSWITCH Connected: 8    Maxconn: INFINITE
  PERSISTENT  RESTRICTED    NONROUTER                 Accounting: OFF
  VLAN Unaware
  State: Ready
  IPTimeout: 5         QueueStorage: 8
  Portname: UNASSIGNED RDEV: 3004 Controller: DTCVSW1  VDEV: 3004
  Portname: UNASSIGNED RDEV: 3008 Controller: DTCVSW2  VDEV: 3008 BACKUP
```

This shows the controller is named DTCVSW1. Then use the NETSTAT command with the controller name to determine the maintenance of the TCP/IP MODULE:

```bash
==> netstat tcp <dtcvsw1> level
VM TCP/IP Netstat Level 520
  IBM 2094; z/VM Version 5 Release 2.0, service level 0601 (64-bit), VM TCP/IP Level 520; RSU 0601 running TCP/IP MODULE E2 dated 02/10/06 at 19:00
  TCP/IP Module Load Address: 00C44000
```

This shows information about the TCP/IP MODULE. Use the TCPSLVL command and the complete file specification (TCP/IP MODULE E in this example) to get more information. Of particular interest is the latest APAR applied to TCTOOSD

```bash
==> tcpslvl tcpip module e
  SLVL  TCPIP     PQ22678
  ... 
  SLVL  TCTOOSD   PK00905
  ...
```

Moving on

You should now be done installing, configuring and servicing z/VM. A great attribute of z/VM is that it normally hums along with little maintenance required. It is now time to change your focus to Linux.
SLES 10

This part of the book describes how to install, customize and clone SLES 10 servers.
Configuring an NFS server for SLES 10

“Anyone who has never made a mistake has never tried anything new.”
— Albert Einstein

There are several possible ways to install SLES 10 on the mainframe. It can be installed from FCP attached CD-ROM (in this case all data are read directly from CD) or from network sources like NFS, FTP, HTTP and SMB. Installation via an NFS server is used in this book. To accomplish this, it is recommended that you set up a PC Linux system. This server will supply both the SLES 10 distribution and the files associated with this book.

It must have at least 3 GB of free disk space for one SLES 10 install server. It can be a Linux PC, but it can also be a UNIX box (Sun Solaris, Hewlett Packard HP-UX, IBM AIX or other). The steps in this chapter explain how to configure a PC Linux box as the NFS server.

You can also choose to use a Windows workstation via SMB, but this option is not addressed in this book. Often more problems are encountered when using a Windows workstation to serve a SLES 10 installation so this option is not recommended. If you have no other choice, refer to Section 2.1.2, “Using a Microsoft Windows Workstation” in the manual SUSE LINUX Enterprise Server ARCHITECTURE SPECIFIC INFORMATION. This manual is included on the SuSE CDs.

To get started with Linux on zSeries using this book, you must perform the following tasks:

- “Downloading files associated with this book” on page 83
- “Setting up a SLES 10 install server” on page 84
- “Enabling the NFS server” on page 86

6.1 Downloading files associated with this book

This book has many files associated with it that will be needed to set your system up quickly. You can download the tar file on the Web at:

http://linuxvm.org/present/misc/virt-cookbook-S10.tgz
The tar file `virt-cookbook-S10.tgz` is about 150 KB. Download the file and untar it. The following example shows this being done from the directory `/nfs/`:

```
# mkdir /nfs
# cd /nfs
... download or copy the file virt-cookbook-S10.tgz ...
# tar xzf virt-cookbook-S10.tgz
```

List the files in the new directory `virt-cookbook-S10/`:

```
# ls virt-cookbook-S10/
MD5SUMS.sles10    sitar-1.0.7-270.1.noarch.rpm
README.txt         virtcook-controller-1-S10.s390x.rpm
clone.sh*          virtcook-master-1-S10.s390x.rpm
cmsfs-1.1-8c.s390.rpm    vm/
```

You now have downloaded and untarred the files associated with this book.

## 6.2 Setting up a SLES 10 install server

You may have a licensed version of SLES 10 on physical CDs or you may choose to try an evaluation copy. There is an evaluation copy on the Web starting at the following URL:


Follow the link named *SUSE Linux Enterprise Server 10 for IBM zSeries* and create an account to download the ISO images.

### 6.2.1 SLES 10 ISO image file names and sizes

Following are the file names for the SLES 10 ISO images.

<table>
<thead>
<tr>
<th>CD number</th>
<th>File name</th>
<th>File size in bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SLES-10-CD-s390x-GMC-CD1.iso</td>
<td>667,990,784</td>
</tr>
<tr>
<td>2</td>
<td>SLES-10-CD-s390x-GMC-CD2.iso</td>
<td>656,015,360</td>
</tr>
<tr>
<td>3</td>
<td>SLES-10-CD-s390x-GMC-CD3.iso</td>
<td>412,803,072</td>
</tr>
<tr>
<td>4</td>
<td>SLES-10-CD-s390x-GMC-CD4.iso</td>
<td>579,188,736</td>
</tr>
</tbody>
</table>

### 6.2.2 SLES 10 Service Pack CDs

At the time of writing of this book there were no service packs available.

### 6.2.3 Starting from physical CDs

**NOTE:** Perform this section *only if* you are starting from physical CD discs.

If you are starting with physical CDs, you must first convert them to ISO images. This can be accomplished via the Linux `dd` command which basically does a byte-for-byte copy of the CD contents.

Put the first CD in the drive. It is often available as the file `/dev/cdrom`. If there is no such file on your system, you will have to determine which file (such as `/dev/hdc`) is the device file for the CD drive.
# cd /nfs
# mkdir sles10

Now copy the contents of the first CD to an ISO image using the `dd` `if` (input file) and `of` (output file) parameters. The following example copies the first SLES 10 CD to the appropriately named file. Sometimes, `/dev/cdrom` is automatically mounted over `/mnt/cdrom` when you put the CD in the drive. If so, you will need to unmount it via the `umount` command after copying the contents of the CD:

```
# cd sles10
# dd if=/dev/cdrom of=SLES-10-CD-s390x-GMC-CD1.iso
# umount /mnt/cdrom
```

The CD should start spinning and this will take a couple of minutes to copy. Remove the CD and insert the next. Repeat the `dd` command for each of the physical CDs you have, using the file names shown in the preceding tables.

### 6.2.4 Verifying the ISO images

You should first verify the integrity of the ISO images. This is done via a file of checksum values and ISO file names. The checksums were calculated from the contents of the CD. After downloading or `dd`ing the ISO images, the checksums are calculated again and compared against the original values via the `md5sum` command and the checksum files. For convenience, the checksum files are included in the `nfs-server/` directory.

```
# ls /nfs/virt-cookbook-S10/MD5SUMS.*
/nfs/virt-cookbook-S10/MD5SUMS.sles10
```

**SLES 10 MD5SUM values**

```
# cat /nfs/virt-cookbook-S10/MD5SUMS.sles10
6c495d8dc23f380c6417d58a0291f  SLES-10-CD-s390x-GMC-CD4.iso
```

Use the `md5sum -c` command to verify the integrity of the ISO images. All should report `OK`. The following example shows how to do it:

```
# cd /nfs/sles10
# md5sum -c /nfs/virt-cookbook-S10/MD5SUMS.sles10
SLES-10-CD-s390x-GMC-CD1.iso: OK
SLES-10-CD-s390x-GMC-CD2.iso: OK
SLES-10-CD-s390x-GMC-CD3.iso: OK
SLES-10-CD-s390x-GMC-CD4.iso: OK
```

Any ISO images that do not report `OK` must be downloaded or copied again.

### 6.2.5 Configuring the SLES 10 install server

Now that you have created and verified the appropriate ISO images, you can create a SLES 10 install tree. Since there are no Service Pack CD for SLES 10 available yet, creating install tree is very easy. There are several possibilities:

- Copy the contents of all four CDs into one directory
- Loop mount each ISO image to different directories
- Use the ISO images directly
The last option is also the easiest. The installation process will access the ISO images directly and mount them automatically. It is not known yet how Service Pack CDs will change this process.

There are two files, initrd and vmrdrikr, which have to be copied manually to z/VM and this is the right time to extract them from ISO for future use.

```bash
# cd /nfs/sles10
# mkdir CD1
# mount -o loop=/dev/loop1 SLES-10-CD-s390x-GMC-CD1.iso CD1
# mkdir boot
# cp CD1/boot/s390x/initrd boot/
# cp CD1/boot/s390x/vmrdr.ikr boot/
# umount CD1
# rm -r CD1
```

You should now have a directory, /nfs/, with two subdirectories as follows:

```
/nfs/
|-- sles10/
| `-- boot
|   `-- virt-cookbook-S10/
|   `-- vm
```

You should have four ISO images in the sles10/ directory and initrd and vmrdrikr files in the boot/ subdirectory. You should also have populated the virt-cookbook-S10/ directory with files associated with this book. The next step is to enable the NFS server.

### 6.3 Enabling the NFS server

The method of enabling an NFS server will differ depending upon the operating system. However, the steps are basically the same:

- Be sure you have NFS RPMs installed.
- Export the appropriate directories.
- Start the NFS server in the current run level.

Be sure the NFS server is installed. Typically the RPM is named nfs-utils. If this RPM is not installed, then install it now.

The directories to export via NFS are set in the /etc/exports configuration file. Make a backup copy of the file. Then edit the original copy and add the two directories to be exported:

```bash
# cd /etc
# cp exports exports.orig
# vi exports    // add two lines at the bottom
/nfs/virt-cookbook-S10 *(ro,sync)
/nfs/sles10 *(ro,sync)
```

The *(ro,sync)* parameter specifies that any client with access to this server can get the NFS mount read-only. You may want to be more restrictive than any client (*) for security reasons. Type `man exports` for more details.

Be sure the NFS server is running in your run level. For a SLES Linux, the service name is nfsserver. This can be accomplished with the `chkconfig --list` command:

```bash
# chkconfig --list nfsserver
nfsserver          0:off 1:off 2:off 3:on  4:off 5:on  6:off
```
This output shows that the NFS server is set up to run in the most common run levels: 3 and 5. If your NFS server is not set to start, you will need to set it to run with the `chkconfig` command and turn it on for the current run level with the `rcnfsserver start` command:

```
# chkconfig nfsserver on
# rcnfsserver start
Starting kernel based NFS server
done
```

**For RHEL:** To start the NFS server on Red Hat RHEL 4 or RHEL 5, the parameter to `chkconfig` is `nfs`, not `nfsserver`. Also, the `service` command is used to start and stop services rather than the `rc*` symbolic links:

```
# chkconfig nfs on
# chkconfig --list nfs
nfs  0:off  1:off  2:off  3:on   4:off  5:on   6:off
# service nfs start
Starting NFS services: [ OK ]
Starting NFS daemon: [ OK ]
Starting NFS mountd: [ OK ]
```

Your NFS server should now be running with the directory exported. It is recommended that you test this by mounting the exported directory locally. The following example shows that the `/mnt/` directory is empty. Then the newly exported `/nfs/` directory is mounted using the `localhost` keyword, and the files are listed.

```
# ls /mnt
# mount -t nfs localhost:/nfs/sles10 /mnt
# ls -F /mnt
SLES-10-CD-s390x-GMC-CD1.iso  SLES-10-CD-s390x-GMC-CD4.iso
SLES-10-CD-s390x-GMC-CD2.iso  SLES-10-s390x-GMC.NOTES
SLES-10-CD-s390x-GMC-CD3.iso  boot/
```

This shows that the SLES 10 install root directory is accessible. Now unmount it and test the `virt-cookbook-S10/` directory:

```
# umount /mnt
# mount -t nfs localhost:/nfs/virt-cookbook-S10 /mnt
# ls -F /mnt
MD5SUMS.sles10        sitar-1.0.7-270.1.noarch.rpm
README.txt             virtcook-controller-1-S10.s390x.rpm
clone.sh*              virtcook-master-1-S10.s390x.rpm
cmsfs-1.1-8c.s390.rpm  vm/
```

```
# umount /mnt
```

You should now be able to use this server as the source of your first mainframe Linux installation. Later you will be able to copy the install tree and keep it on zSeries Linux.
7

Installing and configuring SLES 10

“*The most incomprehensible thing about the world is that it is at all comprehensible.*”
— Albert Einstein

In this chapter you will install your first Linux ID, SLES10. Chapters 4, 5 and 6 must be completed before proceeding. SLES10 will be unique in that it will have two copies of Linux; think of it as a dual-boot Linux PC. You will install the following Linux images onto SLES10:

- **Master image** The copy of Linux that will be cloned. This should be as lean as possible so as to be a generic virtual server and to fit comfortably in one 3390-3 DASD.
- **Controller** The copy of Linux that will normally be running and that does the cloning and other functions

In addition to being the *cloner*, the controller can have other functions:

- SLES 10 install server - a tree of the RPMs and other files needed for installation
- NFS server - to export the *install tree* and possibly other data and directories
- Backup server - for incremental backup of key virtual server configuration files
- Time server - for time synchronization of all other Linux virtual servers

To achieve this, you will perform following:

- “Creating the user ID SLES10” on page 89
- “Adding SLES10 to AUTOLOG1’s PROFILE EXEC” on page 92
- “Preparing SLES10 bootstrap files” on page 92
- “Installing the master image” on page 94
- “Configuring the master image” on page 104
- “Installing the controller” on page 114
- “Configuring the controller” on page 121

### 7.1 Creating the user ID SLES10

In this section you will define the SLES10 user ID to z/VM. To accomplish this, perform the following steps:
1. **Logon to MAINT** and edit the USER DIRECT file:

```sh
=> x user direct c
```

In the USER DIRECT file you can group statements that will be common to many user definitions in a construct called a **profile**. This profile can then become part of the user definitions via the `INCLUDE` statement. You used the existing profile `TCPCMSU` when you defined the `LNXMAINT` user.

2. Create a new profile named `LNXDFLT`. This will contain the user directory statements that will be common to all Linux user IDs. To save typing, you can use the `""` prefix commands to duplicate the `IBMDFLT` profile that should be on lines 37-50:

```sh
""037  *****************************************************
00038  *
00039  PROFILE IBMDFLT
00040  SPOOL 000C 2540 READER *
00041  SPOOL 000D 2540 PUNCH A
00042  SPOOL 000E 1403 A
00043  CONSOLE 009 3215 T
00044  LINK MAINT 019D 019D RR
00045  LINK MAINT 019E 019E RR
00046  LINK MAINT 0402 0402 RR
00048  LINK MAINT 0401 0401 RR
00049  LINK MAINT 0405 0405 RR
""050  ****************************
```

3. Edit the duplicated profile by deleting the three `LINK MAINT 040x` lines, and inserting the lines that are emboldened:

```sh
PROFILE LNXDFLT
IPL CMS
MACHINE ESA 4
CPU 00 BASE
CPU 01
NICDEF 600 TYPE QDIO LAN SYSTEM VSW1
SPOOL 000C 2540 READER *
SPOOL 000D 2540 PUNCH A
SPOOL 000E 1403 A
CONSOLE 009 3215 T
LINK MAINT 019D 019D RR
LINK MAINT 019E 019E RR
LINK LNXMAINT 192 191 RR
LINK TCPMAINT 592 592 RR
```

**Notes:**

1. CMS will be IPLed when the user ID is logged onto
2. Machine will of type ESA with a maximum of 4 CPUs that can be defined
3. Defines the base CPU
4. Defines a second CPU - **do not include this** if your LPAR has only a single IFL/CP
5. Defines a virtual NIC connected to the VSWITCH starting at virtual address 600
6. Provides read access to the `LNXMAINT` 192 disk as the user's 191 disk
7. Provides read access to the `TCPMAINT` 592 disk, so that the user has access to TCP/IP services such as an FTP client.

4. Go to the bottom of the file and add the definition for a new user ID named `SLES10`. This user ID is given class B privilege, aside from the typical class G, in order to run the **FLASHCOPY** command, class D privilege to run the **QUERY ALLOC MAP** command, and class E
privilege to run the **QUERY NSS** command. Be sure to replace the volume labels (MMA711-MMA716 in this example) with the labels of your DASD:

```plaintext
USER SLES10 LNX4VM 256M 1G BDEG
  INCLUDE LNXDFLT
  OPTION LNKNOPAS APPLMON
MDISK 100 3390 0001 3338 <MMA70A> MR LNX4VM LNX4VM LNX4VM
MDISK 200 3390 0001 3338 <MMA70B> MR LNX4VM LNX4VM LNX4VM
MDISK 203 3390 0321 3018 <MMA70C> MR LNX4VM LNX4VM LNX4VM
MDISK 204 3390 0001 3338 <MMA70D> MR LNX4VM LNX4VM LNX4VM
MDISK 205 3390 0001 3338 <MMA70E> MR LNX4VM LNX4VM LNX4VM
MDISK 206 3390 0001 3338 <MMA70F> MR LNX4VM LNX4VM LNX4VM
MDISK 207 3390 0001 3338 <MMA710> MR LNX4VM LNX4VM LNX4VM
```

This Linux user ID will have the following minidisks and virtual disks:

100. The root file system of the master image - this is the copy that will be cloned.

101-102. These are virtual disk (VDISK) swap spaces created by `SWAPGEN` upon logging on. They are NOT defined in **USER DIRECT**, but dynamically in the user’s **PROFILE EXEC** when the user ID logs on.

200. The root file system of the Linux controller. This will serve as the administration point for all your Linux virtual servers.

203. A 3018 cylinder minidisk for a file system to be mounted over `/backup/` for backup purposes.

204-207. Minidisks used to create a logical volume mounted over `/nfs/` for making the SLES 10 installation tree and the files associated with this book available via NFS.

5. Go back to the top of the file and search for string **USER $ALLOC$**. Add cylinder 0 of each of the six new volumes to this dummy user ID so they don’t show up as gaps.

```plaintext
======> top
======> /user $alloc$
USER $ALLOC$ NOLOG
  MDISK A01 3390 000 001 <MVA700> R
  MDISK A02 3390 000 001 <MVA703> R
  MDISK A03 3390 000 001 <MVA704> R
  MDISK A04 3390 000 001 <MMA70C> R
  MDISK A05 3390 000 001 <MMA70A> R
  MDISK A06 3390 000 001 <MMA70B> R
  MDISK A07 3390 000 001 <MMA70D> R
  MDISK A08 3390 000 001 <MMA70E> R
  MDISK A09 3390 000 001 <MMA70F> R
  MDISK A0A 3390 000 001 <MMA710> R
...
======> file
```

6. Run **DISKMAP** to check for overlaps and gaps. You should only see the single 501 cylinder gap.

```plaintext
==> diskmap user
==> x user diskmap
====> all /gap/ /overlap/
------------------------ 4 line(s) not displayed ------------------------
                      0 500 501 GAP
------------------------ 322 line(s) not displayed ------------------------
====> quit
```

7. When the disk layout is correct run **DIRECTXA** to bring the changes online:

```plaintext
==> directxa user
```
You have now defined the user ID that will be both the master Linux image and the controller.

7.2 Adding SLES10 to AUTOLOG1’s PROFILE EXEC

The new Linux ID you defined needs access to the VSWITCH. A `SET VSITCH` command with the `GRANT` parameter can be added to AUTOLOG1’s PROFILE EXEC to do this. Also, an `XAUTOLOG` statement can be added if the user ID is automatically logged on at z/VM IPL time:

Link and access the AUTOLOG1 191 disk read-write and edit the file PROFILE EXEC. Add SLES10 to the sections that grant access to the VSWITCH and that `XAUTOLOG` the Linux user IDs:

```sh
==>
link autolog1 191 1191 mr
==>
acc 1191 f
==>
x profile exec f // add two lines
/****************************/
/* Autolog1 Profile Exec */
/****************************/
 'cp xautolog tcpip' /* start up TCPIP */
 'CP XAUTOLOG DTCVSW1' /* start VSWITCH controller 1 */
 'CP XAUTOLOG DTCVSW2' /* start VSWITCH controller 2 */
 'cp set pf12 ret' /* set the retrieve key */
 'cp set mdc stor 0m 128m' /* Limit minidisk cache in CSTOR */
 'cp set mdc xstore 0m 0m' /* Disable minidisk cache in XSTOR */
 'cp set srm storbuf 300% 250% 200%' /* Overcommit memory */
 'cp set signal shutdown 180' /* Allow guests 3 min to shut down */

/* Grant access to VSWITCH for each Linux user */
 'cp set vswitch vsw1 grant sles10'

/* XAUTOLOG each Linux user that should be started */
 'cp xautolog sles10'

'cp logoff' /* logoff when done */
====> file
```

These changes will not take effect until the next IPL, so you must grant this user ID access to the VSWITCH for this z/VM session. This is done as follows:

```sh
==>
set vswitch vsw1 grant sles10
Command complete
```

7.3 Preparing SLES10 bootstrap files

To IPL a SLES 10 installation system, three bootstrap files must be copied and “punched” to the reader. Then you can IPL an install system from the reader (virtual address 00C). The three files are a kernel, a parameter file and an initial RAMdisk. Think of these files as a PC Linux boot floppy or CD. Also, a small REXX EXEC is commonly used to clean out the reader, punch the three files and IPL the reader. The SLES10 parameter file and SLES10 EXEC were already moved to LNXMAINT in 7.1, “Creating the user ID SLES10” on page 89. Therefore, only the kernel and RAMdisk need to be copied.

1. **Start an SSH session as root** on the NFS server.
2. Use the `ftp` command to copy the SLES 10 kernel and initial RAMdisk to the LNXMAINT 192 disk. These files must have a record format of fixed 80 byte records. This format can
be set with the site fix 80 FTP subcommand (if this subcommand fails, try quote site fix 80). Following is an example:

```
# cd /nfs/sles10/boot
# ftp <129.40.178.120>
Connected to 129.40.178.120.
220-FTP.Serve IBM VM Level 520 at LAT120.PBM.IHOST.COM, 16:24:31 EDT THURSDAY 2006-11-23
220 Connection will close if idle for more than 5 minutes.
Name (129.40.178.120:root): lnxmaint
331 Send password please.
Password:
ftp> cd lnxmaint.192
230 Working directory is LNXMAINT 192
ftp> bin
200 Representation type is IMAGE.
ftp> site fix 80
200 Site command was accepted.
ftp> put vmrdr.ikr SLES10.KERNEL
local: vmrdr.ikr remote: SLES10.KERNEL
...
ftp> put initrd SLES10.INITRD
local: initrd remote: SLES10.INITRD
...
ftp> quit
```

3. Go back to your 3270 session. **Logoff of MAINT and logon to LNXMAINT.**

4. The files SLES10 EXEC and SLES10 PARM-S10 should exist on the LNXMAINT 192 disk (D) as they were copied in 7.1, “Creating the user ID SLES10” on page 89. Use the FILELIST command to verify that they were copied in Fixed 80 byte record format. You should see the following files (the number of records may vary):

```
===> filel sls10 * d
LNXMAINT FILELIST A0 V 169 Trunc=169 Size=4 Line=1 Col=1 Alt=0

<table>
<thead>
<tr>
<th>Cmd</th>
<th>Filename</th>
<th>Filetype</th>
<th>Fm Format</th>
<th>Lrecl</th>
<th>Records</th>
<th>Blocks</th>
<th>Date</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLES10</td>
<td>INITRD</td>
<td>D1 F</td>
<td>80</td>
<td>103269</td>
<td>2017</td>
<td></td>
<td>1/04/07</td>
<td>8:26:16</td>
</tr>
<tr>
<td>SLES10</td>
<td>KERNEL</td>
<td>D1 F</td>
<td>80</td>
<td>75579</td>
<td>1122</td>
<td></td>
<td>1/04/07</td>
<td>8:24:44</td>
</tr>
<tr>
<td>SLES10</td>
<td>EXEC</td>
<td>D1 V</td>
<td>68</td>
<td>9</td>
<td>1</td>
<td></td>
<td>1/04/07</td>
<td>5:53:32</td>
</tr>
<tr>
<td>SLES10</td>
<td>PARM-S10</td>
<td>D1 V</td>
<td>65</td>
<td>9</td>
<td>1</td>
<td></td>
<td>1/04/07</td>
<td>5:50:42</td>
</tr>
</tbody>
</table>
```

5. Quit by pressing **F3.**

6. Verify that the file SLES10 EXEC has the correct information. Note the kernel and RAMdisk have hard coded file names, but the file name of the parameter file will be the user ID (userid() function) of the user running the EXEC:

```
===> type sls10 exec d
/* EXEC to punch SLES10 install system to reader and IPL from it */
'CP SPOOL PUN '*
'CP CLOSE RDR'
'PUR RDR ALL'
'PUN SLES10 KERNEL * (NOH'
'PUN' userid() 'PARM-S10 * (NOH'
'PUN SLES10 INITRD * (NOH'
'CH RDR ALL KEEP'
'IPL 00C CLEAR'
```

7. Edit the file SLES10 PARM-S10. The fields you should change are in **<bold>.** Refer to the worksheet in section 2.7.3, “Linux resources worksheet” on page 19.

```
===> x sls10 parm-s10 d
Before:
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
```
HostIP=n.n.n.n Hostname=xxxxx.xxx.xxx
Gateway=n.n.n.n Netmask=n.n.n.n
Broadcast=n.n.n.n Layer2=0
ReadChannel=0.0.0600 WriteChannel=0.0.0601 DataChannel=0.0.0602
Nameserver=n.n.n.n Portname=don'tcare
Install=nfs://n.n.n.n/nfs/sles10/SLES-10-CD-s390x-GMC-CD1.iso
UseVNC=1 VNCPassword=xxxxxx
InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0

**After:**

ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb

8. Save your changes with the FILE subcommand.

Note: The Hostname parameter is not used by the install process, however, it is used in the clone.sh script which also reads these parameter files.

Now you are ready to start the master image installation.

### 7.4 Installing the master image

The master image will be installed onto the SLES10 100-102 disks.

Each Linux image described in this book will have at least three disks - 100 for root file system, 101 and 102 for swap. Both swap disks are VDISKS, which means they are in z/VM virtual memory and thus provide fast access. Disk 101 is 256MB and will act as a primary swap space. Only after it is full, disk 102 which is 512 MB will be used. In a production environment you should monitor swap usage regularly. If the second swap space is being used often you should probably assign the user ID more memory.

You will pick up a PROFILE EXEC from LNXMAINT 192 that will be run when you logon to SLES10. It will create VDISKS via the SWAPGEN EXEC to be used as an in-memory swap space and will prompt you to IPL Linux. Following are the contents of this EXEC:

```bash
=> type profile exec d
/* PROFILE EXEC for Linux virtual servers */
'CP SET RUN ON'
'CP SET PF11 RETRIVE FORWARD'
'CP SET PF12 RETRIVE'
'ACC 592 C'
'SWAPGEN 101 524288' /* create a 256M VDISK disk swap space */
'SWAPGEN 102 1048576' /* create a 512M VDISK disk swap space */
'PIPE CP QUERY' userid() '|' var user'
parse value user with id . dsc .
if (id = 'SLES9'|id = 'SLES9X'|id = 'SLES10'|id = 'RHEL5') then
  do /* this is a controller */
    ip1Disk = 200
  end
else /* this is a Linux virtual server */
  do
```
iplDisk = 100
end
if (dsc = 'DSC') then /* user is disconnected */
  'CP IPL' iplDisk
else /* user is interactive -> prompt */
do
  say 'Do you want to IPL Linux from DASD' iplDisk'? y/n'
  parse upper pull answer.
  if (answer = 'Y') then
    'CP IPL' iplDisk
  end /* else */
1. Logoff of LNXMAINT and logon to SLES10. When you logon, you should see message indicating that a virtual NIC has been created at address 0600 and that a VDISKs 101 and 102 have been created

LOGON SLES10
00: NIC 0600 is created; devices 0600-0602 defined
00: z/VM Version 5 Release 2.0, Service Level 0601 (64-bit),
00: built on IBM Virtualization Technology
00: There is no logmsg data
00: FILES: NO RDR, NO PRT, NO PUN
00: LOGON AT 08:37:24 EST THURSDAY 01/04/07
z/VM V5.2.0 2006-07-18 08:48

DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 101 (64989 4K pages of swap space)
DIAG swap disk defined at virtual address 102 (129981 4K pages of swap space)

2. You are prompted to IPL Linux, but since you have not installed Linux yet, answer n:

Do you want to IPL Linux from DASD 200? y/n
=> n

3. Before you install Linux, it is good to verify the resources. Verify that you have DASD (minidisks) at virtual addresses 100-102 with the QUERY VIRTUAL command (other Linux IDs without class B privilege will just use the QUERY command):

=> q v 100-102
00: DASD 0100 3390 MMA70A R/W 3338 CYL ON DASD A70A SUBCHANNEL = 0000
00: DASD 0101 9336 (VDSK) R/W 524288 BLK ON DASD VDSK SUBCHANNEL = 0013
00: DASD 0102 9336 (VDSK) R/W 1048576 BLK ON DASD VDSK SUBCHANNEL = 0014

4. Verify that you have a virtual OSA at addresses 600-602 with the QUERY VIRTUAL OSA command:

=> q v osa
00: OSA 0600 ON NIC 0600 UNIT 000 SUBCHANNEL = 0007
00: 0600 DEVTYPE OSA CHPID 10 OSD
00: 0600 QDIO-ELIGIBLE QIOASSIST-ELIGIBLE
00: OSA 0601 ON NIC 0600 UNIT 001 SUBCHANNEL = 0007
00: 0601 DEVTYPE OSA CHPID 10 OSD
00: 0601 QDIO-ELIGIBLE QIOASSIST-ELIGIBLE
00: OSA 0602 ON NIC 0600 UNIT 002 SUBCHANNEL = 0009
00: 0602 DEVTYPE OSA CHPID 10 OSD
00: 0602 QDIO-ELIGIBLE QIOASSIST-ELIGIBLE

5. Use the QUERY VIRTUAL STORAGE command to show that you have a 256MB machine:

=> q v stor
00: STORAGE = 256M
6. This is adequate memory to run a SLES 10 Linux image, however it is often too small to install Linux with. Temporarily modify the storage up to 512MB with the `DEFINE STORAGE` command. Then `IPL CMS` and again answer `n` to the question of IPLing Linux:

```plaintext
===> def stor 512M
00: STORAGE = 512M
00: Storage cleared - system reset.
===> ipl cms
z/VM V5.2.0    2006-07-18 08:48
DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 101 (64989 4K pages of swap space)
DIAG swap disk defined at virtual address 102 (129981 4K pages of swap space)
Do you want to IPL Linux from DASD 200? y/n
n
7. Verify that you have a 512MB virtual machine:

```plaintext
===> q v stor
00: STORAGE = 512M
```

This change is for the duration of the user ID session. When you logoff and log back on this user ID, the storage will go back to 256MB.

7.4.1 Begin the SLES 10 installation

Follow these steps to begin the installation of SLES 10.

1. Run the `SLES10 EXEC`. You should see many screens of questions and answers scrolling by. If you had used the default parameter file shipped with SLES 10, you would have had to answer all the networking questions manually. With the proper parameters set in the file `SLES10 PARM-S10`, the install process should proceed to where you access the install program via a VNC client:

```plaintext
===> sles10
00:  NO FILES PURGED
00: RDR FILE 0001 SENT FROM SLES10   PUN WAS 0001 RECS 076K CPY 001 A NOHOLD NO KEEP
00: RDR FILE 0002 SENT FROM SLES10   PUN WAS 0002 RECS 0009 CPY 001 A NOHOLD NO KEEP
00: RDR FILE 0003 SENT FROM SLES10   PUN WAS 0003 RECS 103K CPY 001 A NOHOLD NO KEEP
00: 0000003 FILES CHANGED
00: 0000003 FILES CHANGED
Linux version 2.6.16.21-0.8-default (geeko@buildhost) (gcc version 4.1.0 (SUSE Linux)) #1 SMP Mon Jul 3 18:25:39 UTC 2006
We are running under VM (64 bit mode)
Detected 2 CPU's
Boot cpu address  0
Built 1 zonelists
...
Important: If you get the following error that the gateway cannot be pinged:

```
HCPIP2833E IP address is already registered....
... Warning: The gateway address 129.40.178.254 did not ping. Do you want to ignore this error and continue anyway? (Yes/No)
```

This may be due to a bug in the OSA Express card. If so it may be due to a race condition. You can try working around this error by IPLing the system again from the reader with the command (you may have to try multiple times):

```
===> #CP IPL 00C
```

This bug was fixed with OSA microcode level 6.26. It is in MCL07 at driver 55 EC J13477. Driver 55 is for the z/890 and z/990. The OSD 626 code level was available on Feb. 16th 2005. Driver 3G is for the z/800 and z/900. The latest code level for the OSD code level 35A was available on March 2nd 2005.

You should see the message:

```
*** You can connect to 129.40.178.130, display :1 now with vncviewer
 *** Or use a Java capable browser on http://129.40.178.130:5801/
```

2. From your workstation you can open a Java-enabled browser to access YaST2 at the specified URL. The logon prompt in Figure 7-1 shows VNC access through a Java-enabled browser.

In addition to a browser, you can also use a standalone VNC client if desired. If this is the case, the server to connect to would be **129.40.178.130:1** in this example.

![Figure 7-1 VNC viewer through a Java-enabled browser](image)

3. You could disconnect from the 3270 session, however messages to the console will be lost. It is recommended that you stay connected, so you will have to clear the screen periodically (or the install process may be delayed waiting for the screen to clear itself).

4. Enter the URL in your browser and log in using the VNC_PASSWORD that was specified in the file SLES10 PARM-S10,

Now the installation process should begin.
7.4.2 Beginning YaST installation

The installation program that is running is yast2. Perform the following steps:

1. Choose the language, **English**, (or your language) and click Next.
2. The **SLES10 license agreement window** should appear. Choose Yes, I agree to the License Agreement and Click Next.
3. The Disk Activation window should appear. Choose Configure DASD Disks.
4. The YaST2 DASD Disk Management window should appear: you will see all the DASD available to SLES10.
   a. Use 100-102 for the master image.
   b. **Highlight each of the three channels** and click Select or Deselect to select them.
   c. You should see a the word Yes next to those you just selected. **Activate them for the Linux you're about to install.** Click Perform Action -> Activate. as shown in the left side of Figure 7-2 on page 98.
   d. Disks 100 must be formatted so that Linux can use it. Deselect disks 101 and 102, so that 100 remains selected. Now click Perform Action -> Format as shown on the right side of the figure.

5. You should see a window asking for one Parallel Formatted Disks: click OK.
6. Click Yes to the question **Really format the following?**
7. A progress indicator window should appear displaying progress. This step can take 5-15 minutes depending on the type of channel and the speed of the disks.
8. When the formatting is complete, click Next, in Disk Activation window click Next again.
9. A window will appear asking for the installation mode: select **New installation** and Click **Next**

10. The *Clock and Time Zone* window will appear. Choose your time zone and set clock, click **Next**.

11. This will bring you to the *Installation Settings* window. Click **Partitioning**. The *Expert Partitioner* window will appear as shown in Figure 7-3 on page 99.

12. Click **Create** and because `/dev/dasda` has no partitions, you will be brought to *Create a Primary Partition on /dev/dasda*.

![Expert Partitioner Window]

*Figure 7-3   Disk partitioner - before customization*

13. You should see a window similar to that shown in the Figure 7-4 on page 100.
   a. Click the **Format** radio button.
   b. Choose **Ext3** as the file system
   c. Select a mount point of `/`
   d. Click **OK**.
14. You don’t have to format `/dev/dasdb1` and `/dev/dasdc1` because they were properly formatted as a Linux swap space by the `SWAPGEN EXEC`, and thus should be recognized as a swap space.

15. Back in the `Expert Partitioner` window, you should see something similar to Figure 7-5. Click **Finish**

---

**Figure 7-4  `/dev/dasda1` (root fs) specification**

**Figure 7-5  Disk partitioner - after customization**

### 7.4.3 Continuing the YaST2 installation

Follow these steps to continue the YaST2 installation
1. In the *Installation Settings* window, select **Software**.

2. The *Software Selection* window opens. Select **Server Base System** and **32Bit Runtime Environment**, uncheck all other setting as showed in Figure 7-6, click **Accept**.

3. Back in the *Installation Settings* window, you may click **Language**.

4. In the *Languages* window, select your language and click **Accept**.

5. While still in *Installation Settings* window, click **Expert**, scroll to the bottom and choose **Default Runlevel**.

6. Change to **3: Full multiuser with network** and click **OK**.

7. You are now ready to begin copying the RPMs onto your root file system. Back in the *Installation Settings* window, check the settings and click **Accept**.

8. A confirmation window will appear: click **Install**.

A minimal SLES 10 system will be installed onto DASD. This should take about 5-30 minutes depending on network speed and the performance of your NFS server.

### 7.4.4 Booting your new Linux system from disk

After the first part of installation completes, your Linux system will shutdown and you will need to return to your z/VM 3270 session. **IPL** the newly installed system from disk to continue installation. Issue the following command and your new kernel should boot from disk:

```
Power down.
00: HCPGSP2629I The virtual machine is placed in CP mode due to a SIGP stop from CPU 01.
```
The virtual machine is placed in CP mode due to a SIGP stop and store status from CPU 01.

Enter IPL 100 and Linux will be IPLed.

```bash
=> ipl 100
00: zIPL v1.5.3 interactive boot menu
00:
00: 0. default (ipl)
00:
00: 1. ipl
00: 2. failsafe
00:
00: Note: VM users please use '#cp vi vmsg <number> <kernel-parameters>'
00:
00: Please choose (default will boot in 10 seconds):
```

At this prompt, you can wait 10 seconds for the default IPL to proceed, or you can try to type `#cp vi vmsg 0` if you are a fast typist :))

```bash
00: Booting default (ipl)...
Linux version 2.6.16.21-0.8-default (geeko@buildhost) (gcc version 4.1.0 (SUSE Linux)) #1 SMP Mon Jul 3 18:25:39 UTC 2006
We are running under VM (64 bit mode)
...
A log file will be written to: /tmp/vncserver.log ...
***
*** You can connect to 129.40.178.130, display :1 now with vncviewer
*** Or use a Java capable browser on http://129.40.178.130:5801/
***
(When YaST2 is finished, close your VNC viewer and return to this window.)
*** Starting YaST2 ***
...
```

The install program will bring up a VNC server again to complete the installation.

### 7.4.5 Completing YaST2 installation

Go back to the same VNC client used for the first part of installation. If it was a browser either click **Login Again** or click the browser's **refresh** button until another VNC login screen appears. If you are using a VNC client then open that application again.

1. **Log in** using the same VNC password (LNX4VM in this example) as the one used to first bring up YaST2.

2. The first window you probably see will be similar you saw last time - package installation. However there is a good chance it will be done already and you will come to **Hostname and Domain Name** window.

3. Enter the required values, in this example lat130 is the host name and pbm.ihost.com is the domain name. Uncheck the **Change Hostname via DHCP** checkbox. Click **Next**.

4. Now you will get a window for setting the root password. Enter your desired root password twice and click **Next. Don’t forget** this password!!

5. In the **Network Configuration** window you will see **Firewall is enabled, click on the word enabled** to change it to **disabled.** All other values should be correct (Figure 7-7) so just click **Next**.
6. In the Test Internet Connection window select No, skip this test and click Next.

7. The next window will be Installation Settings. Select the Skip configuration radio button and click Next.

8. In the User Authentication Method window accept the default of Local (/etc/passwd) and click Next.

9. The next window will be Add a new local user. You may choose to add a local user so as to have a user other than root on all cloned systems. When you are done, click Next.

10. In the Writing the system configuration window the SuSEconfig tool writes all your settings to disk.

11. The next window will be Release Notes. After reviewing the release notes, click Next.

12. In the Hardware Configuration window choose the Skip Configuration radio button and click Next.

13. The last installation window is Installation Completed. Uncheck the check box Clone This System for Autoyast check box and click Finish.

You are done installing Linux! Your VNC session will be shut down. Return to the 3270 session and you may have to clear the screen a few times. Your new Linux will finish its configuration, but you can again disconnect via the command:

```bash
==> # cp disc
```

From this point forward, it is recommended that you access your Linux systems via SSH. If you have a Windows desktop, but do not have an SSH client configured, see 3.1, “PuTTY: A free SSH client for Windows” on page 21.
Start an SSH session into the master image as root.

7.5 Configuring the master image

Now you want to customize the master image before cloning. The following high level steps are recommended though you may add or omit some steps. However, omitting steps can negatively affect the operation of the clone.sh script and the DAZL Web application.

- “Create nightly.sh script” on page 104
- “Adding additional RPMs” on page 104
- “Configuring the VNC server” on page 105
- “Preparing for Online Update” on page 107
- “Removing unnecessary RPMs” on page 107
- “Turning off unneeded services” on page 108
- “Configuring rsyncd” on page 109
- “Applying service - online update” on page 109
- “Configuring sitar” on page 111
- “Setting the software clock accurately” on page 111
- “Setting system to halt on SIGNAL SHUTDOWN” on page 112
- “Turning off the hz_” on page 112
- “Modifying zipl.conf” on page 113
- “Rebooting the system and verifying changes” on page 114

7.5.1 Create nightly.sh script

To run some important commands regularly create a script named nightly.sh which is invoked by cron. This script will run once a day from the /etc/cron.daily/ directory:

```bash
#!/bin/bash
let secs=$RANDOM/9                    # number of seconds between 0 and 1 hour
sleep $secs                           # sleep so cron load is dispersed
sitar --format=html --outfile=/etc/sitar.html  # create sitar report in /etc
/usr/sbin/ntpd -q                     # Adjust the clock
chmod +x nightly.sh
```

The first line creates a random number between 0 and 3640 - approximately the number of seconds in one hour. The second line puts the script to sleep for up to an hour so the cron workload across the LPAR does not all start at once.

The third line creates an HTML report about the virtual server in the /etc/ directory via sitar. This report will be copied to the controller each night via it’s cron script, and then will be referenced by the DAZL Web application

The last line adjusts the clock accurately. In the next section you will set up NTP to point to the controller as a software clock reference.

7.5.2 Adding additional RPMs

Add additional RPMs using the yast command with the -i option. YaST will conveniently install the packages specified and automatically resolve their dependencies. You may choose to add more or fewer RPMs.

- The findutils-locate package will be used to enable the locate and updatedb commands.
- The openmotif package will be used by the VNC server as a usable window manager.
The `xterm` package will provide xterm terminal emulator.

You may choose to add other RPMs, or may choose to omit some of the above. To add the RPMs described above, use the `yast -i` command:

```
# yast -i findutils-locate openmotif xterm
```

You should see `yast` curses screens go by as the software is added.

### 7.5.3 Configuring the VNC server

Often applications require a graphical environment. The `tightvnc` package is a Virtual Network Computing (VNC) server. It allows for a graphical environment to be set up easily via the `vncserver` command.

SLES 10 configures a VNC connection which starts at KDE prompt by default. It is configured through the `xinetd` daemon. It is recommended that you disable it. Edit the file `/etc/xinetd.d/vnc`. In sections `service vnc1` and `service vnchttpd1` add the parameter `disable=yes`.

```
# vi /etc/xinetd.d/vnc
service vnc1
{
    socket_type   = stream
    protocol      = tcp
    wait          = no
    user          = nobody
    server        = /usr/X11R6/bin/Xvnc
    server_args   = :42 -inetd -once -query localhost -geometry 1024x768 -depth 16
    type          = UNLISTED
    port          = 5901
    disable       = yes
}
...

service vnchttpd1
{
    socket_type   = stream
    protocol      = tcp
    wait          = no
    user          = nobody
    server        = /usr/X11R6/bin/vnc_inetd_httpd
    server_args   = 1024 768 5901
    type          = UNLISTED
    port          = 5801
    disable       = yes
}
```

VNC is the only service that `xinetd` starts by default. So you can turn off the `xinetd` service with the `rcxinetd stop` command for this session and the `chkconfig` command at boot time:

```
# rcxinetd stop
Shutting down xinetd: done
# chkconfig xinetd off
```

The `openmotif` package allows for the motif window manager (mwm) that is more usable than the default Tiny Window Manager (twm) that VNC uses by default.

When you first start the VNC server, you are prompted to set a password. After it is set, this will be the password that you will need to connect to it from a VNC client:
```bash
# vncserver
lat130:~ # vncserver

You will require a password to access your desktops.

Password: lnx4vm
Verify: lnx4vm
Would you like to enter a view-only password (y/n)? n

New 'X' desktop is lat130:1

Creating default startup script /root/.vnc/xstartup
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/lat130:1.log

Now it is recommend, but not required to change the window manager from the Tiny Window Manger (twm) to openmotif (mwm). To do this, first stop the VNC server using the -kill :1 argument:

```
# vncserver -kill :1
Killing Xvnc process ID 2621C
```

Change the window manger from twm to mwm in the file /root/.vnc/xstartup:

```
# cd /root/.vnc/
# vi xstartup
#!/bin/sh
xrdb $HOME/.Xresources
xsetroot -solid grey
xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
mwm &
```

Remove the passwd file so the cloned system does not have the same password as you just entered.

```
# rm passwd
```

When a system is cloned, the password will be prompted for the first time that the VNC server is initialized. A sample session is shown in Figure 7-8.

![VNC client session to the VNC server](image_url)

**Figure 7-8** VNC client session to the VNC server
```
7.5.4 Preparing for Online Update

This step is optional but highly recommended. If you do not have Internet access from installed image you will have to skip this step.

Before you can use Online Update (recommended), online sources have to be configured. This is done through the Novell Customer Center Configuration. To configure the Customer Center, a Web browser is needed. For this reason, a VNC server session must be started.

Start the VNC server with the `vncserver` command:

```
# vncserver
```

You will require a password to access your desktops.

Password: <lnx4vm>
Verify: <lnx4vm>
Would you like to enter a view-only password (y/n)? n

New 'X' desktop is lat130:1
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/lat130:1.log

Now connect to <129.40.178.130>:1 with your VNC viewer. In a terminal session, start YaST with command `yast2`. Perform the following steps:

- Choose the Novell Customer Center Configuration selection.
- On next window select Configure Now and click Next. In the Manual Intervention Required window, read the information and press Continue.
- Now mozilla-xulrunner (with many other packages) will be installed so that the browser can be run. Click Continue and wait until everything is installed and a browser is started. This will take couple of minutes.
- In the Web Registration page enter your e-mail address twice and name of the system (optional). If you have activation code for SLES, you can enter it here. If you don’t enter activation code, Novell will create temporary one which will expire in 15 days. Press the Submit button.
- On next page window the information and press Continue.
- Now Contacting server... window will be active for couple of minutes and online sources will be set in YaST. You can check it in Details... window. Click OK.
- Click Quit to quit YaST.

Your system should now be configured to run Online Update.

7.5.5 Removing unnecessary RPMs

For the master image it is desirable to have as lean a Linux image as possible from which to build on. Therefore as many RPMs as possible should be removed. You may choose to add to or omit some from the list.

If you went through Novell Customer Center Configuration above, remove these packages they are not needed any more:
It is recommended that the following RPMs be removed. You may choose to add to or omit some from the list. They can all be removed using one command, however, the following series of commands may be more readable.

Issue the following `rpm` commands:

```
# rpm -e mozilla-xulrunner
# rpm -e Mesa als a atk audi ofile cairo desktop-file-utils dialog esound fam gconf2 glitz
gnome-keyring gnome-mime-data gnome-vfs2 gtk21 libart1gpl libbonobo libbonoboui
libglade2 libgnome libgnomecanvas libgnomeui libid1 libsmcclient mDNSResponder-1ib
mozilla-nsp r mozilla-nss orbit2 pango shared-mime-info
```

After deleting the RPMs you should see that the usage of the root file system has dropped to approximately 38%:

```
# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/dasda1           2.3G  827M  1.4G  38% /
udev                  247M   76K  247M   1% /dev
```

### 7.5.6 Turning off unneeded services

There are a number of services which are started in a SLES 10 minimum system. In order to keep the master image as lean as possible in terms of CPU usage, some of these services can be turned off. Turn off the following services via the `chkconfig` command:

```
# chkconfig nfsboot off
# chkconfig slpd off
# chkconfig novell-zmd off
# chkconfig postfix off
# chkconfig resmgr off
```

If you didn’t remove splash package turn off following services

```
# chkconfig splash off
# chkconfig splash_early off
```

You may choose to leave these services on, or turn others off. You can review which services are now configured to start in run level 3 with the following command:

```
# chkconfig -l | grep 3:on
```

<table>
<thead>
<tr>
<th>Service</th>
<th>0:off</th>
<th>1:off</th>
<th>2:on</th>
<th>3:on</th>
<th>4:off</th>
<th>5:on</th>
<th>6:off</th>
</tr>
</thead>
<tbody>
<tr>
<td>cron</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dbus</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>haldaemon</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>network</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nfs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nscd</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>portmap</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>random</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
7.5.7 Configuring rsync

It is important for Linux configuration files to be backed up. There are many ways to
accomplish this and one of them is rsync. Rather than copying entire files as with the scp -rp
command, rsync just copies the differences or deltas of changed files. For this reason it is an
efficient tool for backups.

Use the chkconfig command to set rsync to start up:

```
# chkconfig rsync on
```

There are two files that need to be configured: the main configuration file /etc/rsyncd.conf
and a file to allow only the controller to connect, /etc/trusted.hosts. First make a backup
copy then edit the file /etc/rsyncd.conf commenting out two lines and adding five lines at the
bottom.

```
# cd /etc
# cp rsyncd.conf rsyncd.conf.orig
# vi rsyncd.conf
gid = users
read only = true
# use chroot = true
transfer logging = true
log format = %h %o %f %l %b
log file = /var/log/rsyncd.log
hosts allow = trusted.hosts
# slp refresh = 300
max connections = 1
[etc]
path = /etc
read only = yes
list = yes
```

The new [etc] section at the bottom will allow access from the controller to the /etc/
directory.

The file /etc/trusted.hosts needs a single entry, the TCP/IP address of the controller. This
will allow only the controller to get an rsync session to the virtual servers. Create this file now:

```
# vi trusted.hosts // add one line - the TCP/IP address of the controller
<129.40.178.130>
```

7.5.8 Applying service - online update

If you have a SuSE Maintenance Web account then you can use it to retrieve the latest
patches for SLES 10. Because many of these patches contain security and bug fixes, it is
recommended that you apply the patches for the master image so that it is up to date.
Subsequently all the servers you clone after the master image will also be up to date.

Section 7.5.4, “Preparing for Online Update” on page 107 has to be finished before you
continue here.

Start the VNC server:

```
# vncserver
```
Connect to <129.40.178.130>:1 with your VNC viewer. In a terminal session start YaST with command `yast2`. Perform the following tasks:

- In YaST choose *Software* on the left and *Online Update* on the right.
- You will be presented with list of downloadable patches. Patches to packages you have installed will be checked for download. You may choose here which packages to install and which not. It is highly recommended to download security patches.

![Available patches in Online Update.](image)

- When done, click **Accept**. There may be other windows coming with warnings, read them carefully. For example, you may get window telling about kernel security update which informs you to reboot the system after you finish. Click **Accept**.
- Patches will be downloaded and installed. It will take some time depending on number of patches, their size and your network speed.
- Click **Finish** when install ends.
- Click **Quit** to quit YaST.

You may stop the VNC server now and remove password so that cloned virtual servers will not share the same one.

```
# vncserver -kill :1
# rm /root/.vnc/passwd
```

When you finish here, don’t reboot yet, you will reboot shortly after some boot configuration changes are made.
7.5.9 Configuring sitar

SITAR is a tool that creates documentation describing your system. It is an acronym for System Information At Runtime. In addition to backing up the /etc/ configuration files, a document describing your system might be helpful. If sitar is run once a day and the output is put in a file in /etc/, then you will have useful information about your system backed up.

The sitar output will be used by the DAZL application later.

The Sitar package that comes with SLES 10 has a bug which only allows write access to the directory /tmp/. Check which version of sitar is on your system.

```
# rpm -qa|grep sitar
sitar-1.0.6-7.2
```

If your version is lower than 1.0.7, you may choose to replace it with a 1.0.7 version included in the tar file associated with this book. Copy it from the PC NFS server:

```
# scp -rp <129.40.46.206>:/nfs/virt-cookbook-S10/sitar* /tmp
Password:

sitar-1.0.7-270.1.noarch.rpm                  100%   88KB  87.6KB/s   00:00
#
# rpm -Uvh /tmp/sitar-1.0.7-270.1.noarch.rpm
Preparing...                ########################################### [100%]
1:sitar                  warning: /etc/sysconfig/cfg2scm created as /etc/sysconfig/cfg2scm.rpmnew
warning: /etc/sysconfig/sitar created as /etc/sysconfig/sitar.rpmnew
########################################### [100%]
#
# rpm -q sitar
sitar-1.0.7-270.1
```

An HTML file, written to /etc/sitar.html, describing your system configuration can be created with the following command. And then you can view your system information in /etc/sitar.html.

```
# sitar --format=html --outfile=/etc/sitar.html
pcilib: Cannot open /proc/bus/pci
lspci: Cannot find any working access method.
Generating /etc/sitar.html ...
```

The warning messages should not be a problem. You can view the resulting HTML file, /etc/sitar.html with a browser.

You created nightly.sh in /etc/cron.daily/ directory in section 7.5.1, “Create nightly.sh script” on page 104, it will run sitar every day.

Because the output is sent to the /etc/ directory and because that directory is backed up nightly with rsync (if all the customization steps are completed successfully), you will have a copy of the sitar output backed up once a day.

7.5.10 Setting the software clock accurately

It is important to have the Linux software clock set properly. This is commonly accomplished by running the xntpd daemon against some accurate time sources. However, this is expensive in terms of CPU costs. A compromise between an extremely accurate clock (running xntpd) and using fewer CPU cycles (not running it) is to reset the clock once a day. This can be done with a script that runs the ntpd -q command from the /etc/cron.daily/ directory. NTP will be configured so the controller will be the external clock. In 7.7.13, “Turning on the NTP server” on page 124 the controller will be configured to point to external clocks on the Internet.
You need to edit the file /etc/ntp.conf. You will comment out two lines that are pointing to an “Undisciplined Local Clock” and add one line pointing to the external time reference:

```
# cd /etc
# cp ntp.conf ntp.conf.orig
# vi ntp.conf // change the line “server n.n.n.n” to point to controller
...
## Undisciplined Local Clock. This is a fake driver intended for backup
## and when no outside source of synchronized time is available.
##
## server 127.127.1.0            # local clock (LCL)
## fudge  127.127.1.0 stratum 10 # LCL is unsynchronized
##
## Outside source of synchronized time
##
## server xx.xx.xx.xx           # IP address of server
server <129.40.178.130>
...  
```

Now you should have the configuration file /etc/ntp.conf pointing to the controller, and a nightly.sh script in the directory /etc/cron.daily/ that will run once a day to set the software clock accurately.

### 7.5.11 Setting system to halt on SIGNAL SHUTDOWN

By default, SLES 10 reboots when a Ctrl-Alt-Del key sequence is trapped. This key sequence is simulated by z/VM when it issues a `SIGNAL SHUTDOWN` command. Rather than rebooting, you want your system to halt (shutdown).

Change this setting by changing `shutdown -r` to `shutdown -h` in the `/etc/inittab` file:

```
# cd /etc
# vi inittab // change shutdown -r to shutdown -h
...
# what to do when CTRL-ALT-DEL is pressed
ca::ctrlaltdel:/sbin/shutdown -h -t 4 now
...  
```

This change will be picked up when the system is rebooted.

### 7.5.12 Turning off the hz_timer

By default, the Linux kernel wakes up 100 times per second to see if there is any work to be done. While this is fine for a PC running a single copy of Linux, it can consume many CPU cycles as the number of virtual servers goes up. A rule of thumb on System z is to turn off this unless the server has a heavy, constant workload.

Turning off the hz_timer can be accomplished by adding a `sysctl` command to the file `/etc/init.d/boot.local` which is run each time the virtual server is booted:

```
# vi /etc/init.d/boot.local // add one line at the bottom
#!/bin/sh
...#
# Here you should add things, that should happen directly after booting
# before we're going to the first run level.
sysctl -w kernel.hz_timer=0

```

Before shutting down, note that the hz_timer is on:
When the system is rebooted, the hz_timer, or pop, should be off.

### 7.5.13 Modifying zipl.conf

When a SLES 10 system boots there are two ways it can determine which disks to bring online.

1. The default method is that the device addresses are built into the initial RAMdisk.
2. The disks to bring online can also be written to the zipl configuration file, /etc/zipl.conf after which the zipl command must be run.

It is recommended that you use the second method so it will be easier to determine which minidisks are designed to be part of the system. Is is much easier to read a file (/etc/zipl.conf) than to find where the disks are specified in the initial RAMdisk.

The dasd= parameter must be added to parameter line. When specifying it you must maintain the ordering of the existing minidisks (/dev/dasda = 100, /dev/dasdb = 101 and /dev/dasdc = 102). Make a backup copy of the file and add the string dasd=100-102 to the parameter line in the [ipl] section:

```bash
# cd /etc
# cp zipl.conf zipl.conf.orig
# vi zipl.conf  // add the dasd= string
# Modified by YaST2. Last modification on Sun Oct 29 17:18:11 UTC 2006
[defaultboot]
defaultmenu = menu

:menu
target = /boot/zipl
timeout = 10
prompt = 1
1 = ipl
2 = failsafe
default = 1

###Don't change this comment - YaST2 identifier: Original name: ipl###

[ipl]
target = /boot/zipl
image = /boot/image
ramdisk = /boot/initrd,0x1000000
parameters = "root=/dev/dasda1 dasd=100-102 TERM=dumb"
...
```

Now, run the zipl command so the changes are written to the boot record:

```bash
# zipl
Using config file '/etc/zipl.conf'
Building bootstrap in '/boot/zipl'
Building menu 'menu'
Adding #1: IPL section 'ipl' (default)
Adding #2: IPL section 'failsafe'
Preparing boot device: dasda (0100).
Done.
```
7.5.14 Rebooting the system and verifying changes

You are now done customizing the master Linux image. Now `reboot` to test your changes:

```bash
# reboot
```

Broadcast message from root (pts/0) (Thu Jan  4 12:54:36 2007):

The system is going down for reboot NOW!

When the system comes back up you should verify the changes that you made.

1. **SSH back into the master image** and check a few settings.
2. Use the `df` command to display your file systems. Note that out of a 2.3GB root file system you are using about 38% of it (your output may differ):

```bash
# df -h
```

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>Size</th>
<th>Used</th>
<th>Avail</th>
<th>Use%</th>
<th>Mounted on</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/dasda1</td>
<td>2.3G</td>
<td>822M</td>
<td>1.4G</td>
<td>38%</td>
<td>/</td>
</tr>
<tr>
<td>udev</td>
<td>247M</td>
<td>72K</td>
<td>247M</td>
<td>1%</td>
<td>/dev</td>
</tr>
</tbody>
</table>

3. Confirm that both of your swap spaces are operational:

```bash
# swapon -s
```

<table>
<thead>
<tr>
<th>Filename</th>
<th>Type</th>
<th>Size</th>
<th>Used</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/dasdb1</td>
<td>partition</td>
<td>259956</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>/dev/dasdc1</td>
<td>partition</td>
<td>519924</td>
<td>0</td>
<td>-2</td>
</tr>
</tbody>
</table>

Device with higher priority will be used first. You can also specifically use the `pri=nn` parameter in the `/etc/fstab` file for swap devices.

4. Verify that the `hz_timer` is off:

```bash
# cat /proc/sys/kernel/hz_timer
```

5. Shutdown your master image from the SSH session. Next you will install the controller image.

```bash
# shutdown -h now
```

Congratulations! You have now successfully installed the master image. This image will normally be shut down. It is now time to install the controller Linux image which will normally be running.

7.6 Installing the controller

In this section you will install the controller Linux image. Logon to SLES10 or resume your 3270 session. If you are still logged on, you should see messages that result from the master image being shut down.

1. At this point you could IPL CMS run the SLES10 EXEC to purge the reader, punch the bootstrap files and IPL from the reader, but you’d be purging and punching the same three files. You can save some time by simply IPLing from the reader at virtual address 00c with the `#CP` prefix. Think of it as booting your PC from a floppy disk:

```bash
```
4. Accept the license agreement
5. Click Configure DASD disks. At the DASD Disk Management window, the configuration is different:
   a. Select all nine read-write disks (100-102, 200-207).
   b. Click Activate as shown on the left side of Figure 7-10 on page 115. The DASD will be activated quickly.
   c. Deselect 100, 101 and 102 so that six disks, 200-207, are selected. Be sure you do not format 100 as this will destroy the work you did installing the master image.
   d. Click Format as shown on the right side of Figure 7-10 on page 115.
   e. Click OK to the query to format 6 parallel disks and then Yes to confirm. The DASD will be formatted in parallel and will take a number of minutes.
   f. Click Next when the formatting is complete.
   g. In the Disk Activation window click Next.

6. Accept the default of New installation and Click OK.
7. Select your time settings and click Next.
8. In the Installation Settings window click Partitioning and the Expert Partitioner window will appear. Following is a summary of the DASD partitioning formatting:

<table>
<thead>
<tr>
<th>minidisk</th>
<th>Linux device</th>
<th>Format</th>
<th>Mount Point</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>/dev/dasda1</td>
<td>No!</td>
<td>/sles10master</td>
<td>Master image - don’t format!</td>
</tr>
<tr>
<td>101</td>
<td>/dev/dasdb1</td>
<td>No</td>
<td>swap</td>
<td>Should be detected as a swap space</td>
</tr>
<tr>
<td>102</td>
<td>/dev/dasdc1</td>
<td>No</td>
<td>swap</td>
<td>Should be detected as a swap space</td>
</tr>
</tbody>
</table>
a. Select /dev/dasda1 and click Edit. Select a mount point of /sles10master for /dev/dasda1 as shown in the following figure. Do not format the disk! Again, doing so would destroy all the work you did on the master image.

![Figure 7-11 Mounting /dev/dasda1 over /sles10master](image1)

b. You can skip /dev/dasdb1 and /dev/dasdc1 as they should be recognized as swap spaces. The SWAPGEN EXEC writes the swap signature to them.

c. Select /dev/dasdd and click the Create button. You should see the window shown in the left side of Figure 7-12.

d. Select the /dev/dasdd radio button and click OK. You should see the window in the center of Figure 7-12. Select the Format check box and choose an Ext3 file system. Set the mount point to /backup. Click OK.

![Figure 7-12 Partitioning /dev/dasdd to mount over /backup/](image2)

e. Select /dev/dasde and click Create. Again Format as an Ext3 file system. Set the mount point to /backup. Click OK.
f. Select `/dev/dasdf` and click the `dasdfmt` button. Click **Yes** on the confirmation window. This will create a `/dev/dasdf1` partition (fdasd may be a more accurate button name). You will be prompted to confirm. Click **Yes**.

g. Repeat the above step for `/dev/dasdg`, `/dev/dasdh` and `/dev/dasdi`. When you are done, the summary should be the same as the following figure.

![Expert Partitioner](image)

**Figure 7-13** Partitioning before creating an LVM

h. Click the **LVM** button. A window **Create a Volume Group** will appear. Accept the default **Volume Group name** of `system` and **Physical Extent Size** of **4M**. Click **OK**.

i. In the **Physical Volume Setup** window as shown in Figure 7-14 on page 118, select `/dev/dasdf1` and click the **Add Volume** button.

j. Repeat the previous step to add the three remaining volumes `/dev/dasdg1`, `/dev/dasdh1` and `/dev/dasdi1`. Click **Next**.
**Figure 7-14 Physical Volume Setup**

k. In the *Logical Volumes* window as shown in Figure 7-15 on page 118, click **Add**.

**Figure 7-15 Logical Volumes window**
l. You should see the *Create Logical Volume* window as shown in Figure 7-16 on page 119. Set the *Logical volume name* to *nfs*. Click the button *max* so it gets the full 9.1GB. Select *Format* as *File system* type of *ext2* (a journalled file system is probably not necessary as it will seldom be written to). Specify a *Mount Point* of */nfs*. Click *OK*.

![Figure 7-16 Create Logical Volume window](image)

m. Back in the *Logical Volumes* window, click *Next*.

n. You should see a summary as shown in the next figure.

![Figure 7-17 Partitioning summary](image)
o. Click Finish in the Expert Partitioner window.


10. In the Installation Settings window, click Accept.

11. On the Confirm Installation window, click Install. Copying the RPMs should take about 5-10 minutes depending on network and disk speeds. When copying of the RPMs is done, your VNC viewer session will stop and may close.

**Completing the controller installation**

1. Go back to your 3270 session. After the initial system is shutdown, you must IPL from minidisk 200 (not 100 which is the master image).

```
... 01: HCP6SP2629I The virtual machine is placed in CP mode due to a SIGP stop from CPU 00.
00: HCP6SP2630I The virtual machine is placed in CP mode due to a SIGP stop and store status from CPU 00.
===> #CP IPL 200
...```

2. Start the VNC viewer session again. You should see more RPMs being added.

3. On the Host and Domain Name window, set the Hostname and Domain Name fields appropriately. Uncheck the box Change Hostname via DHCP. Click Next.

4. On the Password for root user window, set the root password and click Next.

5. On the Network Configuration window, disable the firewall and click Next.

6. On the Test Internet Connection window, select No, Skip This Test and click Next.

7. On the Installation Settings window, accept the defaults and click Next.

8. On the User Authentication Method window, accept the default and click Next.

9. On the Add a new local user window, add a non-root user and click Next.

10. Accept all defaults to complete the installation. When your system is completely installed, you can go DISCONNECT from the 3270 session using the #CP prefix.

```
===> #cp disc
```

You have now installed the master image.

**7.6.1 Verifying the installation**

Start a new SSH session to the controller. You may see a warning from PuTTY about a “POTENTIAL SECURITY BREACH”. This is expected because a new set of SSH keys were generated for the same IP address the second time you installed. Click Yes to begin the session.

Verify some settings via the mount and swapon commands. You should see the following file systems mounted and the two swap spaces:

```
# mount
/dev/dasdd1 on / type ext3 (rw,acl,user_xattr)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
debugfs on /sys/kernel/debug type debugfs (rw)
udev on /dev/tmpfs type tmpfs (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
```
7.7 Configuring the controller

Now that your controller is installed, it must be configured. The following steps are involved:

- “Copying files to the controller” on page 121
- “Adding additional RPMs” on page 122
- “Configure the VNC server” on page 122
- “Preparing for Online Update” on page 122
- “Removing unnecessary RPMs” on page 122
- “Turning off unneeded services” on page 122
- “Applying service if necessary - online update” on page 122
- “Configuring sitar” on page 123
- “Installing the cmsfs package” on page 123
- “Turning on the NFS server” on page 124
- “Cleaning temporary files” on page 124
- “Prepare special files in master” on page 124
- “Turning on the NTP server” on page 124
- “Enabling the vmcp module” on page 126
- “Setting system to halt on SIGNAL SHUTDOWN” on page 126
- “Turning off the hz_” on page 127
- “Configuring SSH keys” on page 127
- “Configuring Apache for DAZL” on page 127
- “Setting ownership of Linux backup directories” on page 130
- “Rebooting the system” on page 130
- “Verifying the changes” on page 130

7.7.1 Copying files to the controller

There are many files associated with this book which should be installed the controller. They are now packaged as an RPM. Use the `scp` command to copy package from your NFS server:

```
# scp <129.40.46.206>:/nfs/virt-cookbook-S10/virtcook* /tmp
```

The authenticity of host '129.40.46.206 (129.40.46.206)' can't be established.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '129.40.46.206' (RSA) to the list of known hosts.
Password:
```
virtcook-controller-1-S10.s390x.rpm           100%   24KB  24.0KB/s   00:00
```

Now, install the RPM via the `rpm -i` command:

```
# rpm -i /tmp/virtcook-controller-1-S10.s390x.rpm
```

The files installed will be addressed in the sections that follow:

```
/backup/linux/                      An empty directory for Linux backups
/backup/vm/                        An empty directory for z/VM configuration file backups
/etc/cron.daily/nightly.sh        A script to run backup
```
/sbin/clone.sh
/sbin/backupLinux.sh
/sbin/backupVM.sh
/srv/www/*

7.7.2 Adding additional RPMs
It is recommended that the same packages as you did on the master image. Again use the yast -i command to install these packages:

```
# yast -i findutils-locate openmotif xterm
```
You may choose to add more or fewer RPMs.

7.7.3 Configuring the VNC server
Often applications require a graphical environment. The tightvnc package is a Virtual Network Computing (VNC) server. It allows for a graphical environment to be set up easily via the vncserver command.

Follow the same steps as described in 7.5.3, “Configuring the VNC server” on page 105, with a small difference, you don’t have to remove /root/.vnc/passwd file at the end.

7.7.4 Preparing for Online Update
This step is optional but recommended. You must have Internet access from the installed image, otherwise you can skip this step.

Follow the steps described in 7.5.4, “Preparing for Online Update” on page 107.

7.7.5 Removing unnecessary RPMs
It is recommended to remove the following RPMs as you did on the master image. You may choose to add to or omit some from the list. Follow the same steps as you did in “Removing unnecessary RPMs” on page 107.

7.7.6 Turning off unneeded services
There are a number of services which are started in a SLES 10 minimum system. It is recommended that you turn them off as you did for master image in “Turning off unneeded services” on page 108.

7.7.7 Applying service if necessary - online update
This is the same step done for the master image, so refer to 7.5.8, “Applying service - online update” on page 109.

When you finish, stop the VNC server:

```
# vncserver -kill :1
```
If you do apply service remember to run mkinitrd and zipl if these steps are necessary. Don’t reboot yet, you will reboot shortly after some zipl changes will be made.
7.7.8 Configuring sitar

Follow the same steps as described in 7.5.9, “Configuring sitar” on page 111.

7.7.9 Installing the cmsfs package

The clone.sh script requires the cmsfs package, written by Rick Troth, in order read CMS files. cmsfs package is part of SLES 10 distribution. Install it via the yast -i command:

```
# yast -i cmsfs
```

To test that the cmsfs package is properly installed, see if you can read SLES10’s parameter file. First you need to bring the 191 disk online via the chccwdev -e command. Then view the DASD that the system knows about via the 1sdasd command:

```
# chccwdev -e 191
Setting device 0.0.0191 online
Done
```

Test some of the CMSFS utilities. The cmsfs1st command lists files on a minidisk:

```
# cmsfs1st -d /dev/dasdj
```

```
FILENAME FILETYPE FM FORMAT LRECL RECS BLOCKS DATE TIME
DIRECTOR P0 F 64 19 1 1/04/2007 8:37:06
ALLOCMAP P0 F 4096 2 2 1/04/2007 8:37:06
CHPW52 XEDIT B1 V 70 180 3 1/04/2007 5:50:42
CPFFORMAT EXEC B1 V 79 231 3 1/04/2007 5:50:42
LABEL520 EXEC B1 V 75 108 2 1/04/2007 5:50:42
LABEL520 XEDIT B1 V 71 41 1 1/04/2007 5:50:42
PROFILE EXEC D1 V 63 26 1 1/04/2007 8:37:06
RHEL5 CONF-RHS B1 V 38 13 1 1/04/2007 5:50:42
RHEL5 EXEC B1 V 67 9 1 1/04/2007 5:50:42
RHEL5 PARM-RHS B1 V 80 3 1 1/04/2007 5:50:42
SLES10 EXEC D1 V 68 9 1 1/04/2007 5:53:32
SLES10 INITRD B1 F 80 103269 2017 1/04/2007 8:26:16
SLES10 KERNEL B1 F 80 75579 1122 1/04/2007 8:24:44
SLES10 PARM-S10 D1 V 67 9 1 1/04/2007 8:32:47
SLES9X EXEC B1 V 74 9 1 1/04/2007 5:50:42
SLES9X PARMFILE B1 V 73 8 1 1/04/2007 5:50:42
SWAPGEN EXEC B1 V 72 358 5 1/04/2007 5:50:42
```

The cmsfs1st command types the contents of a file:

```
# cmsfs1st -d /dev/dasdj -a sles10.parm-s10
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
HostIP=129.40.178.130 Hostname=lat130.pbm.ihost.com
Gateway=129.40.178.254 Netmask=255.255.255.0
Broadcast=129.40.178.255 Layer2=0
ReadChannel=0.0.0600 WriteChannel=0.0.0601 DataChannel=0.0.0602
```
7.7.10 Turning on the NFS server

The NFS server will be used to export the SLES 10 install tree. Both the nfslock and the nfsserver services are needed. Turn on these two services with the `chkconfig` command:

```
# chkconfig nfsserver on
```

When you restart the controller, the NFS lock daemon and server will be started.

7.7.11 Cleaning temporary files

Now is the best time to clean temporary files from master image. If they stay there they will be copied to each clone and occupy space there. There are two directories which can be cleaned up:

```
# rm -rf /sles10master/tmp/*
# rm -f /sles10master/var/log/YaST2/*
```

*Note:* Remember to run these commands after any changes to master image.

7.7.12 Prepare special files in master

There are some special files in `/dev/` directory which are created during boot process. For this reason they are not available in `/sles10master/`. This is a good time to prepare them for future use in chrooted environment.

```
# chroot /sles10master
# mknod -m 644 /dev/random c 1 8
# mknod -m 644 /dev/urandom c 1 9
# mknod /dev/loop0 b 7 0
# exit
```

These commands will prepare three special device files in `/dev` directory. `/dev/random` and `/dev/urandom` are needed for SSH services in a chrooted environment. `/dev/loop` is needed for mounting ISO files loopback in section 8.3.1, “Changing source of installation on master image” on page 133.

7.7.13 Turning on the NTP server

It is desirable to have all the system clocks on your Linux images be the same. This can be accomplished by setting up a time server on the controller, and have the master image and therefore all the virtual servers synchronize their times against the controller. The controller's time can be set accurately by having it synchronize against public time servers on the Internet. If you don't want your controller to communicate with the outside world then don't set up the controller to sync with a public time server. If you do want to access an external public time server, then make sure your firewall allows traffic through between the controller and the public time servers on the well-known NTP port, 123.

The standard open source time server on Linux is the Network Time Protocol (NTP) daemon. The controller's NTP daemon needs to synchronize its time with public NTP servers.
There are two methods of utilizing public NTP servers on Internet. The traditional method was to choose random servers from public list and your system would be synchronized against only those servers. Now there is a new method of using a pool of NTP servers. This new method is recommended to avoid public server abuse.

**Note:** You can read more about public server abuse on the Web at:

http://groups.google.com/group/comp.protocols.time.ntp/browse_frm/thread/fc51c6ee6ad66f66/9642ee4940b9982d#9642ee4940b9982d

**Pool of NTP servers**

DNS names of [012].pool.ntp.org point to set of random NTP servers throughout the world that change every hour. You can read more about this on the Web at:

http://www.pool.ntp.org/

Test to see that the pool of servers is indeed accessible by running ntpdate. In the following example, two servers are used:

```
# ntpdate pool.ntp.org
5 Jan 09:47:24 ntpdate[7709]: step time server 89.212.6.198 offset -60.992462 sec
```

The ntpd configuration file is /etc/ntp.conf. You have to make two changes there. Comment out the two lines specifying the local clock settings in the default configuration and add the three external time sources (0.pool.ntp.org, 1.pool.ntp.org, 2.pool.ntp.org). It is recommended that you backup the original file.

```
# cd /etc
# cp ntp.conf ntp.conf.orig
# vi ntp.conf
...
## Undisciplined Local Clock. This is a fake driver intended for backup
## and when no outside source of synchronized time is available.
##
## server 127.127.1.0            # local clock (LCL)
## fudge  127.127.1.0 stratum 10 # LCL is unsynchronized
##
## Outside source of synchronized time
##
## server xx.xx.xx.xx           # IP address of server
## 0.pool.ntp.org
## 1.pool.ntp.org
## 2.pool.ntp.org
```

Start the ntpd server on the controller with the rcntp command:

```
# rcntp start
Try to get initial date and time via NTP from 0.pool.ntp.org 1.pool.ntp.org done
Starting network time protocol daemon (NTPD) done
# chkconfig ntp on

If you immediately run ntptrace, you might see that your time server is stratum 16 which means that the clock is not set accurately:

```
# ntptrace
localhost: stratum 16, offset 0.000000, synch distance 0.001170
```
After you start `ntpd`, you will have to wait some time for the clock to be adjusted. Use the `ntpq` command with the `peers` subcommand to show the two NTP servers that were set in the `/etc/ntp.conf` file (your results will vary):

```
# ntpq
ntpq> peers
remote refid st t when poll reach delay offset jitter
==============================================================================
alien.astrosmo.38.14.160.172   2 u   61 64   3 124.862  2.762  3.920
163.25.109.18   140.112.4.189    2 u   61   64   3 301.322 -51.900  3.696
uicsgtw.cs.ui.a .INIT.         16 u    -   64    0    0.000  0.000 4000.000
ntpq> quit
```

You will see different IP addresses because pool of servers is random. After a few minutes (can take up to half an hour), the `ntptrace` command should show that your system is now a stratum 2 server:

```
# ntptrace
localhost: stratum 2, offset -0.002555, synch distance 0.006114
```

### 7.7.14 Enabling the `vmcp` module

The `vmcp` module/command allows z/VM CP commands to be issued from Linux. It is critical to the functioning of the `clone.sh` script.

To enable it, edit the file `/etc/sysconfig/kernel` and add `vmcp` to the variable `MODULES_LOADED_ON_BOOT` (around line 30):

```
# cd /etc/sysconfig
# vi kernel // add vmcp to MODULES_LOADED_ON_BOOT
...
## Type:                string
## ServiceRestart:      boot.loadmodules
#
# This variable contains the list of modules to be loaded
# once the main filesystem is active
#
MODULES_LOADED_ON_BOOT="vmcp"
...
```

Save the file and you should be able to issue CP commands via the `vmcp` after your system is rebooted.

### 7.7.15 Setting system to halt on SIGNAL SHUTDOWN

The Ctrl-Alt-Del key sequence is simulated by z/VM when it issues a `SIGNAL SHUTDOWN` command. Rather than rebooting, you want your system to halt (shutdown). Change this setting by changing `shutdown -r` to `shutdown -h` in the `/etc/inittab` file:

```
# cd /etc
# vi inittab // change shutdown -r to shutdown -h
...
# what to do when CTRL-ALT-DEL is pressed
ca::ctrlaltdel:/sbin/shutdown -h -t 4 now
...
```

This change will be picked up when the system is rebooted.
7.7.16 Turning off the hz_timer

From the existing SSH session, turn the hz_timer off so the controller doesn’t consume unnecessary CPU cycles:

```bash
# vi /etc/init.d/boot.local // add one line at the bottom
#!/bin/sh
...
# Here you should add things, that should happen directly after booting
# before we're going to the first run level.
sysctl -w kernel.hz_timer=0
```

7.7.17 Configuring SSH keys

SSH sessions are typically authenticated via passwords typed in from the keyboard. With SSH key-based authentication sessions can be authenticated via public and private keys so that no password is needed. To accomplish this, the following must be true:

- The SSH server system must have the client’s public key.
- The SSH client must send its private key.
- The keys must match cryptographically.

SSH key-based authentication can be set up from the controller (client) to the virtual servers. If the master image has a copy of controller’s public key in the file `/root/.ssh/authorized_keys`, and the controller has a symbolic link to its private key in the file `/root/.ssh/id_dsa`, then key based authentication will work to the cloned virtual servers.

Copy the controller’s public key to the master image’s `authorized_keys` file:

```bash
# cd /etc/ssh
# cp ssh_host_dsa_key.pub /sles10master/root/.ssh/authorized_keys
```

Make a symbolic link from `/root/.ssh/id_dsa` to the controller’s private key:

```bash
# cd /root/.ssh
# ln -s /etc/ssh/ssh_host_dsa_key id_dsa
```

7.7.18 Configuring Apache for DAZL

A working Web server is needed for the Web application named Data About z/VM and Linux (DAZL). For details on this application, see section 13.5, “Monitoring z/VM and Linux via a Web application” on page 211. Because this application exposes sensitive data, the entire Web site should be secured. To accomplish this, a valid user ID/password pair should be required to get to the Web site.

Installing Apache

First, install the Apache RPMs via the `yast -i` command:

```bash
# yast -i apache2-prefork apache2 apache2-doc apache2-example-pages
```

You will see `yast` menus go by as the Apache RPMs are installed. When it is complete you can confirm the RPMs have been added via the `rpm -qa` command:

```bash
# rpm -qa | grep apache
apache2-prefork-2.2.0-21.7
apache2-example-pages-2.2.0-21.2
apache2-2.2.0-21.7
apache2-doc-2.2.0-21.2
```
Securing the Web site

Now that the Apache RPMs are installed, it is recommended that you secure the Web site because the DAZL application exposes much sensitive data. Create a password file that Apache will use. The `htpasswd2` command is used to create a password file named `.htpasswd` in Apache's default data directory, `/srv/www/htdocs/`. In this example the user ID root and the root password are used.

```
# cd /srv/www/htdocs
# htpasswd2 -c .htpasswd root
New password: <lnx4vm>
Re-type new password: <lnx4vm>
Adding password for user root
```

Once the password file is created, the configuration file `/etc/apache2/default-server.conf` is modified so the DocumentRoot Directory statement will require valid credentials in this password file:

```
# cd /etc/apache2
# cp default-server.conf default-server.conf.orig
# vi default-server.conf   // Add 4 lines to require correct user ID and password
...
# Configure the DocumentRoot
#
<Directory "/srv/www/htdocs">
    # Possible values for the Options directive are "None", "All",
    # or any combination of:
    # Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
    # Note that "MultiViews" must be named *explicitly* --- "Options All"
    # doesn't give it to you.
    # The Options directive is both complicated and important. Please see
    # http://httpd.apache.org/docs-2.2/mod/core.html#options
    # for moreinformation.
    Options None
    # AllowOverride controls what directives may be placed in .htaccess files.
    # It can be "All", "None", or any combination of the keywords:
    # Options FileInfo AuthConfig Limit
    AllowOverride None
    # add 4 - make this a restricted Web site
    AuthType Basic
    AuthName "This is a restricted Web site"
    AuthUserFile /srv/www/htdocs/.htpasswd
    Require user root
    # Controls who can get stuff from this server.
    Order allow,deny
    Allow from all
</Directory>
```

Allowing Apache to run CP and SSH commands

The DAZL application uses CP commands via the `vmcp` command to get z/VM data, and the `ssh` command with key-based authentication to get Linux data. By default the `www` group that Apache runs under cannot issue these commands. Therefore, `sudo` is used to allow this access. Modify the `/etc/sudoers` configuration file to allow this access:

```
# cd /etc
# cp sudoers sudoers.orig
# vi sudoers   // add three lines at the bottom
...
```
# Samples

```bash
# %users  ALL=/sbin/mount /cdrom,/sbin/umount /cdrom
# %users  localhost=/sbin/shutdown -h now

# allow www group to run vmcp and ssh so Apache2 can populate DAZL pages
%www    ALL=NOPASSWD:/sbin/vmcp
%www    ALL=NOPASSWD:/usr/bin/ssh
```

Now Apache should be able to run `vmcp` and `ssh` commands via `sudo`.

### Starting Apache

Start Apache for this session with the `rcapache2` service script. Also, use the `chkconfig` command so that the Apache Web server starts when the system is booted:

```bash
# rcapache2 start
Starting httpd2 (prefork) done
# chkconfig apache2 on
```

Now the Apache Web server will start and will require the root user ID and password to get access. Point a browser to the IP address of the controller. You should be prompted for credentials. The following figure shows the prompt window from a FireFox browser:

![Figure 7-18 Browser prompting for credentials](image)

When you supply the correct credentials, you should see Apache’s default `index.html` page:

![Figure 7-19 Default Apache index.html page](image)

Backup the original `index.html` file and copy the `index.html` and two other files from the `virtcook-controller-S10 RPM` over it. They are located in the `/srv/www/html/` directory so as to avoid an RPM conflict. Doing so will make the DAZL main page the default:

```bash
# cd /srv/www/htdocs/
# mv index.html index.html.orig
# cp ../html/* .
```
Go back to your browser and click Refresh. The DAZL logon screen should be presented. You should now have Apache configured for DAZL. Before you create any clones, there is not a lot that this application will do for you. This application is described later in section 13.5, “Monitoring z/VM and Linux via a Web application” on page 211.

### 7.7.19 Setting ownership of Linux backup directories

In order for DAZL to be able to update the description/log files of each virtual server, the Apache Web server will need permission to write to those files. Apache runs as the group named www so one way to allow this is via the group write and set-uid bits. Use the following chgrp and chmod commands to accomplish this:

```bash
# cd /backup
# chgrp www linux
# chmod g+ws linux
```

The `chgrp` command sets the group owner of the directory `/backup/linux/` to `www`. Setting the group write (g) bit gives that group write permissions and setting the group set-uid (s) bit pushes that permission down to newly created directories.

### 7.7.20 Rebooting the system

You should now reboot the system to test the changes:

```bash
# reboot
```

After your system comes back in a couple of minutes, start a new SSH session to the controller.

### 7.7.21 Verifying the changes

You are now done customizing the controller Linux image. SSH back into the controller and check a few settings. Test the `vmcp` command with a CP command such as `QUERY NAMES`:

```bash
# vmcp q n
FTPSERVE - DSC , DTCVSW2 - DSC , DTCVSW1 - DSC , TCP/IP - DSC
OPERSYMP - DSC , DISKACNT - DSC , EREP - DSC , OPERATOR - DSC
SLES10 - DSC
VSM - TCP/IP
```

Confirm that both of your swap spaces are operational:

```bash
# swapoff -s
```

Verify that the `hz_timer` is off:

```bash
# cat /proc/sys/kernel/hz_timer
0
```

Verify the NFS server is running:

```bash
# rcnfsserver status
Checking for kernel based NFS server: running
```

Congratulations! You have installed and configured Linux twice onto the SLES10 user ID. You are now ready to configure NFS on the controller.
Chapter 8. Configuring NFS on controller

Imagination is more important than knowledge.
--Albert Einstein

The SLES10 user ID is now customized with both a master image and a controller. The controller should have a 9GB logical volume ext2 file system mounted over /nfs/. It can now be configured to replace the NFS (PC) server to make the SLES10 install tree and the files associated with this book available via NFS.

The following steps are involved in configuring NFS on the controller:

- "Copying files from NFS server to controller" on page 131
- "Configuring the NFS server" on page 132
- "Changing the YaST installation location" on page 133

8.1 Copying files from NFS server to controller

In this section you will copy the SLES 10 ISO images and the files associated with this book from the NFS server to the controller's logical volume file system mounted over the directory /nfs/.

8.1.1 Copying the SLES10 ISO images

This section assumes you have already set up the SLES10 install tree on another server as described in 6.2, “Setting up a SLES 10 install server” on page 84. You can copy the tree recursively then configure the NFS server to export it.

Open or continue an SSH session on the controller. Copy all SLES10 install ISO images using the `scp -rp` command. You will need to type the root password of the NFS server:

```
# cd /nfs
# ls
... lost+found
```

This shows that the file system is empty except for the `lost+found/` directory which is where a file system stores damaged files.
Copy the SLES10 ISO images from the NFS server to the `/nfs/` directory:

```
# scp -rp <129.40.46.206>:/nfs/sles10 /nfs
Password: vmrdr.ikr
```

<table>
<thead>
<tr>
<th>Hostname</th>
<th>Size</th>
<th>Speed</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>initrd</td>
<td>8068KB</td>
<td>7.9MB/s</td>
<td>00:01</td>
</tr>
<tr>
<td>SLES-10-s390x-GMC.NOTES</td>
<td>16KB</td>
<td>15.7KB/s</td>
<td>00:00</td>
</tr>
<tr>
<td>SLES-10-CD-s390x-GMC-CD1.iso</td>
<td>656MB</td>
<td>10.8MB/s</td>
<td>01:01</td>
</tr>
<tr>
<td>SLES-10-CD-s390x-GMC-CD2.iso</td>
<td>626MB</td>
<td>10.4MB/s</td>
<td>01:00</td>
</tr>
<tr>
<td>SLES-10-CD-s390x-GMC-CD3.iso</td>
<td>394MB</td>
<td>7.6MB/s</td>
<td>00:52</td>
</tr>
<tr>
<td>SLES-10-CD-s390x-GMC-CD4.iso</td>
<td>552MB</td>
<td>8.1MB/s</td>
<td>01:08</td>
</tr>
</tbody>
</table>

This step should take about 10-20 minutes depending on network and disk speed. Look at the newly copied directory.

```
# ls -a
. .. lost+found sles10
# du -sh sles10/
2.2G    sles10/
```

This shows that it occupies about 2.2 GB on disk.

### 8.1.2 Copying the files associated with this book

Now recursively copy the files associated with this book that were untarred to `/nfs/` on the NFS server:

```
# scp -rp <129.40.46.206>:/nfs/virt-cookbook-S10 /nfs
```

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Speed</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHPW52.XEDIT</td>
<td>8820</td>
<td>8.6KB/s</td>
<td>00:00</td>
</tr>
<tr>
<td>RHEL5.PARM-RH5</td>
<td>243</td>
<td>0.2KB/s</td>
<td>00:00</td>
</tr>
<tr>
<td>SLES10.PARM-S10</td>
<td>473</td>
<td>0.5KB/s</td>
<td>00:00</td>
</tr>
<tr>
<td>SLES9X.PARMFILE</td>
<td>420</td>
<td>0.4KB/s</td>
<td>00:00</td>
</tr>
<tr>
<td>RHEL5.CONF-RH5</td>
<td>255</td>
<td>0.3KB/s</td>
<td>00:00</td>
</tr>
<tr>
<td>SLES10.EXEC</td>
<td>248</td>
<td>0.2KB/s</td>
<td>00:00</td>
</tr>
<tr>
<td>PROFILE.EXEC</td>
<td>755</td>
<td>0.7KB/s</td>
<td>00:00</td>
</tr>
<tr>
<td>LABELS20.XEDIT</td>
<td>1667</td>
<td>1.6KB/s</td>
<td>00:00</td>
</tr>
<tr>
<td>SWAPGEN.EXEC</td>
<td>16KB</td>
<td>16.0KB/s</td>
<td>00:00</td>
</tr>
<tr>
<td>RHEL5.EXEC</td>
<td>247</td>
<td>0.2KB/s</td>
<td>00:00</td>
</tr>
<tr>
<td>CPFORMAT.EXEC</td>
<td>9755</td>
<td>9.5KB/s</td>
<td>00:00</td>
</tr>
<tr>
<td>LBLS20-9.EXEC</td>
<td>3697</td>
<td>3.6KB/s</td>
<td>00:00</td>
</tr>
<tr>
<td>LABELS20.EXEC</td>
<td>4259</td>
<td>4.2KB/s</td>
<td>00:00</td>
</tr>
<tr>
<td>SLES9.EXEC</td>
<td>254</td>
<td>0.3KB/s</td>
<td>00:00</td>
</tr>
<tr>
<td>LBLS20-9.XEDIT</td>
<td>1792</td>
<td>1.8KB/s</td>
<td>00:00</td>
</tr>
<tr>
<td>clone.sh</td>
<td>20KB</td>
<td>19.9KB/s</td>
<td>00:00</td>
</tr>
<tr>
<td>sitar-1.0.7-270.1.noarch.rpm</td>
<td>8888</td>
<td>87.6KB/s</td>
<td>00:00</td>
</tr>
<tr>
<td>cmsfs-1.1-8c.s390.rpm</td>
<td>2682</td>
<td>25.6KB/s</td>
<td>00:00</td>
</tr>
<tr>
<td>virtcook-controller-1-s10.s390x.rpm</td>
<td>2454</td>
<td>24.0KB/s</td>
<td>00:00</td>
</tr>
<tr>
<td>README.txt</td>
<td>726</td>
<td>0.7KB/s</td>
<td>00:00</td>
</tr>
<tr>
<td>MD5SUMS.sles10</td>
<td>252</td>
<td>0.3KB/s</td>
<td>00:00</td>
</tr>
</tbody>
</table>

You have used many of these files. They are being copied to the controller so you can retire the PC NFS server.

### 8.2 Configuring the NFS server

Set up the NFS server to export the two directories that you just copied. Add two lines to the `/etc/exports` file as follows:

```
# cd /etc
# vi exports  // add a line at the bottom
```
# See the exports(5) manpage for a description of the syntax of this file.  
# This file contains a list of all directories that are to be exported to  
# other computers via NFS (Network File System).  
# This file used by rpc.nfsd and rpc.mountd. See their manpages for details  
# on how make changes in this file effective.  
/nfs/virt-cookbook-S10 *(ro,sync)  
/nfs/sles10 *(ro,sync)  

The NFS server was set to start when you configured the controller. Double-check that it is running with the `rcnfsserver` command:

```
# rcnfsserver status  
Checking for kernel based NFS server: \nrunning
```

You can now force the `/etc/exports` file to be reread with the command `exportfs -a`:

```
# exportfs -a
```

The NFS server should now be running and customized. Test that you can mount the file systems locally over an empty directory such as `/mnt/` using the DNS name `localhost`, with the following `mount` commands:

```
# ls /mnt  
# mount localhost:nfs/virt-cookbook-S10 /mnt  
# ls -af /mnt  
./ MD5SUMS.sles10 sitar-1.0.7-270.1.noarch.rpm \nvm/  
./ README.txt virtcook-controller-1-S10.s390x.rpm  
# umount /mnt  
# mount localhost:nfs/sles10 /mnt  
# ls -af /mnt  
./ SLES-10-CD-s390x-GMC-CD3.iso  
./ SLES-10-CD-s390x-GMC-CD4.iso  
SLES-10-CD-s390x-GMC-CD1.iso SLES-10-CD-s390x-GMC-CD2.iso SLES-10-CD-s390x-GMC-CD2.iso boot/  
# umount /mnt
```

This shows that you can mount the NFS-exported directories locally.

## 8.3 Changing the YaST installation location

When you installed both the master image and the controller, you set the location of the SLES10 ISO images to the PC NFS server. Now that you have configured an NFS server on the controller, you can reset this location on both the master image and the controller. After doing this you will no longer need the NFS server as you will be able to use the controller to serve the SLES10 ISO images and the files associated with this book.

The steps in this section are as follow:

- “Changing source of installation on master image” on page 133
- “Changing source of installation on controller” on page 135
- “Changing source of installation in parameter file” on page 137

### 8.3.1 Changing source of installation on master image

There is a bit of a paradox in getting the master image to point to the controller for its source of installation. You have to shutdown the controller to bring up the master image, but if the controller is shut down, `yast` will not be able to find the NFS-exported directory.
To work around this issue, you can use the `chroot` command which creates a shell with a new root. After `chroot`ing you will be working on the running controller, but you will be able to modify the master image because that becomes your new root file system. This technique can be used to perform other modifications to the master image, such as adding RPMs (however, this technique does not work for all commands).

1. Enter the `chroot` command to the `/sles10master/` directory and it will be as though you have booted the master image:

```
# chroot /sles10master/
```

**Note:** After you get into chrooted environment check if there is at least one loop device with `ls -l /dev/loop*`. If there is none, create it with `mknod /dev/loop0 b 7 0`.

2. Now you are in a shell as if you were logged into the master image. Enter the `yast` command to change the source of installation.

```
# yast
```

3. Use the Tab key to move to the right side of the YaST Control Center. Use the Down Arrow key three times to select **Installation Source**. Press **Enter**.

```
[Help] [Quit]
```

4. You will be presented with the *Media Containing the Software Catalog* panel shown below. Depending upon whether you configured online sources or not, you will see one or three install sources. Two of them (online sources) are named YUM while remaining one is named SUSE Linux Enterprise Server 10. This is the source used to install the master image and it is the one you have to change.

   a. Use the Tab key to move to the **Edit** button and press **Enter**.

   b. Choose the default **Replace**... button.

   c. This will bring up a small menu entitled **NFS Server Source**. Change the TCP/IP address (or DNS name) to that of the controller. The **ISO image** box should be checked and you should see the **Path** set to:

```
nfs/sles10/SLES-10-CD-s390x-GMC-CD1.iso
```
d. Use Tab key to move to **OK** and press Enter.

e. **License Agreement** panel will be shown. Select **Yes, I Agree to the License Agreement** and then **Next**.

f. Move the **SUSE Linux Enterprise Server 10** to the top position using the **Up** button.

g. When you have the desired installation sources, select **Finish**. You may get several zmd errors, you can ignore them. If you are asked about saving changes, select **Yes**.

5. From the **yast** main menu select **Quit**.

6. Use the **exit** command to leave the **chroot**ed system.

   # exit

   Now any cloned virtual server that is cloned should point to the controller for its source of RPMs.

### 8.3.2 Changing source of installation on controller

Perform the same tasks for the controller, but rather than using NFS as the protocol, just point to the local directories. To do this, you must remove **SUSE Linux Enterprise Server 10** installation source and new one. Invoke **yast** and perform the following steps:

1. Choose **Software => Installation Source** on the **YaST Control Center**.
2. On the Media Containing the Software Catalog panel, choose Add.

3. Choose Local Directory as shown in the following example.

4. Set the directory to:

   /nfs/sles10/SLES-10-CD-s390x-GMC-CD1.iso

5. Check the ISO Image check box and press OK.

6. The License Agreement panel will be shown. Select Yes, I Agree to the License Agreement and then Next.

7. You should now have four entries: three old entries at the top and a new entry at the bottom. Now delete old install source:

   a. Tab to Delete and press Enter.
8. Move the **SUSE Linux Enterprise Server 10** to the top position using the **Up** button.

9. Tab to **Finish** and press Enter.

10. From the **yast** main menu select **Quit**.

### 8.3.3 Changing source of installation in parameter file

There is one more place that the source of the installation has to be changed and that is the parameter file on the **LNXMAINT 192 disk**. Logon to **LNXMAINT** and edit the **SLES10 PARM-S10**.

Change the value of the **Install** variable. In the following example it is set to 129.40.178.130:

```plaintext
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb HostIP=129.40.178.130 Hostname=lat130.pbm.ihost.com Gateway=129.40.178.254 Netmask=255.255.255.0 Broadcast=129.40.178.255 Layer2=0 ReadChannel=0.0.0600 WriteChannel=0.0.0601 DataChannel=0.0.0602 Nameserver=129.40.106.1 Portname=donntcare Install=nfs://<129.40.178.130>/nfs/sles10/SLES-10-CD-s390x-GMC-CD1.iso UseVNC=1 VNCPassword=lnx4vm InstNetDev=osq Os Lairface=qdio Osamedium=eth Manual=0
```

Changing this value is somewhat of a paradox. The parameter file sets both the install server IP address and the Linux IP address to the same value. This will never work as you cannot install onto a system that is also the installation server. However, the sections that follow have you copy this parameter file, so this way it will be copied with the correct value.

If for some reason you do have to reinstall onto SLES10, you will have to set up another source of installation (perhaps the PC NFS server again), and change this value back to point to it.
8.4 Retire the PC NFS server

You have now copied all files related to this project from a PC (or other platform) server to the controller on IBM System z. You should be in a position to retire your PC NFS server.
Configuring SLES 10 for cloning

It has become appallingly obvious that our technology has exceeded our humanity.
--Albert Einstein

The SLES10 user ID is now customized with both a master image and a controller. The controller should now be running. In this chapter, you will perform the following steps:

- “Defining a new user ID for a virtual server” on page 139
- “Cloning a virtual server manually” on page 142
- “Cloning a virtual server automatically” on page 145
- “Cloning six more virtual servers” on page 148

9.1 Defining a new user ID for a virtual server

In this section you will define a new user ID, LINUX01, in z/VM and clone the master image to it.

1. Logon to MAINT and edit the USER DIRECT file to add more Linux ID’s.

=> x user direct c

2. Go to the bottom of the file and add the following four lines. In this example the user ID will be LINUX01 with a password of LNX4VM. A single 3390-3 DASD is used for a 3338 cylinder (about 2.3 GB) root file system. In this example it is at device address A711 which was formatted and given a label of MMA711 earlier:

USER LINUX01 LNX4VM 256M 1G G
   INCLUDE LNXDFLT
   OPTION APPLMON
   MDISK 100 3390 0001 3338 <MMA711> MR LNX4VM LNX4VM LNX4VM

3. Add the new volume to the $ALLOC$ user ID so cylinder 0 won’t show up in the disk map as a gap. Save your changes with the FILE subcommand:

====> top
=====> /alloc
 USER $ALLOC$ NOLOG
   MDISK A01 3390 0000 001 MVA700 R
...
MDISK A0B 3390 000 001 <MMA711> R
====> file

4. Again check for gaps and overlaps. You can use the ALL subcommand with the logical OR operator "|" to check for both strings. You should see only one 501 cylinder gap.

  => diskmap user
  => x user diskmap
  ===> all /gap/|/overlap/

  --------------------  4 line(s) not displayed --------------------
  0         500         501    GAP
  --------------------  368 line(s) not displayed --------------------

  ===> quit

5. Bring the changes online with the DIRECTXA command:

  => directxa user

z/VM USER DIRECTORY CREATION PROGRAM - VERSION 5 RELEASE 2.0
EOJ DIRECTORY UPDATED AND ON LINE

The new Linux user ID has now been defined.

9.1.1 Adding LINUX01 to AUTOLOG1’s PROFILE EXEC

The new Linux ID you defined needs access to the VSWITCH. A SET VSWITCH command with the GRANT parameter can be added to AUTOLOG1’s PROFILE EXEC to do this. Also, an XAUTOLOG statement can be added if the user ID is automatically logged on at z/VM IPL time:

Link and access the AUTOLOG1 191 disk read-write and edit the file PROFILE EXEC. Add LINUX01 to the sections that grant access to the VSWITCH and that automatically start the Linux user IDs:

  => link autolog1 191 1191 mr
  => acc 1191 f
  => x profile exec f   // add two lines
  /***************************/
  /*  Autolog1 Profile Exec */
  /***************************/
  'cp xautolog tcpip'                /* start up TCPIP */
  'CP XAUTOLOG DTCVSW1'             /* start VSWITCH controller 1 */
  'CP XAUTOLOG DTCVSW2'             /* start VSWITCH controller 2 */
  'cp set pf12 ret'                 /* set the retrieve key */
  'cp set mdc stor 0m 128m'         /* Limit minidisk cache in CSTOR */
  'cp set mdc xstore 0m 0m'         /* Disable minidisk cache in XSTOR */
  'cp set srm storbuf 300% 250% 200%' /* Overcommit memory */
  'cp set signal shutdown 180'      /* Allow guests 3 min to shut down */

  /* Grant access to VSWITCH for each Linux user */
  'cp set vswitch vsw1 grant sles10'
  'cp set vswitch vsw1 grant linux01'

  /* XAUTOLOG each Linux user that should be started */
  'cp xautolog sles10'
  'cp xautolog linux01'

  'cp logoff'                        /* logoff when done */
  ===> file

These changes will not take effect until the next IPL, so you must grant this user ID access to the VSWITCH for this z/VM session. This is done as follows:

  => set vswitch vsw1 grant linux01
Chapter 9. Configuring SLES 10 for cloning

9.1.2 Creating a parameter file for the new LINUX ID

For each Linux guest you want to clone, you need to create a parameter file. This file specifies many of the installation parameters. It will be used both when cloning to this user ID and when installing SLES10 manually.

1. Logon to LNXMAINT.
2. Copy an existing parameter file and edit the new file to apply to the new Linux.
   ```
   ==> copy sles10 parm-s10 d linux01 parm-s10 d
   ==> x linux01 parm-s10 d
   ```
3. Edit the new parameter file as you did for SLES10 (see 7.3, “Preparing SLES10 bootstrap files” on page 92). If the new Linux is going to be on the same network as the controller you will likely only have to change two variables: the IP address and the DNS name. In this example the IP address is set to 129.40.178.121 and the DNS name to lat121.pbm.ihost.com:
   ```
   ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
   HostIP=<129.40.178.121> Hostname=lat121.pbm.ihost.com
   Gateway=<129.40.178.254> Netmask=<255.255.255.0>
   Broadcast=<129.40.178.255> Layer2=0
   Nameserver=129.40.106.1 Portname=dontcare
   Install=nfs://<129.40.178.130>/nfs/sles10/SLES-10-CD-s390x-GMC-CD1.iso
   UseVNC=1 VNCPassword=<lnx4vm>
   InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
   ```
4. Logoff of LNXMAINT and logon to LINUX01.
5. Answer N to the question “Do you want to IPL Linux from DASD 100”. Verify that the new Linux user ID has a NIC at addresses 600-602:
   ```
   LOGON LINUX01
   00: NIC 0600 is created; devices 0600-0602 defined
   00: z/VM Version 5 Release 2.0, Service Level 0601 (64-bit),
   00: built on IBM Virtualization Technology
   ...```
6. Verify that the minidisk at addresses 100 and the VDISKs at address 101 and 102 are read-write:
   ```
   ==> q da
   00: DASD 0100 3390 MMA711 R/W  3338 CYL ON DASD A711 SUBCHANNEL = 0000
   00: DASD 0101 9336 (VDSK) R/W  524288 BLK ON DASD VDSK SUBCHANNEL = 0000
   00: DASD 0102 9336 (VDSK) R/W  1048576 BLK ON DASD VDSK SUBCHANNEL = 0000
   00: DASD 0190 3390 MVA700 R/O  107 CYL ON DASD A700 SUBCHANNEL = 0008
   00: DASD 0191 3390 MVA70C R/O  300 CYL ON DASD A70C SUBCHANNEL = 0008
   ...```
7. Logoff LINUX01.

You should now be ready to clone to this new user ID.
9.2 Cloning a virtual server manually

Before using the shell script clone.sh to clone a server, it is recommended that you clone a server manually to better understand the process.

There are many ways to clone Linux under z/VM. The steps in this section are just one way to do it. The following assumptions are made based on what you have done so far:

- The source Linux image has a single root file system on minidisk 100.
- The target user ID, LINUX01 in this example, has an identically sized minidisk 100.
- The vmcp command is available to issue z/VM CP commands.
- The z/VM FLASHCOPY command can be used but if you don’t have that support, the Linux dasdfmt and dd commands will work.

Given these assumptions, one set of steps that can be used to clone a system is as follows:

1. Link the target disk read-write.
2. Copy the source to the target disk via FLASHCOPY or the Linux dd command.
3. Mount the newly copied root file system.
4. Modify the networking information on the target system.
5. Detach the target disk.
6. IPL the target system.
7. Modify the SSH keys on the target system.

**Linking the target disk read-write**

**Start an SSH session** to the controller as root.

The target root file system, LINUX01 100, is linked multi-read (read-write if no other user ID has write access) as virtual device 1100 via the CP LINK command:

```
# vmcp link linux01 100 1100 mr
```

**Copying the source to the target disk via FLASHCOPY**

The two disks are copied via the CP FLASHCOPY command:

```
# vmcp flashcopy 100 0 end to 1100 0 end
Command complete: FLASHCOPY 100 0 END TO 1100 0 END
```
Mounting the newly copied root file system

Disable and enable the new minidisk at virtual device address 1100:

```
# chccwdev -d 1100
Setting device 0.0.1100 offline
Done
# 1sdasd
0.0.0100(ECKD) at (94: 0) is dasda : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0101(FBA ) at (94: 4) is dasdb : active at blocksize 512, 524288 blocks, 256 MB
0.0.0102(FBA) at (94: 8) is dasdc : active at blocksize 512, 1048576 blocks, 512MB
0.0.0200(ECKD) at (94:12) is dasdd : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0203(ECKD) at (94:16) is dasde : active at blocksize 4096, 543240 blocks, 2122 MB
0.0.0204(ECKD) at (94:20) is dasdf : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0205(ECKD) at (94:24) is dasdg : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0206(ECKD) at (94:28) is dasdh : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0207(ECKD) at (94:32) is dasdi : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.1100(ECKD) at (94:36) is dasdj : active at blocksize 4096, 600840 blocks, 2347 MB
```

In this example the source device is /dev/dasda, while the target device is /dev/dasdj. Format the disk via the dasdfmt command then copy it via the dd command using a block size of 4K (4096) bytes:

```
# dasdfmt -b 4096 -y -f /dev/dasdj
...
# dd if=/dev/dasda of=/dev/dasdj bs=4096
600840+0 records in
600840+0 records out
```

In this case, the device node is /dev/dasdj. Because there is only one partition in the root file system, it will be /dev/dasdj1. Mount it over a newly created mount point /mnt/linux01:

```
# mkdir /mnt/linux01
# mount /dev/dasdj1 /mnt/linux01
```
Observe that this appears to be a root file system:

```bash
# ls -a /mnt/linux01/
.  bin  dev  home  lib64  media  opt  root  srv  sys  usr
..  boot  etc  lib  lost+found  mnt  proc  sbin  success  tmp  var
```

**Modifying networking information on the target system**

In this example, the only two pieces of networking information that are modified are the IP address (from 129.40.178.130 to 129.40.178.121) and the host name (from `lat130` to `lat121`). The host name is changed in the file `/etc/HOSTNAME`:

```bash
# cd /mnt/linux01/etc
# vi HOSTNAME
lat121.pbm.ihost.com
```

The IP address is changed in the file `/etc/sysconfig/network/ifcfg-qeth-bus-ccw-0.0.0600`:

```bash
# cd sysconfig/network/
# ls ifcfg-*
ifcfg-lo  ifcfg-qeth-bus-ccw-0.0.0600
# vi ifcfg-qeth-bus-ccw-0.0.0600
BOOTPROTO="static"
UNIQUE=""
STARTMODE="onboot"
IPADDR="129.40.178.121"
MTU="1500"
NETMASK="255.255.255.0"
NETWORK="129.40.178.0"
BROADCAST="129.40.178.255"
```

**Detach the target disk**

Change to the default directory, use the `sync` command to flush the disks and the `umount` command to unmount the modified root file system:

```bash
# cd
# sync
# umount /mnt/linux01
```

Detach the target minidisk via the CP `DETACH` command:

```bash
# vmcp det 1100
DASD 1100 DETACHED
```

**IPL the target system**

You should now be ready IPL the manually cloned system. Logon to a 3270 session as LINUX01. CMS will IPL and the `PROFILE EXEC` will ask you if you want to IPL from minidisk 100:

```
LOGON LINUX01
00: NIC 0600 is created; devices 0600-0602 defined
00: z/VM Version 5 Release 2.0, Service Level 0601 (64-bit),
00: built on IBM Virtualization Technology
00: There is no logmsg data
00: FILES: 0003 RDR, NO PRT, NO PUN
00: LOGON AT 12:46:44 EST FRIDAY 01/05/07
z/VM V5.2.0 2006-07-18 08:48
DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 101 (64989 4K pages of swap space)
DIAG swap disk defined at virtual address 102 (129981 4K pages of swap space)
```
Do you want to IPL Linux from DASD 100? y/n
y
00: zIPL v1.5.3 interactive boot menu
00:
00: 0. default (ipl)
00:
00: 1. ipl
00: 2. failsafe
00:
00: Note: VM users please use 'cp vi vmsg <number> <kernel-parameters>'
00:
00: Please choose (default will boot in 10 seconds):
00: Booting default (ipl)...
Linux version 2.6.16.21-0.25-default (geeko@buildhost) (gcc version 4.1.0 (SUSE Linux)) #1 SMP Tue Sep 19 07:26:15 UTC 2006
We are running under VM (64 bit mode)
...

Your new system should come up cleanly using the modified IP address and host name.

Modify the SSH keys on the target system
Start an SSH session to the new clone as root.

The SSH keys that were copied are identical to those of the master image. Manually create three new ones with the ssh-keygen command:

```
# cd /etc/ssh
# ssh-keygen -t rsa -N "" -q -f ssh_host_rsa_key
ssh_host_rsa_key already exists.
Overwrite (y/n)? y
# ssh-keygen -t dsa -N "" -q -f ssh_host_dsa_key
ssh_host_dsa_key already exists.
Overwrite (y/n)? y
# ssh-keygen -t rsa1 -N "" -q -f ssh_host_key
ssh_host_key already exists.
Overwrite (y/n)? y
```

Congratulations! You have now cloned a Linux system manually. You can look around the new system. It should be identical to the master image except for the IP address and host name.

Next you will learn how to do it automatically. You will use the LINUX01 user ID again. To clone, the target user ID must be logged off. You could shut the new system down cleanly, but because you will be cloning again, it does not matter. Go to the 3270 session and crash the LINUX01 user ID via the LOGOFF command:

```
==>
# cp log
```

### 9.3 Cloning a virtual server automatically

Now that you have cloned a server manually and better understand the steps, you can use the `clone.sh` script to clone automatically.

Start an SSH session to the controller. The `clone.sh` script should be in your PATH in the directory `/sbin/`. You can verify this with the `which` command:

```
# which clone.sh
/sbin/clone.sh
```
The script takes one parameter which is the Linux user ID that the master image will be cloned to. That user ID must be logged off. It reads the parameter file on the LNMAINT 192 disk (the controller’s 191 disk) to obtain information necessary to give the new Linux virtual server an identity. It calls CP FLASHCOPY via the vmcp module/command to try to copy the 100 (master image) disks. If FLASHCOPY fails, the script falls back to copying the disk via the Linux dasdfmt and dd commands. The script then boots the new Linux via the XAUTOLOG command. It also creates an empty backup directory under /backup/linux/ and adds the server’s public key is added to the controller’s known_hosts file.

It should take less than a minute to clone with FLASHCOPY support and 3-20 minutes without it. Following is an example of cloning to the LINUX01 user ID with FLASHCOPY support. The output is divided into sections.

```
# clone.sh linux01
clone.sh linux01
  Invoking CP command: QUERY LINUX01
  HCPCQU045E LINUX01 not logged on
  Error: non-zero CP response for command 'QUERY LINUX01': #45
  Setting device 0.0.0191 offline
  Done
  Setting device 0.0.0191 online
  Done
  LINUX01  PARM-S10 D1 V         68          9          1  1/05/2007 12:37:59
  WARNING!!: this will copy disk 100 to LINUX01 100
  New host name will be:  lat121.pbm.ihost.com
  New TCP/IP address will be: 129.40.178.121
  Other network data is retrieved from LINUX01 PARM-S10 on 191 disk
  Are you sure you want to overwrite these disks (y/n): y
  Please enter a brief description of this server
  This is a Web Server running SLES 10

In the section of output above, the script makes sure the user ID to be cloned to exists and is logged off. It then searches for the correct PARMFILE and obtains the necessary networking information. It then asks if you are sure you want to overwrite the disks on the target user ID. Then it asks for a one line description of the server which will later be used by DAZL.

```

```
  Copying 0100 root file system to LINUX01 ...
  Invoking CP command: QUERY VIRTUAL 1100
  HCPCQVD040E Device 1100 does not exist
  Error: non-zero CP response for command 'QUERY VIRTUAL 1100': #40
  Invoking CP command: LINK LINUX01 0100 1100 MR
  Invoking CP command: FLASHCOPY 0100 0 END 1100 0 END
  Command complete: FLASHCOPY 0100 0 END TO 1100 0 END
  Invoking CP command: DETACH 1100
  DASD 1100 DETACHED
  Copying disk via FLASHCOPY succeeded ...

In the section above the master image root file system is copied to the LINUX01 100 minidisk via the FLASHCOPY command. Should it fail, the code will fall back to the dasdfmt and dd commands to perform the copies.

```

```
  Mounting newly cloned image over /mnt/sles10cloned ...
  Invoking CP command: LINK LINUX01 100 1100 MR

  Setting device 0.0.1100 online
  Done
  Mounting /dev/dasdk1 over /mnt/sles10cloned ...
  Modifying cloned image under /mnt/sles10cloned ...
  Regenerating SSH keys in /mnt/sles10cloned/etc/ssh/ ...
```
Adding 129.40.178.121 to known_hosts file
Setting device 0.0.1100 offline
Done
Invoking CP command: DETACH 1100
DASD 1100 DETACHED

In the section above, the newly cloned file system (LINUX01 100) is linked, activated and mounted over a temporary directory /mnt/sles10cloned/. Then the networking information is modified in files such as /etc/sysconfig/network/ifcfg-qeth-bus-ccw-0.0.0600 and /etc/HOSTNAME. Then the SSH keys are regenerated so they are unique for the new virtual server. Then the server's public key is added to the controller's /root/.ssh/known_hosts file. Finally the new disk is set offline and detached.

Invoking CP command: XAUTOLOG LINUX01
Command accepted
Creating a directory under /backup/linux
Created directory /backup/linux/129.40.178.121/
Successfully cloned /sles10master to LINUX01
You should be able to ping 129.40.178.121 within one minute

In the final section, the target user ID is logged on via XAUTOLOG. Because the PROFILE EXEC detects that the ID is logged on in a disconnected mode, Linux is IPLed from minidisk 100. A directory with the user ID and IP address in the name is created under /backup/linux/. The new clone should be on the network in about 30-45 seconds.

**Note:** If the clone.sh script fails you can also add the -v flag for some more diagnostics. Also, check that:
- The target user ID has been granted access to the VSWITCH
- The parameter file is copied and set correctly on LNXMAINT 192

A block diagram of this process is as follows.

![Cloning block diagram](image)

The left side of the figure shows the controller which is on the SLES10 200 disk. It has the master image mounted over the directory /sles10master/, which is the SLES10 100 disk. Note
that the VDISK-based swap spaces, SLES10 101 and 102, are created in memory, so they do not need to be copied.

The script /sbin/clone.sh is invoked and it uses either CP FLASHCOPY or the Linux dd command to copy the 100 minidisk to the target z/VM user ID. The script then mounts the newly copied 100 disk and modifies the networking information to use those values found in the parameter file on the LNXMAINT 192 disk. The script then invokes the CP XAUTOLOG command to log that user ID on. Because the user ID is logged on disconnected, the common PROFILE EXEC from the LNXMAINT 192 disk IPLs from virtual device address 100 and the newly cloned Linux system is brought to life.

9.4 Cloning six more virtual servers

So far you have installed Linux manually twice on SLES10 to create a master image and a controller. You have created a new user ID LINUX01 and cloned it. Now it is time to clone six more times to have one system for each of the virtual servers described in the remaining chapters.

The following steps are involved:

- “Formatting and label six new DASD” on page 148
- “Defining six more user IDs” on page 149
- “Testing logging on to a new user ID” on page 152
- “Creating six new parameter files” on page 150
- “Granting user IDs access to VSWITCH” on page 151

9.4.1 Formatting and label six new DASD

Decide which DASD will be used for the six new user IDs by referring to 2.7.2, “z/VM DASD worksheet” on page 18.

Logon to MAINT. In this example the devices are A712-A717. Query the devices that you want to assign as PERM space:

```shell
g> q <a712-a717>
DASD A712 ZAA712 , DASD A713 ZAA713 , DASD A714 ZAA714 , DASD A715 ZAA715
DASD A716 ZAA716 , DASD A717 ZAA717
```

Detach them from SYSTEM if they are attached:

```shell
g> det <a712-a717> system
DASD A712 DETACHED SYSTEM
DASD A713 DETACHED SYSTEM
...
```

Attach the six DASD to MAINT. When attaching volumes to your own user ID, the * parameter can be used.

```shell
g> att <a712-a717> *
A712-A717 ATTACHED TO MAINT
```

Now format the DASD for PERM or minidisk space with the CPFORMAT command that is associated with this book.

```shell
g> cpformat <a712-a717> as perm
```

Label the following DASD:

<table>
<thead>
<tr>
<th>TargetID</th>
<th>Tdev</th>
<th>OwnerID</th>
<th>Odev</th>
<th>Dtype</th>
<th>Vol-ID</th>
<th>Rdev</th>
<th>StartLoc</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAINT</td>
<td>A712</td>
<td>MAINT</td>
<td>A712</td>
<td>3390</td>
<td>ZAA712</td>
<td>A712</td>
<td>0</td>
<td>3339</td>
</tr>
</tbody>
</table>
### 9.4.2 Defining six more user IDs

Define six more user IDs for Linux virtual servers in the USER DIRECT file named LINUX02 - LINUX07. You will need to use the DASD volumes you just formatted: one for each virtual server. You can repeat the definition of LINUX01 six times with the block copy ""6 prefix command. For example:

```plaintext
==> x user direct
====> /user linux01
...
""6 *
01846 USER LINUX01 LNX4VM 256M 1G G
01847 INCLUDE LNXDFLT
01848 OPTION APPLMON
""6 MDISK 100 3390 0001 3338 MMA711 MR LNX4VM LNX4VM LNX4VM
```
This will create six more copies of the LINUX01. Modify them to have a user ID of LINUX02 - LINUX07, and give each new ID the proper 3390-3 identified by label (MMA712-MMA717 in this example):

```plaintext
USER LINUX02 LNX4VM 256M 1G G
   INCLUDE LNXDFLT
   OPTION APPLMON
   MDISK 100 3390 0001 3338 MMA712 MR LNX4VM LNX4VM LNX4VM
   *
USER LINUX03 LNX4VM 256M 1G G
   INCLUDE LNXDFLT
   OPTION APPLMON
   MDISK 100 3390 0001 3338 MMA713 MR LNX4VM LNX4VM LNX4VM
   *
...
USER LINUX07 LNX4VM 256M 1G G
   INCLUDE LNXDFLT
   OPTION APPLMON
   MDISK 100 3390 0001 3338 MMA717 MR LNX4VM LNX4VM LNX4VM
```

Go to the top of the file and find the definition for the user $ALLOC$. Add dummy definitions for cylinder 0 of each of the new volumes and save the changes:

```plaintext
====> top
====> /alloc
USER $ALLOC$ NOLOG
   MDISK A01 3390 000 001 520RES R
...
   MDISK A0B 3390 000 001 MMA711 R
   MDISK A0C 3390 000 001 MMA712 R
   MDISK A0D 3390 000 001 MMA713 R
   MDISK A0E 3390 000 001 MMA714 R
   MDISK A0F 3390 000 001 MMA715 R
   MDISK A10 3390 000 001 MMA716 R
   MDISK A11 3390 000 001 MMA717 R
...
====> file
```

Check for overlaps and the single gap. Quit out of the USER DISKMAP file:

```plaintext
=> diskmap user
=> x user diskmap
====> all /gap//overlap/
------------------------  4 line(s) not displayed ------------------------
          0      500      501      GAP
------------------------  368 line(s) not displayed ------------------------
====> quit
```

Bring the changes online with the DIRECTXA USER command:

```plaintext
=> directxa user
```

You have now created six new user IDs that can be cloned to.

### 9.4.3 Creating six new parameter files

A new parameter must be created for each of the user IDs with the proper networking information. Link and access the LNXMAINT 192 disk read-write and create six new parameter files:
==>
\[\text{link lnxmaint 192 1192 mr} \]
==>
\[\text{acc 1192 f} \]
==>
\[\text{copy linux01 parm-s10 f linux02 = =} \]
==>
\[\text{copy linux01 parm-s10 f linux03 = =} \]
==>
\[\text{copy linux01 parm-s10 f linux04 = =} \]
==>
\[\text{copy linux01 parm-s10 f linux05 = =} \]
==>
\[\text{copy linux01 parm-s10 f linux06 = =} \]
==>
\[\text{copy linux01 parm-s10 f linux07 = =} \]

Edit each of the six files replacing the appropriate network values. For example, in the LINUX02 PARM-S10, only the TCP/IP address and DNS name need to be modified as all other network and other values are the same:

==>
\[\text{x linux02 parm-s10 f} \]
\[\text{ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb} \]
\[\text{HostIP=129.40.178.122 Hostname=lat122.pbm.ihost.com} \]
\[\text{Gateway=129.40.178.254 Netmask=255.255.255.0} \]
\[\text{Broadcast=129.40.178.255 Layer2=0} \]
\[\text{ReadChannel=0.0.0600 WriteChannel=0.0.0601 DataChannel=0.0.0602} \]
\[\text{Nameserver=129.40.106.1 Portname=dontcare} \]
\[\text{Install=nfs://129.40.178.130/nfs/sles10/SLES-10-CD-s390x-GMC-CD1.iso} \]
\[\text{UseVNC=1 VNCPassword=lnx4vm} \]
\[\text{InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0} \]

When you are finished, release and detach the LNXMAINT 192 disk.

==>
\[\text{rel f (det DASD 1192 DETACHED)} \]

9.4.4 Granting user IDs access to VSWITCH

Modify the PROFILE EXEC on AUTOLOG1 191 to grant access to the VSWITCH for the six new user IDs and add XAUTOLOG commands so they will booted when the z/VM system IPLs.

Link and access the AUTOLOG1 191 disk so the file can be modified from MAINT:

==>
\[\text{link autolog1 191 1191 mr} \]
==>
\[\text{acc 1191 f} \]

Edit the PROFILE EXEC:

==>
\[\text{x profile exec f} \]
...  
/* Grant access to VSWITCH for each Linux user */  
'CP SET VSWITCH VSW1 GRANT SLES10'  
'CP SET VSWITCH VSW1 GRANT LINUX01'  
'CP SET VSWITCH VSW1 GRANT LINUX02'  
'CP SET VSWITCH VSW1 GRANT LINUX03'  
'CP SET VSWITCH VSW1 GRANT LINUX04'  
'CP SET VSWITCH VSW1 GRANT LINUX05'  
'CP SET VSWITCH VSW1 GRANT LINUX06'  
'CP SET VSWITCH VSW1 GRANT LINUX07'  
/* XAUTOLOG each Linux user that should be started */  
'CP XAUTOLOG SLES10'  
'CP XAUTOLOG LINUX01'  
'CP XAUTOLOG LINUX02'  
'CP XAUTOLOG LINUX03'  
'CP XAUTOLOG LINUX04'
It is easiest to grant access to the new user IDs for the current z/VM session with the **SET VSWITCH** command:

```
=> set vswitch vsw1 grant linux02
Command complete
=> set vswitch vsw1 grant linux03
Command complete
=> set vswitch vsw1 grant linux04
Command complete
=> set vswitch vsw1 grant linux05
Command complete
=> set vswitch vsw1 grant linux06
Command complete
=> set vswitch vsw1 grant linux07
Command complete
```

Verify that the user IDs have access with the **QUERY VSWITCH ACCESSLIST** command:

```
=> query vswitch vsw1 acc
VSWITCH SYSTEM VSW1     Type: VSWITCH Connected: 4 Maxconn: INFINITE
     PERSISTENT RESTRICTED NONROUTER Accounting: OFF
     VLAN Unaware
     State: Ready
     IPTimeout: 5     QueueStorage: 8
     Portname: UNASSIGNED RDEV: 3004 Controller: DTCVSW1 VDEV: 3004
     Portname: UNASSIGNED RDEV: 3008 Controller: DTCVSW2 VDEV: 3008 BACKUP
Authorized userids:
     SLES10  LINUX01  LINUX02  LINUX03  LINUX04  LINUX05
     LINUX06  LINUX07
     SYSTEM
```

### 9.4.5 Testing logging on to a new user ID

You should now be able to logon to a new user ID and verify the integrity of the definitions. **Logon to LINUX02** and you should first notice that a NIC is created as well as two VDISKS:

```
LOGON LINUX02
00: NIC 0600 is created; devices 0600-0602 defined
00: z/VM Version 5 Release 2.0, Service Level 0601 (64-bit),
00: built on IBM Virtualization Technology
00: There is no logmsg data
00: FILES: 0003 RDR,   NO PRT,   NO PUN
00: LOGON AT 14:55:43 EST SUNDAY 01/07/07
z/VM V5.2.0    2006-07-18 08:48

DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 101 (64989 4K pages of swap space)
DIAG swap disk defined at virtual address 102 (129981 4K pages of swap space)
Do you want to IPL Linux from DASD 100? y/n
```

If you forgot to grant access to the VSWITCH you will see an error message. Verify that you have OSA devices at addresses 600-602, and read-write DASD devices at addresses 100-102:
Logoff of LINUX02.

Congratulations, you have cloned one Linux virtual server and defined six more user IDs that should now be ready for cloning to. You will clone to some of these user IDs in the chapters that follow.

### 9.5 Reviewing system status

You can step back now and view your system from a DASD point of view as shown in Figure 9-2. If you have followed all sections in this book you should have used 24 3390-3 volumes: 10 for your z/VM system, 7 for the Linux controller and master image and one for each of the seven virtual servers.

![Figure 9-2 Linux virtual server system - DASD view and role view](image)
Cloning open source virtual servers

The secret to creativity is knowing how to hide your sources.
--Albert Einstein

This chapter describes how to clone and customize the following Linux virtual servers:

- “Creating a virtual Web server” on page 155
- “Creating a virtual LDAP server” on page 160
- “Creating a virtual file and print server” on page 167
- “Creating a virtual application development server” on page 171

10.1 Creating a virtual Web server

The example in this section uses the LINUX01 user ID to create a virtual Web server. You should have a vanilla virtual server cloned to the user ID LINUX01 as described in Chapter 9, “Configuring SLES 10 for cloning” on page 139.

10.1.1 Installing Apache RPMs

SSH into the IP address of the new LINUX01 server. Install the following Apache RPMs via the yast -i command:

```bash
# yast -i apache2-prefork apache2 apache2-doc apache2-example-pages
```

You will see yast menus go by as the Apache RPMs are installed. When it is complete you can confirm the RPMs have been added via the rpm -qa command:

```bash
# rpm -qa | grep apache
apache2-example-pages-2.2.0-21.2
apache2-prefork-2.2.0-21.2
apache2-doc-2.2.0-21.2
apache2-2.2.0-21.2
```
10.1.2 Testing Apache

Start the Apache Web server to verify it is installed successfully. You must start Apache as root, but it will then launch child processes as a less privileged user to listen for and handle requests.

```
# rcapache2 start
Starting httpd2 (prefork) done
```

**Note:** You may get warnings similar to the following:

```
Starting httpd2 (prefork) httpd2-prefork: apr_sockaddr_info_get() failed for lat121
httpd2-prefork: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1 for ServerName done
```

This may mean your DNS resolution is not set correctly. These warning will not prevent you from continuing this section.

To verify that Apache is installed correctly, after it's been started, point a Web browser to the server and see the Apache test page. In your Web browser, put in the host name or IP address of your Web server as the URL. For example, the virtual server running on LINUX01 has a DNS name of lat121.pbm.ihost.com:

```
http://lat121.pbm.ihost.com
```

You should see a test page with words **It Works!**

If you get an error in starting Apache, look in the log file `/var/log/apache2/error-log` for clues. If Apache started successfully but you can't reach the test page from a browser, try accessing it using the IP address rather than the DNS name.

10.1.3 Configuring SSL for Apache

Secure Sockets Layer (SSL) is used to encrypt data between the client (browser) and the server. In order for the client to know you are a legitimate Web server, you will first need to create a server certificate. Then, with your certificate in hand, you can setup Apache to provide secure communications with SSL.

**Creating a server certificate**

It is recommended that you first create a self-signed certificate to test that your SSL configuration is correct. Then for production purposes you may wish to purchase a certificate signed by a trusted Certificate Authority (CA).

Use OpenSSL to create the certificate. This process includes:

1. Creating a public/private key pair
2. Creating a certificate request.
3. Creating a server certificate
4. Use the `openssl genrsa` command to generate a RSA key pair. The `-rand` switch is used to provide OpenSSL with randomized data to ensure that generated keys are unique and unpredictable. Substitute `<file1:file2:file3>` with paths to large, random files on your system.
5. Create a certificate request. This is needed to create a self-signed certificate or to obtain a CA-signed certificate. The creation process will ask you questions about your business. Answer them as appropriate:

```
# openssl req -new -key <lat121.key> -out <lat121.csr>
```

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

-----

Country Name (2 letter code) [AU]:<US>
State or Province Name (full name) [Some-State]:<NY>
Locality Name (eg, city) []:<Poughkeepsie>
Organization Name (eg, company) [Internet Widgits Pty Ltd]:<IHOST>
Organizational Unit Name (eg, section) []:<PBM>
Common Name (eg, YOUR name) []:<Admin>
Email Address []:<admin@pbm.ihost.com>

Please enter the following 'extra' attributes to be sent with your certificate request
A challenge password []:<a3tfgm>
An optional company name []:<IHOST>

6. To obtain a server certificate, you can either create a self-signed certificate, or obtain a trusted CA-signed certificate. Trusted Certificate Authorities include GeoTrust, VeriSign and Thawte. Submit your certificate signing request (the .csr file you just created) to one of them for processing (See 10.1.5, “Apache resources” on page 159 for their URLs). To create your own self-signed certificate, run the following command:

```
# openssl x509 -req -days 30 -in <lat121.csr> -signkey <lat121.key> -out <lat121.crt>
```

```
Signature ok
subject=/C=US/ST=NY/L=Poughkeepsie/O=IHOST/OU=PBM/CN=Admin/emailAddress=admin@ihost.pbm.com
Getting Private key
```

7. Move the SSL files into the appropriate directories under Apache:

```
# mv <lat121.key> /etc/apache2/ssl.key
# mv <lat121.crt> /etc/apache2/ssl.crt
```

### Setting up virtual hosts

Because SSL-protected Web pages run on a different port than the non-protected Web pages, you should consider them separate Web servers. A common way of serving an SSL-enabled Web site is to create a virtual host on the Web server. Apache's Virtual Host capability allows you to have multiple Web servers on one machine. The main configuration file `/etc/apache2/httpd.conf` reads virtual hosts information from included configuration files in the directory `/etc/apache2/vhosts.d/`. To create a virtual host configuration file, copy the template file `vhost-ssl.template` to your own configuration file and customize it with the paths to SSL logs and your SSL certificate and key. The lines that need to be modified are in bold:

```
# cd /etc/apache2/vhosts.d
```
# cp vhost-ssl.template ssl.conf
# vi ssl.conf
...
<VirtualHost _default_:443>
  # General setup for the virtual host
  DocumentRoot "/srv/www/htdocs"
  #ServerName www.example.com:443
  #ServerAdmin webmaster@example.com
  ErrorLog /var/log/apache2/ssl_error_log
  TransferLog /var/log/apache2/ssl_access_log
  ...
  # Server Certificate:
  # Point SSLCertificateFile at a PEM encoded certificate. If
  # the certificate is encrypted, then you will be prompted for a
  # pass phrase. Note that a kill -HUP will prompt again. Keep
  # in mind that if you have both an RSA and a DSA certificate you
  # can configure both in parallel (to also allow the use of DSA
  # ciphers, etc.)
  SSLCertificateFile /etc/apache2/ssl.crt/<lat121.crt>
  #SSLCertificateFile /etc/apache2/ssl.crt/server-dsa.crt
  # Server Private Key:
  # If the key is not combined with the certificate, use this
  # directive to point at the key file. Keep in mind that if
  # you've both a RSA and a DSA private key you can configure
  # both in parallel (to also allow the use of DSA
  # ciphers, etc.)
  SSLCertificateKeyFile /etc/apache2/ssl.key/<lat121.key>
  #SSLCertificateKeyFile /etc/apache2/ssl.key/server-dsa.key
  ...

Edit the configuration file for the Apache startup script, /etc/sysconfig/apache2, to add a
start time flag to let Apache know to enable SSL:

# vi /etc/sysconfig/apache2
...
APACHE_SERVER_FLAGS="SSL"

If Apache is already running, you need to restart it to take the changes:

# rcapache2 restart
Syntax OK
Shutting down httpd2 (waiting for all children to terminate) done
Starting httpd2 (prefork) done

Test the SSL-enables Web server by pointing a browser to:

https://<lat131.pbm.ihost.com>

If you are using a self-signed certificate, then you will see a warning before your browser
downloads the page. It is simply telling you that it is not signed by a trusted CA, click Yes to
proceed:
You should again see a **It Works!** words as you have seen before, the only difference being that this one has an **https** prefix, not **http**.

You can customize your Web site with SSL in many ways, such as choosing which pages are SSL enabled and which SSL Ciphers to use, etc. Refer to the Apache documentation in 10.1.5, “Apache resources” on page 159 for more details.

### 10.1.4 Populating your Web site

You can begin to put your Web pages in the directory `/srv/www/htdocs/`, which is the default Web root. For security and customization purposes, you might want to change the default Web root to point to another directory. The easiest way to do this is to copy `/etc/apache2/default-server.conf` to your own configuration file, i.e. `/etc/apache2/my-server.conf`.

Make the changes in `/etc/apache2/my-server.conf`, and then edit `/etc/apache2/httpd.conf` to use `my-server.conf`.

### 10.1.5 Apache resources

The following Web sites contain additional information on Apache:

- [http://www.securityfocus.com/infocus/1786](http://www.securityfocus.com/infocus/1786)
10.2 Creating a virtual LDAP server

The Lightweight Directory Access Protocol (LDAP) is commonly implemented via the OpenLDAP package which comes standard with most Linux distributions. Among other directory functions, OpenLDAP allows for centralized login authentication and user and group ID resolution.

In this section you will install Linux manually and set up login authentication to the new virtual LDAP server. Then you will go back to the virtual Web server you just created and point it to the new LDAP server.

Then you may want to configure the master image so that it is pointing to this virtual server. If you do so, all Linux images that are cloned will be able to use this virtual LDAP server.

The steps in this section are as follow:

- “Manually installing a Linux virtual server” on page 160
- “Configuring the virtual LDAP server” on page 163
- “Investigating the new virtual server” on page 164
- “Adding a new user” on page 164
- “Setting another virtual server to use LDAP server” on page 165

10.2.1 Manually installing a Linux virtual server

For OpenLDAP to work properly, you must have a working DNS server and the DNS name and IP address values set in the LINUX02 PARMFILE must be associated.

It is recommended that you manually install Linux to create a virtual LDAP server rather than cloning a virtual server. The reason for this is because of the LDAP and certificate configuration module which is invoked in the second half of a manual install. When installing both the master image and controller, it was recommended that you skip the Service Configuration window. When installing the virtual LDAP server, this YaST configuration module will be used - it makes setting up OpenLDAP much easier.

Open a 3270 session to LINUX02 to install Linux manually. You should be prompted to IPL Linux - answer n to the question:

```
LOGON LINUX02
...
Do you want to IPL Linux from DASD 100? y/n
n
```

If you use the QUERY VIRTUAL STORAGE command it will show that you have a 256MB machine:

```
==> q v stor
00: STORAGE = 256M
```

This is adequate memory to run a SLES10 Linux image, however it is not enough to install Linux with. Temporarily modify the storage up to 512MB with the DEFINE STORAGE command. Then IPL CMS and again answer n to the question of IPLing Linux:

```
==> def stor 512m
00: STORAGE = 512M
00: Storage cleared - system reset.
==> ipl cms
z/VM V5.2.0 2006-07-18 08:48
```

DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 101 (64989 4K pages of swap space)
DIAG swap disk defined at virtual address 102 (129981 4K pages of swap space)
Do you want to IPL Linux from DASD 100? y/n

n

Verify that you have a 512MB virtual machine:

```bash
===> q v stor
00: STORAGE = 512M
```

This change is for the duration of the user ID session. When you logoff and log back on this user ID, the storage will go back to 256MB.

Invoke the SLES10 EXEC to begin a manual install of a system

```bash
===> sles10
```

... Complete the first half of the install as described in 7.4.1, “Begin the SLES 10 installation” on page 96. If you have cloned to LINUX02 you can safely skip the formatting of disk 100 after you activate it. The first half of the install will be the same.

From a 3270 session you will IPL 100 to complete the second half of the installation which is described in 7.4.5, “Completing YaST2 installation” on page 102. It is recommended that you set the root password, turn off the firewall and skip the Internet test as before. When you get to the Installation Settings panel, there will be a difference:

1. Accept the default of the Use Following Configuration radio button and click OpenLDAP server as shown in Figure 10-2.

   **CA Management**
   
   Creating default CA and certificate.
   
   with higher security requirements, you should change the password.

   * CA Name: YaST Default CA
   * Common Name: YaST Default CA (lati22)
   * Server Name: lati22.pbn.linux.com
   * Country: US
   * Password: [root password]
   * Email: postmaster@pbn.linux.com

   **OpenLDAP Server**
   
   Start LDAP Server: NO

   ![Image](Image)

   **Figure 10-2 YaST module to configure OpenLDAP and certificates**

2. You will see a warning message starting with Changing anything in this dialog ... Click OK.

3. This should bring you to the Configure LDAP Server panel as shown in Figure 10-3 on page 162. Click the Enable Server radio button. Uncheck the Register at an SLP Daemon check box at the bottom and click Next. The LDAP Administrator password will be set to the root password you typed earlier. You may wish to change this.
4. This will bring you back to the Installation Settings panel. Click Next.
5. You should see a certificate being created.
6. This will take you to a message box stating These packages need to be installed: openldap2. Click Continue and the appropriate RPMs will be installed.
7. In the User Authentication Method panel accept the default of LDAP and click Next.
8. In the LDAP Client Configuration panel, accept the defaults and click Next.
9. You will again see a message that the pam_ldap, nss_ldap, pam_ldap-32bit and nss_ldap-32bit packages must be installed. Click Continue and the appropriate RPMs will be installed.
10. You will get message This change only affects..., click OK.
11. In the New LDAP user panel, add one new LDAP user. An example is shown in Figure 10-4 on page 163 of adding the user mikem. Click Next when you are ready.
12. In the *Writing the system configuration* window the *SuSEconfig* tool writes all your settings to disk.

13. Complete the installation as you did earlier.

14. Go back to your 3270 session and clear the screen. Your VNC session will end and your new system will complete booting.

You should now have a vanilla SLES10 system with an LDAP server installed and initially configured.

### 10.2.2 Configuring the virtual LDAP server

Because you did not clone this server via the *clone.sh* script, there are some configuration steps that need to be completed. Go back to section 7.5, “Configuring the master image” on page 104 and repeat the steps that you performed on the master image.

The *clone.sh* script also performs two other tasks to enable key-based authentication from the controller and to create a directory for backup and the DAZL application. Perform the following steps:

1. **Start an SSH session to the controller.**

2. Use the `ssh` command to first create the directory `/root/.ssh/` then copy the controller’s public key to `authorized_keys` file (to allow for key-based authentication) via `scp` from the controller to the LDAP server running on `LINUX02` (129.40.178.122 in this example):

   ```
   # cd /etc/ssh
   # ssh <129.40.178.122> 'mkdir /root/.ssh; exit'
   # scp ssh_host_dsa_key.pub <129.40.178.122>:/root/.ssh/authorized_keys
   [Password:
   ssh_host_dsa_key.pub                          100%  601     0.6KB/s   00:00]
   
   You should now be able to get an SSH session with no password needed:
   # ssh 129.40.178.122
   ```
3. Create backup directory on controller for LINUX02 with the IP address in the name. In this example the IP address is 129.40.178.122:

```bash
# cd /backup/linux
# mkdir LINUX02-on-129.40.178.122
# chmod g+w LINUX02-on-129.40.178.122
```

4. Create a description and log file with one line describing the server

```bash
# cd LINUX02-on-129.40.178.122
# echo "This is an LDAP server" > descLog.txt
```

You have now configured the server as though it were cloned.

### 10.2.3 Investigating the new virtual server

Start an SSH session as root to your new virtual server. Verify that LDAP is running:

```bash
# rcldap status
Checking for service ldap: running
```

The user that was added (mikem in this example) should exist. Look for it the entry in the /etc/passwd file:

```bash
# id mikem
uid=1000(mikem) gid=100(users) groups=100(users)
# grep mikem /etc/passwd
```

The `grep` command gives no output. Why is this? Because the user was not added to the local file system. Rather, it was added to the OpenLDAP database. Confirm this via the `ldapsearch` command searching for the entry with `uid=mikem` in this example:

```bash
# ldapsearch -x uid=mikem
... mikem, people, pbm.ihost.com
dn: uid=mikem,ou=people,dc=pbm,dc=ihost,dc=com
...
```

This shows that the user named mikem has been added to the LDAP server.

### 10.2.4 Adding a new user

The first LDAP user was added during Linux installation. There are a number of different ways to adding LDAP users. It can be done with the `ldapadd` command and a manually edited LDIF file, or it can be done with a graphical LDAP browser such as `gq`. But perhaps `yast` in curses mode (line mode graphics) is a good compromise between these two.

1. Invoke `yast`:

```bash
# yast
```
2. On the main panel, choose Security and Users then User Management on the right side.

3. You will be prompted for the LDAP (root) password. Enter the password, Tab to OK and press Enter.

4. You will see the User and Group Administration panel. Tab to Add and press Enter.

5. You will see the New LDAP User panel. Enter the information for the new user, tab to Accept and press Enter.

6. In the User and Group Administration panel tab to Finish and press Enter. The user will be added to the LDAP server.

7. In the YaST Control Center, tab to Quit and press Enter.

Now you should be able to SSH into the LDAP virtual server with the new user's credentials.

10.2.5 Setting another virtual server to use LDAP server

Now that you have a virtual LDAP server, you may want to point the other virtual servers to it so you will have a centralized user database. If you have been following along in this book you have created a Web server running on the LINUX01 user ID. To point it to an LDAP server is fairly easy: you will need to install some RPMs and do some configuration. In this section you will perform the following steps:

- “Testing that the LDAP client is not working” on page 165
- “Adding four LDAP RPMs” on page 166
- “Using YaST to modify the LDAP authentication client” on page 166
- “Testing the LDAP client” on page 167
- “Testing the LDAP client” on page 167

Testing that the LDAP client is not working

Before you start, try a couple of commands to show that LDAP is not working. Get an SSH session to the virtual Web server running on the user ID LINUX01. Use the LDAP user ID that you added earlier to the virtual LDAP server. In this example it is mikem.

```
# ldapsearch -x uid=<mikem>
ldap_bind: Can't contact LDAP server (-1)
# id <mikem>
id: mikem: No such user
```

The ldapsearch command cannot resolve the LDAP user because it cannot contact the LDAP server. Similarly, the id command gives no output for the same reason.
Adding four LDAP RPMs

Use the `yast -i` command to add the RPMs `pam_ldap` and `nss_ldap`:

```sh
# yast -i pam_ldap nss_ldap pam_ldap-32bit nss_ldap-32bit
```

You should see the packages being added in YaST curses screens. When the process is complete verify that the two packages were added with the following `rpm` command:

```sh
# rpm -qa | grep _ldap
pam_ldap-180-13.2
nss_ldap-246-14.2
pam_ldap-32bit-180-13.2
nss_ldap-246-14.2
```

**Note:** If you don’t install these packages manually they will be installed automatically in the next section.

Using YaST to modify the LDAP authentication client

The `yast` system administration interface can be used to configure the LDAP authentication client. Select **Network Services** on the left side of the main screen, then LDAP Client:

```sh
# yast
```

On the panel that follows, use the Tab key to move to Use LDAP and press the **space bar** to select that choice, then enter IP address of your LDAP server. You can either enter LDAP base DN manually or press **Fetch DN** and then **OK** in result window. This way you can make sure LDAP server is accessible.:
That is all. Use the Tab key to move to Finish and press Enter. You will get warning window, press OK there. Press the Quit button on the main window to quit YaST.

Testing the LDAP client
Save the file. Now try the id and ldapsearch commands again. This time they should both succeed:

```
# id mikem
uid=1000(mikem) gid=100(users)
groups=100(users),14(uucp),16(dialout),17(audio),33(video)
# ldapsearch -x uid=mikem
# extended LDIF
#
# LDAPv3
# base <> with scope sub
# filter: uid=mikem
# requesting: ALL
#
#
# mikem, people, pbm.ihost.com
dn: uid=mikem,ou=people,dc=pbm,dc=ihost,dc=com
...
```

You should also be able to start an SSH session to the virtual Web server using the LDAP user.

10.3 Creating a virtual file and print server

Samba allows Windows clients to map Linux file systems as shared drives. Samba can also act as a middle-man between Windows clients and a Linux print server. The recommended Linux print server is CUPS - the Common UNIX Printing System. This section does not describe the configuration of CUPS but it does describe how the necessary RPMs are installed.

The steps in this section are as follow:

- “Cloning a Linux virtual server” on page 168
- “Installing necessary RPMs” on page 168
10.3.1 Cloning a Linux virtual server

From the controller, clone a basic virtual server. In this example the user ID LINUX03 is used.

```bash
# clone.sh linux03
...
```

SSH in to the new virtual server.

10.3.2 Installing necessary RPMs

Add the following RPMs via the `yast -i` command:

```bash
# yast -i samba yast2-samba-server samba-doc samba-pdb samba-vscan samba-winbind cups \
cups-drivers ghostscript-serv
```

You will see a number of YaST curses screens flash by as the RPMs are added to the system.

Confirm that the RPMs were added:

```bash
# rpm -qa | egrep "samba|cups"
cups-client-1.1.23-40.6
samba-doc-3.0.22-20.30
cups-1.1.23-40.6/nsamba-vscan-0.3.6b-42.16
samba-3.0.22-13.16
cups-drivers-1.1.23-28.2
samba-pdb-3.0.22-13.16
cups-libs-1.1.23-40.6
yast2-samba-client-2.13.22-0.2
samba-client-3.0.22-13.16
yast2-samba-server-2.13.11-1.3
samba-winbind-3.0.22-13.16
```

When completed you should still have about 1.3GB free (your values may differ):

```bash
# df -h
Filesystem Size Used Avail Use% Mounted on
/dev/dasda1 2.3G 1.3G 933M 58% /
udev 122M 76K 122M 1% /dev
```

10.3.3 Configuring Samba configuration file

The one configuration file for Samba is `/etc/samba/smb.conf`. It is easy to add an SMB share that will be made available by the Samba server. A good test directory is `/usr/share/doc/` as it has much good Linux documentation. The following example will create a file `share` named `sharedoc`:

```bash
# cd /etc/samba
# cp smb.conf smb.conf.orig
# vi smb.conf // add three lines at the bottom of the file:
...
[sharedoc]
    comment = SLES10 on System z documentation
```
path = /usr/share/doc/

This will cause an SMB share named sharedoc consisting of the contents of /usr/share/doc to be created when Samba is started.

### 10.3.4 Adding a Samba user

The default method that Samba uses to determine users’ credentials is to look in the /etc/samba/smbpasswd file. That user must first exist in the Linux file system (/etc/passwd, /etc/shadow, etc). To create a new Samba user, the `smbpasswd -a` command is used. The following example shows adding the user mikem first to Linux then to the `smbpasswd` file.

**Note:** If you added your user before either as a local user or as the LDAP one, you can skip adding user and follow with `smbpasswd` command.

```bash
# id mikem
id: mikem: No such user
# useradd mikem
# passwd mikem
Changing password for mikem.
New password:
Re-enter new password:
Password changed
# mkdir /home/mikem
# chown mikem.users /home/mikem
# smbpasswd -a mikem
New SMB password:
Retype new SMB password:
...
Added user mikem.
```

You can see that the last `smbpasswd` command added mikem to the file `smbpasswd`:

```bash
# cat smbpasswd
...
mikem:1000:2E6F0C45D305054CAAD3B435B51404EE:74154D10CC18CFC98F1ED128DB8764DA:[U ]:LCT-456AFE0E:
```

This method of maintaining Samba users, groups and passwords is good for a small number of users. For a larger number of users, merging Samba and LDAP is recommended. It is not a simple as pointing the virtual file and print server at the virtual LDAP server as described in 10.2.5, “Setting another virtual server to use LDAP server” on page 165, because the Samba schema must first be added to LDAP. Details are outside the scope of this book, but there are related presentations, *Directory Serving Solutions Using OpenLDAP* and *File Serving Solutions Using Samba* on the Web at:

http://linuxvm.org/present/

### 10.3.5 Starting Samba at boot time

Samba consists of two daemons `nmbd` and `smbd`. They can be started for the current session with the `rcnmbd` and `rcsmbd` commands:

```bash
# rcnmb start
Starting Samba NMB daemon
done
```
The following `chkconfig` commands will set these daemons to start at boot time:

```
# chkconfig nmb on
# chkconfig smb on
```

Samba should now be running and configured to start at boot time.

### 10.3.6 Testing your changes

You can verify that the Samba daemons are running via the `status` parameter to the `rcnmb` and `rcsmb` commands:

```
# rcnmb status
Checking for Samba NMB daemon
running
# rcsmb status
Checking for Samba SMB daemon
running
```

You can test getting a Samba share from a Windows desktop. Go to any Windows Explorer window (such as `My Computer`) and select **Tools -> Map Network Drive**. Use the Universal Naming Convention (UNC) to specify the Samba server and share name as shown in the upper left corner of Figure 10-5 on page 170. In this example the UNC is `\129.40.178.123\sharedoc`. Then click **Finish**. If all the steps were correct, you should see the files in a new Explorer window as shown in the bottom right corner of the figure.

![Map Network Drive](image)

*Figure 10-5  Mapping a network drive to the Samba server*

You should now have Samba configured and running with one new share available.
If you prefer command line, you can achieve the same with these commands:

```bash
c:\> net use y: \129.40.178.123\sharedoc
The password is invalid for \129.40.178.123\sharedoc.

Enter the user name for '129.40.178.123': mikem
Enter the password for 129.40.178.123:
The command completed successfully.

c:\>net use y: /delete
y: was deleted successfully.
```

### 10.3.7 Configuring printing

Configuring printing is more complex and is beyond the scope of this section. For many more details see the Redpaper *Printing with Linux on zSeries Using CUPS and Samba*, REDP-3864:


### 10.4 Creating a virtual application development server

Most Linux distributions come with a basic set of application development tools, making Linux one of the most versatile development systems. These basic tools are ideal for projects of any size.

There are three main areas of development in Linux:

- Linux kernel development (C) for the Linux operating system itself, such as subsystems, device drivers, memory management.
- Application development (C/C++ and Java) for software to be used on Linux.
- Web development for applications to be run on the Web, such as stock trade applications or E-mail applications.

The development languages used in implementation range from scripting languages such as Python or Tcl, to compiled languages such as C/C++ and Java. There are software available on Linux to help form a development system for developers to create integrated applications. MySQL and Apache are among them. A popular open source Web platform is LAMP, which stands for the open source software and programming languages used to make up the platform: Linux, Apache, MySQL, Python or PHP. Other times, it is just as useful to know about Linux development tools when you want to build an application from source code downloaded from www.sourceforge.net.

#### 10.4.1 Cloning a Linux virtual server

From the controller clone a basic virtual server to LINUX04.

```
# clone.sh linux04
...
```

SSH in to the new virtual server.

#### 10.4.2 Scripting Languages

Scripts are good for quickly automating a process or writing your own commands. They are also used for being the backbone of robust applications. There are numerous scripting
languages used in Linux application development, here are overviews of the most popular
and general ones, obtained from their package descriptions.

- **Python**: Python is an interpreted, object-oriented programming language, and is often
  compared to Tcl, Perl, Scheme, or Java. You can find an overview of Python in the
documentation and tutorials included in the python-doc (HTML) or python-doc-pdf (PDF)
packages. To install the python interpreter, execute the command:

  
  # yast -i python

- **Perl**: Practical Extraction and Report Language. Perl is optimized for scanning arbitrary
  text files, extracting information from those text files, and printing reports based on that
  information. It is also good for many system management tasks. Perl is intended to be
  practical (easy to use, efficient, and complete) rather than beautiful (tiny, elegant, and
  minimal). To install perl, execute the command:

  
  # yast -i perl

- **tcl**: the “Tool Command Language”, is a very simple programming language. Tcl provides
  basic language features such as variables, procedures, and control. It runs on almost any
  modern OS, such as Unix, Macintosh, and Windows 95/98/NT computers. The key feature
  of Tcl is its extensibility. Tcl was originally developed as a reusable command language for
  experimental Computer Aided Design (CAD) tools. The interpreter was implemented as a
  C library which could be linked to any application. It is very easy to add new functionality to
  the Tcl interpreter, so it is an ideal, reusable “macro language” that can be integrated into
  many applications. One of Tcl's best loved features is the ease with which one can add
  new commands (known as extensions). New commands can range from something as
  simple as a new format for producing output, to extensions such as Tk which provide
  graphically oriented programming paradigms. Another very popular extension is Expect
  which can be used to automate console-based interactive applications. To install tcl,
  execute the command

  
  # yast -i tcl

- **PHP**: PHP (recursive acronym for “PHP: Hypertext Preprocessor”) is a widely-used Open
  Source general-purpose scripting language that is especially suited for Web development
  and can be embedded into HTML. PHP development is focused on server-side scripting,
  but you can do much more with it. To install PHP, execute the command:

  
  # yast -i php

### 10.4.3 C/C++ development tools

Most Linux distributions come with the C/C++ compiler, gcc. This is also known as the GNU
compiler collection because it can compile other languages such as Fortran but it's most
frequently used to compile C and C++ code. In the minimal SLES10 installation, none of the
development packages are installed. In order to use gcc, you must install it using yast:

  
  # yast -i gcc

```
# rpm -qa | grep gcc
libgcc-4.1.0-28.4
gcc-4.1.0-28.4
```

gcc does preprocessing, compilation, assembly and linking for files with extensions .c, .cpp,
and numerous others (see the gcc manual page). Most C/C++ programs will require
preprocessing, compilation and assembly first to create object files, then linking combines all
the object files into an executable file.

For security reasons, you shouldn’t use root for application development. You should either
get another session as a non-root user or from root su to a non-root user. In this example, the

172   z/VM and Linux on IBM System z: The Virtualization Cookbook
non-root user developer1 is used. The files readfile.c and writefile.c are compiled into the executable files:

```
# su - <developer1>
$ gcc -O -Wall -I/usr/local/include -o readfile.o -c readfile.c
$ gcc -O -Wall -I/usr/local/include -o writefile.o -c writefile.c
$ gcc -o fileoperations readfile.o writefile.o
```

The -O option is to generate optimized code, -Wall is used to display all warnings. The option -I is used to include header files needed by the source and -c is to tell gcc not to run the linker. The last command links the two object files into one executable file. For debugging using gdb, you can generate symbolic information using the -g option:

```
$ gcc -g -O -Wall -I/usr/local/include -o readfile.o readfile.c
$ gcc -g -O -Wall -I/usr/local/include -o writefile.o writefile.c
$ gcc -g -o fileoperations readfile.o writefile.o
```

The GNU debugger, or gdb, is a very popular and robust debugger for C/C++ programs. You can step through your program (that has been successfully compiled) to see where it is failing. Install it using yast:

```
# yast -i gdb
```

There is a good tutorial on getting started with gdb:

http://www.unknownroad.com/rtfm/gdbtut/gdbuse.html

Keep in mind that you can also set breakpoints at functions in the code. Refer to the manual page of gdb for more information: man gdb.

To make a large program more manageable, developers usually create a makefile that specifies instructions on how to compile a program. Then use the GNU make tool to use the makefile to make a working program. For more information on makefiles, see

http://vertigo.hsr1.rutgers.edu/ug/make_help.html

To install make, issue the command:

```
# yast -i make
```

### 10.4.4 Java development tools

SLES10 comes with IBM Java Standard Development Kit (SDK) which is needed if you want to develop Java applications. You need a Java Runtime Environment (JRE) if you only want to run Java applications. Make sure you have the right Java package, if not or you're not sure which one you need, just install the SDK:

```
# yast -i java-1_4_2-ibm-devel
```

Again, you should do application development as a non-root user. Open another SSH session and log in as a non-root user, or from the current session, su to a non-root user. Java programs are compiled using javac. Here's an example:

```
# su - <developer1>
$ javac HelloWorld.java
$ java HelloWorld
Hello World!
```

The resulting file is HelloWorld.class which can be run if there is a main method defined in HelloWorld.java. For Java applets, run with appletviewer. If you are using methods from other classes that aren't in the same package, you can reference them using the -classpath option. For debugging information, use the -g option.
If your executable program has multiple class files, you can save the user time by making an executable .jar file. All you need to do is specify which class has the executable main method. This way the user just need one .jar file, instead of numerous class files.

First create a manifest file that specifies where the main method is by adding a simple one-liner. You must specify the package name as well:

```
$ vi mainClass
Main-Class: myHello.HelloWorld
```

Now use the `jar` command to create the executable jar file that knows where the main method is. All of the needed files are in the directory `myHello`, and are in the package `myHello`. The `jar` command (similar to `tar`) packages the directory contents into one file with the `cmf` flags. After packaging, check the contents with the `tf` flags:

```
$ jar cmf mainClass myHello.jar myHello
$ jar tf myHello.jar
META-INF/
META-INF/MANIFEST.MF
myHello/
myHello/HelloWorld.java
myHello/HelloWorld.class
myHello/PrintScreen.java
myHello/PrintScreen.class
```

Run the .jar file by invoking `java` with the `-jar` option.

```
$ java -jar myHello.jar
Hello World!
```

A good Java debugger is `jdb`, it comes with IBMJava2-SDK and can be run similar to `gdb`. A good tutorial is on the Web at:

```
http://java.sun.com/j2se/1.3/docs/tooldocs/solaris/jdb.html
```

You can use the GNU `make` to build from Java makefiles or the more recent and popular Ant. Ant uses XML technology. Here's a great guide to get you started with either tool:

```
```

Your application development server is now ready to use.

### Additional resources

The following Web sites are resources for additional information on application development topics:

**Scripting languages**

- [http://www.perl.com/](http://www.perl.com/)
- [http://www.python.org/](http://www.python.org/)

**C/C++**

- [http://vertigo.hsrl.rutgers.edu/ug/make_help.html](http://vertigo.hsrl.rutgers.edu/ug/make_help.html)

**Java**

http://java.sun.com/
http://java.sun.com/j2se/1.3/docs/tooldocs/solaris/jdb.html

**Linux kernel development**
http://www.kernel.org/pub/linux/docs/lkml/#blk

**Web development**
http://www.onlamp.com/
http://cgi.resourceindex.com/
http://www.perl.com/

**Help with vi**
http://www.freeos.com/guides/lsst/misc.htm#commonv
Cloning IBM middleware on SLES 10

Any fool can make things bigger, more complex, and more violent. It takes a touch of genius, and a lot of courage, to move in the opposite direction.

--Albert Einstein

Because not all middleware is supported on SLES10 yet, this chapter is left blank. This chapter will be finished and describe how the following IBM middleware products can be cloned:

- WebSphere Application Server
- DB2
- MQ Series
Chapter 12. Miscellaneous recipes

Two things are infinite: the universe and human stupidity; and I'm not sure about the universe.
--Albert Einstein

This chapter has the following sections of miscellaneous tasks that you might want to perform:

- “Adding a logical volume” on page 179
- “Extending an existing logical volume” on page 185
- “Rescuing a Linux system” on page 191

12.1 Adding a logical volume

There are times when you require more disk space than a single direct access storage device (DASD) volume provides. For example, if you want to have a shared /home/ directory you will want it to be of sufficient size. When this is the case, you can use the Logical Volume Manager (LVM) to combine multiple DASD volumes into one logical volume.

The following process describes how to create a logical volume with additional DASD on a Linux guest. The overall steps in adding a logical volume are:

- “Adding minidisks to the z/VM directory entry” on page 179
- “Making the new minidisks available” on page 180
- “Formatting and partitioning the minidisks” on page 180
- “Creating a logical volume and file system” on page 181
- “Updating the file system table” on page 183

12.1.1 Adding minidisks to the z/VM directory entry

The overall steps are the following. Specific details of these steps on z/VM are not given, but they are given for Linux. Use the LINUX02 user ID for this test.

- Determine the labels of the two volumes that will be added.
Add minidisk statements to define minidisks (at virtual addresses 103 and 104 of size 3338 cylinders each in this example) to the appropriate Linux user ID definition in the USER DIRECT file.

Create the USER DISKMAP file to verify the disk layout.

Bring the changes online with the DIRECTXA command.

Shutdown the Linux system.

Logoff the user ID.

Log back on to it and IPL Linux.

### 12.1.2 Making the new minidisks available

When your system comes back up, start an SSH session to it. Use the lsdasd command to verify that the new minidisks are not seen yet (because there is dasd=100-102 in /etc/zipl.conf):

```bash
# lsdasd
0.0.0100(ECKD) at ( 94:  0) is dasda : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0101(FBA) at ( 94:  4) is dasdb : active at blocksize 512, 524288 blocks, 256 MB
0.0.0102(FBA) at ( 94:  8) is dasdc : active at blocksize 512, 1048576 blocks, 512 MB
```

Bring disks 103 and 104 online via the chccwdev command and verify they are available via the lsdasd command:

```bash
# chccwdev -e 103
Setting device 0.0.0103 online
Done
# chccwdev -e 104
Setting device 0.0.0104 online
Done
# lsdasd
0.0.0100(ECKD) at ( 94:  0) is dasda : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0101(FBA) at ( 94:  4) is dasdb : active at blocksize 512, 524288 blocks, 256 MB
0.0.0102(FBA) at ( 94:  8) is dasdc : active at blocksize 512, 1048576 blocks, 512 MB
0.0.0103(ECKD) at ( 94: 12) is dasdd : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0104(ECKD) at ( 94: 16) is dasde : active at blocksize 4096, 600840 blocks, 2347 MB
```

You should be ready to format the new minidisks.

### 12.1.3 Formatting and partitioning the minidisks

You could format the minidisks sequentially, but you can also use the following bash for loop to put two dasdfmt jobs in the background so as to format both minidisks in parallel:

```bash
# for i in d e
>     do
>     > dasdfmt -b 4096 -y -f /dev/dasd$i &
>     done
[1] 2713
[2] 2714
```

When the jobs are finished use the fdasd command with the -a flag to create a single partition from each minidisk:

```bash
# fdasd -a /dev/dasdd
auto-creating one partition for the whole disk...
writing volume label...
writing VTOC...
rereading partition table...
```
# fdasd -a /dev/dasde
auto-creating one partition for the whole disk...
writing volume label...
writing VT0C...
rereading partition table...

The minidisks are now ready for you to use in the logical volume. If you are creating a new logical volume, see 12.1.4, “Creating a logical volume and file system” on page 181. If you are extending an existing logical volume, skip ahead to 12.2, “Extending an existing logical volume” on page 185.

12.1.4 Creating a logical volume and file system

The overall steps involved in creating a logical volume are:

- Create physical volumes from the two DASD
- Create a single volume group
- Create a single logical volume
- Make a file system from the logical volume

Figure 12-1 shows a block diagram of the logical volume manager reflecting this example.

Creating physical volumes from the two DASD

The `pvcreate` command initializes DASD for use by LVM. Initialize the two new DASD partitions. Verify with the `pvdisplay` command:

```bash
# pvcreate /dev/dasdd1 /dev/dasde1
Physical volume "/dev/dasdd1" successfully created
Physical volume "/dev/dasde1" successfully created

# pvdisplay
--- NEW Physical volume ---
```
PV Name               /dev/dasdd1
VG Name
PV Size               2.29 GB
Allocatable           NO
PE Size (KByte)       0
Total PE              0
Free PE               0
Allocated PE          0
PV UUID               n34AGF-NTQA-2v9M-1MqQ-oMOM-6PHK-wBUuGX

--- NEW Physical volume ---
PV Name               /dev/dasde1
VG Name
PV Size               2.29 GB
Allocatable           NO
PE Size (KByte)       0
Total PE              0
Free PE               0
Allocated PE          0
PV UUID               yln9kK-29O3-b7qL-nXf2-t9t7-tSwW-5AUF32

Creating a single volume group
The vgcreate command can be used to create a volume group named homevg from the two DASD. Use the vgdisplay command to verify:

```
# vgcreate homevg /dev/dasdd1 /dev/dasde1
Volume group "homevg" successfully created
# vgdisplay
--- Volume group ---
  VG Name               homevg
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               4.58 GB
  PE Size               4.00 MB
  Total PE              1172
  Alloc PE / Size       0 / 0
  Free  PE / Size       1172 / 4.58 GB
  VG UUID               6MbaFI-8Gkh-HNAV-OXZG-YcHs-r5uz-QJPWgm
```

In this example, there are 1172 free physical extents.

Creating a single logical volume
The lvcreate command is used to create a logical volume. The -l 1172 flag specifies to use all free extents, in this example. The -n homelv specifies the name of the new logical volume. The last argument homevg specifies the name of the volume group from which the logical volume will be created. Use the lvdisplay command to verify:

```
# lvcreate -l 1172 -n homelv homevg
Logical volume "homelv" created
```
Making a file system from the logical volume

Now you have a logical volume. Use the `mke2fs` command to create a file system out of it. The `-j` flag adds a journal so it will be of type ext3:

```
# mke2fs -j /dev/homevg/homelv
mke2fs 1.38 (30-Jun-2005)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
600288 inodes, 1200128 blocks
60006 blocks (5.00%) reserved for the super user
...
```

The file system created from the logical volume is now ready to be mounted.

12.1.5 Updating the file system table

You could now mount the file system manually. However if you add the mount to the file system table file, `/etc/fstab`, you can effectively test the change by using the `mount` command with only one argument. Make a backup copy then add the following line to the file:

```
# cd /etc
# cp fstab fstab.orig
# vi fstab
/dev/dasda1          /                    ext3       acl,user_xattr        1 1
/dev/dasdb1          swap                 swap       defaults              0 0
/dev/dasdc1          swap                 swap       defaults              0 0
/dev/homevg/homelv   /home                ext3       defaults              0 0
proc                 /proc                proc       defaults              0 0
sysfs                /sys                 sysfs      noauto                0 0
dbgsf                /sys/kernel/debug    debugfs    noauto                0 0
devpts               /dev/pts             devpts     mode=0620,gid=5       0 0
```

Mount the `/home` file system with one argument. Use the `ls` command to verify that there is no data in the logical volume except the `lost+found` directory. Use the `df -h` command to verify that it is mounted:

```
# mount /home
# ls /home
lost+found/
# df -h
Filesystem  Size  Used  Avail  Use%  Mounted on
/dev/dasda1  2.3G  904M  1.3G   42%  /
udev          122M  404K   121M   1%  /dev
```
When `/home/` can be mounted with one parameter, it shows that the syntax in `/etc/fstab` is correct.

### Moving data from existing file system

You may have noticed that there may be some data in the directory which will serve as the mount point. To complete the addition of the logical volume, it is recommended that you move any data from the existing directory to the new logical volume. First unmount the logical volume and view the contents of the `/home/` directory:

```
# cd /
# umount home
# ls home
mikem
```

In this example, there is only one subdirectory. You can move the contents from the root file system to the new logical volume a number of different ways. Following is one method:

```
# cd /home
# mkdir /tmp/home
# mv * /tmp/home
# cd ..
# mount /home
# cd /home
# mv /tmp/home/* .
# ls -F
lost+found/ mikem/
# rmdir /tmp/home
```

Edit the `/etc/zipl.conf` file and add your new disks there so that next time the system is rebooted it will find the disks.

```
# cd /etc
# vi zipl.conf  // change the dasd= string
# Modified by YaST2. Last modification on Sun Oct 29 17:18:11 UTC 2006
[defaultboot]
defaultmenu = menu

:menu
target = /boot/zipl
timeout = 10
prompt = 1
1 = ipl
2 = failsafe
default = 1

###Don't change this comment - YaST2 identifier: Original name: ipl###
[ipl]
target = /boot/zipl
image = /boot/image
ramdisk = /boot/initrd,0x1000000
parameters = "root=/dev/dasda1 dasd=100-104 TERM=dumb"
...
```

Now, run the `zipl` command so the changes are written to the boot record:

```
# zipl
Using config file '/etc/zipl.conf'
Building bootstrap in '/boot/zipl'
```
Building menu 'menu'
Adding #1: IPL section 'ipl' (default)
Adding #2: IPL section 'failsafe'
Preparing boot device: dasda (0100).
Done.

Even though you tested mounting the file system via reading the /etc/fstab file, you should test a reboot to verify the new logical volume is successfully mounted over /home/.

### 12.2 Extending an existing logical volume

This section describes the process of adding a new minidisk to an existing LVM. This is useful when your logical volume has run out of space. In this example, the `vgdisplay` command shows that the existing volume group is full.

```bash
# vgdisplay
--- Volume group ---
VG Name               homevg
...                    
Total PE              1172
Alloc PE / Size       1172 / 4.58 GB
Free PE / Size        0 / 0
VG UUID               6MbaFI-8Gkh-HNAV-OXZG-YcHs-r5uz-QJPWgm
```

First, repeat the steps at the beginning of this section to add a new minidisk. In this example, a minidisk at virtual address 105 is added of size 3338 cylinders.

When your system comes back, bring disk 105 online and check it is available:

```bash
# chccwdev -e 105
Setting device 0.0.0105 online
Done
# lsdasd
0.0.0100(ECKD) at (94: 0) is dasda : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0101(FBA ) at (94: 4) is dasdb : active at blocksize 512, 524288 blocks, 256 MB
0.0.0102(FBA ) at (94: 8) is dasdc : active at blocksize 512, 1048576 blocks, 512 MB
0.0.0103(ECKD) at (94: 12) is dasdd : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0104(ECKD) at (94: 16) is dasde : active at blocksize 4096, 600840 blocks, 2347 MB
0.0.0105(ECKD) at (94: 20) is dasdf : active at blocksize 4096, 600840 blocks, 2347 MB
```

Format the minidisk via the `dasdfmt` command and make a single partition via the `fdasd -a` command as you did before:

```bash
# dasdfmt -b 4096 -y -f /dev/dasdf
Finished formatting the device.
Rereading the partition table... ok
# fdasd -a /dev/dasdf
auto-creating one partition for the whole disk...
writing volume label...
writing VTOC...
rereading partition table...
```

### Creating a physical volume

Use the `pvcreate` command to create a physical volume from the minidisk:

```bash
# pvcreate /dev/dasdf1
Physical volume "/dev/dasdf1" successfully created
```
**Extending the volume group**

Use the `vgextend` command to extend the volume group into the new physical volume. Then, use `vgdisplay` to verify that the volume group has free space.

```
# vgextend homevg /dev/dasdf1
Volume group "homevg" successfully extended
# vgdisplay
--- Volume group ---
  VG Name     homevg
  ...        
  Total PE   1758
  Alloc PE / Size    1172 / 4.58 GB
  Free  PE / Size   586 / 2.29 GB
  VG UUID           6MbaFI-8Gkh-HNAV-0XZG-YcHs-r5uz-QJPWgm
```

Note there are 586 new free physical extents (PEs).

**Extend the logical volume and the file system**

Now that you have free space in the volume group, you can increase the size of the existing logical volume with the `lvextend` command. The `-l` option specifies the number extents to add. Finally, use the `resize2fs` command, which increases the size of the file system.

You can use the `df` command to show the file system size before and after you extend it as the following example shows:

```
# df -h | grep home
/dev/mapper/homevg-homelv         4.6G   129M  4.2G   3% /home
# lvextend -l +586 /dev/homevg/homelv
  Extending logical volume homelv to 6.87 GB
  Logical volume homelv successfully resized
```

**Attention:** On some systems this can be done while the file system is mounted, but on SLES 10, it appears that it must be unmounted:

```
# resize2fs -p /dev/homevg/homelv
resize2fs 1.38 (30-Jun-2005)
/dev/homevg/homelv is mounted; can't resize a mounted filesystem!
```

Unmount the `/home/` directory and first run `e2fsck` to verify the file system:

```
# umount /home
# e2fsck -f /dev/homevg/homelv
  e2fsck 1.38 (30-Jun-2005)
  Pass 1: Checking inodes, blocks, and sizes
  Pass 2: Checking directory structure
  Pass 3: Checking directory connectivity
  Pass 4: Checking reference counts
  Pass 5: Checking group summary information
  /dev/homevg/homelv: 11/600288 files (0.0% non-contiguous), 51656/1200128 blocks
```

Now use the `resize2fs` command to extend the logical volume, then remount `/home/`:

```
# resize2fs -p /dev/homevg/homelv
```
Resizing the filesystem on /dev/homevg/homelv to 1800192 (4k) blocks.
Begin pass 1 (max = 18)
Extending the inode table
The filesystem on /dev/homevg/homelv is now 1800192 blocks long.

Use the `df -h` command to show that the file system is now 2.2GB larger:

```
# df -h | grep home
/dev/mapper/homevg-homelv 6.8G 129M  6.4G   2% /home
```

**Add minidisk 105** to the parameter line in `/etc/zipl.conf` and run `zipl` command!

You have now added the new minidisk as free space to the existing logical volume. It is recommended to test your changes by rebooting your system:

```
# reboot
```

---

### 12.3 Centralizing home directories for LDAP users

If you have completed section 10.2, “Creating a virtual LDAP server” on page 160, you have a working LDAP authentication server on LINUX02 that provides centralized login authentication. If you have completed section 12.1, “Adding a logical volume” on page 179, you have a larger file system mounted over `/home` for storing user data. Now you can bring these together with the automount service to have a centralized `/home` file system. In this fashion, users can login to all virtual servers using the same credentials and being able to access the same data. Further, you can modify the golden image so that all newly cloned virtual servers will inherit this setup. Following is a block diagram:

![Figure 12-2  Block diagram of centralized LDAP authentication and automounted /home file system](image)
12.3.1 Configuring the NFS server

Configure LINUX02 to export /home over NFS. This is similar to the configuration of the controller in section 7.7.10, “Turning on the NFS server” on page 124. First install the nfs-utils RPM via the yast -i command:

```
# yast -i nfs-utils
```

Add one line to the bottom of the /etc/exports file to export the /home/ directory read/write:

```
# vi /etc/exports
# See the exports(5) manpage for a description of the syntax of this file.
# This file contains a list of all directories that are to be exported to
# other computers via NFS (Network File System).
# This file used by rpc.nfsd and rpc.mountd. See their manpages for details
# on how make changes in this file effective.
/home *(rw,sync)
```

Set the NFS service to start at boot time via the chkconfig command and for this session via the service command:

```
# chkconfig nfsserver on
# rcnfsserver start
Starting kernel based NFS server done
```

Test mounting the newly exported directory locally:

```
# ls /mnt
# mount localhost:/home /mnt
# ls /mnt
lost+found mikem
# umount /mnt
```

This shows that NFS server is set up and the /home/ directory is available for mounting.

12.3.2 Configuring the master image for LDAP and automount

In this section you will use chroot to enter an environment to modify the master image. You will configure it to point the virtual server running on LINUX02 for the following:

- LDAP authentication
- Automounting of /home/ via NFS

Start an SSH session to the controller. Use the chroot command to make the master image the current root:

```
# chroot /sles10master/
```

**Configuring master image for LDAP authentication**

Again, yast can be used to configure the LDAP authentication client. Select **Network Services** on the left side of the main screen, then **LDAP Client**:

```
# yast
```
On the panel that follows, use the Tab key to move to **Use LDAP** and press the **space bar** to select that choice, then enter IP address of your LDAP server, 129.40.178.122 in this example. You can either enter LDAP base DN manually or press **Fetch DN** and then **OK** in result window. This way you can make sure LDAP server is accessible:

![YaST Control Center](image)

Use the Tab key to move to **Finish** and press **Enter**. You will get a warning window, press **OK** there. Press the **Quit** button on the main window to quit YaST.

Your LDAP client should now be pointing to the LDAP server. Test it via the `id` command to the ldap user (**mikem** in this example):

```
# id mikem
uid=1000(mikem) gid=100(users) groups=100(users)
```

**Configuring automount**

Install the automount package named `autofs` via the `yast -i` command:

```
# yast -i autofs
...  
```

Add an entry to the bottom of `/etc/auto.master` for `/home`.

```
# cd /etc
```
This line specifies that when the /home/ directory is referenced, use the file /etc/auto.home to automatically mount it. Create the file auto.home and add one line to it:

```
# vi auto.home
*
129.40.178.122:/home/&
```

The asterisk (*) is used as a wildcard character to represent the local subdirectory. The ampersand (&) is used as the remote system name or the remote subdirectory. Whatever is in the local directory name field replaces the ampersand character.

Start autofs via rcautos for this session and via chkconfig at boot time:

```
# rcautos start
Starting service automounter done
# chkconfig autofs on
```

You have now configured the master image so autofs will mount the user’s home directory under /home/. Now exit the chroot’ed environment

```
# exit
```

Now that you are back into the controller environment, you can clone new virtual servers to test your changes. In this example two new virtual servers are cloned to the LINUX05 and LINUX06 user IDs:

```
# clone.sh linux05
... 
# clone.sh linux06
...
```

**Start an SSH session** to one of the new virtual servers and login as the LDAP user, mikem in this example:

```
login as: mikem
Password: mikem@lat125:~> pwd
/home/mikem
mikem@lat125:~> echo "this file was created from .125" > foo
```

Start an SSH session to the second server. Again login as the LDAP user. You should see the file created on the first server. Append a line to the file

```
login as: mikem
Password: mikem@lat126:~> cat foo
this file was created from .125
mikem@lat126:~> echo "this line was added from .126" >> foo
```
Back on the first server, the file should now also contain the line added from the second server:

```
mikem@lat125:~> cat foo
this file was created from .125
this line was added from .126
```

This shows that you have modified the master image so that cloned virtual servers have a centralized LDAP server for authentication and common home directories via NFS and automount.

## 12.4 Rescuing a Linux system

This section describes how to boot your Linux server into different modes for troubleshooting purposes. It covers booting Linux into single user mode, and also entering a rescue environment when you require more advanced troubleshooting.

### 12.4.1 Entering single user mode

Single user mode is helpful when you need to recover the root password, or if you are having problems while booting Linux into the default runlevel. To enter single user mode, first IPL your Linux server from the 3270 console. You will see a message similar to:

```
> zIPL v1.3.2 interactive boot menu
> 0. default (linux)
> 1. linux

Note: VM users please use '#cp vi vmsg <input>
```

Please choose (default will boot in 15 seconds):

Enter the `#cp vi vmsg` command to boot the desired menu option (zero in this example), followed by the number one for single user mode:

```
===> #cp vi vmsg 0 1
```

After some initial kernel boot messages, you will see:

```
[1A..done INIT: Entering runlevel: 1
  Boot logging started on /dev/ttyS0(/dev/console) at Tue Jan  9 14:42:02 2007
  Master Resource Control: previous runlevel: N, switching to runlevel: 1
  Sending all processes the TERM signal...
  done
  Sending all processes the KILL signal...
  done
  INIT: Going single user
  Master Resource Control: runlevel 1 has been reached
  INIT: Sending processes the TERM signal
  Terminated
  Give root password for login: <lnx4vm>

  lat124:~ #
```

In single user mode, you are logged in as the root user. You can use the `passwd` command to set the root password. All of the file systems in `/etc/fstab` are mounted, but networking has not been started. To exit single user mode, you can type `reboot`, or enter `init 3` to continue booting normally.
12.4.2 Entering a rescue environment

If you encounter errors mounting the root file system, or have other problems that prevent you from entering single user mode, you can enter a rescue environment. To enter into rescue environment you start installation of Linux again, but will not go further than activating DASDs.

To enter a rescue environment, initiate an interactive Linux installation. Because you will have to SSH into installation image during the installation you cannot use VNC because SSH server is not started in VNC type of install. You will have to use SSH. Perform the following steps:

1. Logon to LNXMAINT. Copy the SLES10 EXEC file to a new file named RESCUE EXEC, and copy the user's PARM-S10 file to a new file (LINUX04 RESCUE in this example):

```bash
=> copy sles10 exec d rescue =
=> copy linux04 parm-s10 d = rescue =
```

2. Next, edit RESCUE EXEC to point to the new RESCUE parmfile:

```bash
=> xedit RESCUE EXEC
...
00005 'PUN SLES10 KERNEL * (NOH'
00006 'PUN userid() ' RESCUE * (NOH'
00007 'PUN SLES10 INITRD * (NOH'
...
=> file
```

3. Edit the LINUX04 RESCUE file, replacing any VNC with SSH option:

```bash
=> xedit LINUX04 RESCUE
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
HostIP=129.40.178.124 Hostname=lat124.pbm.ihost.com
Gateway=129.40.178.254 Netmask=255.255.255.0
Broadcast=129.40.178.255 Layer2=0
ReadChannel=0.0.0600  WriteChannel=0.0.0601  DataChannel=0.0.0602
Nameserver=129.40.106.1 Portname=dontcare
Install=nfs://129.40.178.130/nfs/sles10/SLES-10-CD-s390x-GMC-CD1.iso
UseSSH=1 SSHPassword=lnx4vm
InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
```

4. Now logoff of LNXMAINT and logon to LINUX04. Answer no to IPL from DASD question, and run the RESCUE EXEC.

```bash
=> RESCUE
00: 0000003 FILES PURGED
00: RDR FILE 0004 SENT FROM LINUX04 PUN WAS 0004 RECS 076K CPY 001 A NOHOLD NO KEEP
00: RDR FILE 0005 SENT FROM LINUX04 PUN WAS 0005 RECS 0009 CPY 001 A NOHOLD NO KEEP
00: RDR FILE 0006 SENT FROM LINUX04 PUN WAS 0006 RECS 103K CPY 001 A NOHOLD NO KEEP
00: 0000003 FILES CHANGED
00: 0000003 FILES CHANGED
Linux version 2.6.16.21-0.8-default (geeko@buildhost) (gcc version 4.1.0 (SUSE Linux)) #1 SMP Mon Jul 3 18:25:39 UTC 2006
We are running under VM (64 bit mode)
Detected 2 CPU's
Boot cpu address  0
Built 1 zonelists
Kernel command line: ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
HostIP=129.40.178.124 Hostname=lat124.pbm.ihost.com
Gateway=129.40.178.254 Netmask=255.255.255.0
Broadcast=129.40.178.255 Layer2=0
ReadChannel=0.0.0600  WriteChannel=0.0.0601  DataChannel=0.
```
The install process directs you to telnet or SSH to the IP address of your Linux server to begin the first stage of the installation.

5. Use SSH to connect to the IP address and log in as root. Run `yast`. After choosing the language, accept licence agreement, choose Configure DASD Disks and Activate your disks. Do not leave this session and start another SSH connect. There you will be able to mount your root filesystem and do any needed corrections:

```
inst-sys:~ # mount /dev/dasda1 /mnt
inst-sys:~ # 11 /mnt
```

```
total 92
  drwxr-xr-x  2 root root  4096 Jan  5 06:44 bin
  drwxr-xr-x  3 root root  4096 Jan  5 06:26 boot
  drwxr-xr-x  5 root root  4096 Jan  5 09:46 dev
  drwxr-xr-x 58 root root  4096 Jan  9 14:42 etc
  drwxr-xr-x  2 root root  4096 Jun 16 2006 home
  drwxr-xr-x  9 root root  4096 Jan  8 12:26 lib
  drwxr-xr-x  5 root root  4096 Jan  8 13:36 lib64
  drwx------  2 root root  4096 Jan  5 06:22 lost+found
  drwxr-xr-x  2 root root  4096 Jun 16 2006 media
  drwxr-xr-x  2 root root  4096 Jun 16 2006 mnt
  drwxr-xr-x  3 root root  4096 Jan  5 06:43 opt
  drwxr-xr-x  2 root root  4096 Jan  5 06:22 proc
  drwx------ 11 root root  4096 Jan  9 14:00 root
  drwxr-xr-x  3 root root  8192 Jan  5 06:44 sbin
  drwxr-xr-x  4 root root  4096 Jan  5 06:22 srv
  drwxr-xr-x  3 root root  4096 Jan  5 06:22 sys
  drwxrwxrwt  4 root root  4096 Jan  9 14:41 tmp
  drwxr-xr-x 13 root root  4096 Jan  5 06:25 usr
  drwxr-xr-x 14 root root  4096 Jan  5 06:23 var
```
Common tasks

This part of the book discusses tasks common to z/VM, SLES 10 and RHEL 5.
Monitoring z/VM and Linux

Not everything that can be counted counts, and not everything that counts can be counted.
--Albert Einstein

This chapter briefly describes how to monitor z/VM and Linux. For a more thorough chapter on z/VM performance and monitoring, see Chapter 11, Monitoring performance and capacity, in the Manual Getting Started With Linux, SC24-6096 on the Web at:

http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/Shelves/hcsh2a70

There are a number of z/VM monitoring tools such as Computer Associates' VM:Monitor, IBM's z/VM Performance Toolkit and Velocity Software's ESALPS. The IBM z/VM Performance Toolkit is briefly described in this section.

For more information on Computer Associates' VM:Monitor, see:

http://www.ca.com/

For more information on Velocity's ESALPS, see:

http://www.velocitysoftware.com/

This chapter describes the following topics:

- “Using INDICATE and other commands” on page 197
- “The z/VM Performance Toolkit” on page 201
- “Monitoring Linux” on page 208
- “Registering Linux images with the Performance Toolkit” on page 210

13.1 Using INDICATE and other commands

z/VM has many commands to monitor the state of the system. CP INDICATE is the most commonly used, and there are other commands that are addressed.
### 13.1.1 Using the INDICATE command

z/VM has some basic commands such as `INDICATE`. There are many parameters that can be included. Use the command `HELP INDICATE` for a basic understanding and then press `F11` for help on each parameter.

**INDICATE LOAD**

If no parameter is specified `INDICATE LOAD` is the default option. There are two flavors of this, a class G and a class E. Class G users can use `INDICATE` to display recent contention for system resources, display environment characteristics and measurements of resources used by their virtual machine.

The output from user ID with class E privilege (e.g. `MAINT`, `OPERATOR`) is shown here. The lines are number for clarity of the description that follows:

```
===> ind load
  1 AVGPROC-038% 03
  2 XSTORE-000021/SEC MIGRATE-0001/SEC
  3 MDC READS-000068/SEC WRITES-000001/SEC HIT RATIO-099%
  4 PAGING-0031/SEC STEAL-000%
  5 Q0-00006(00000) DORMANT-00357
  6 Q1-00001(00000) E1-00000(00000)
  7 Q2-00001(00000) EXPAN-002 E2-00000(00000)
  8 Q3-00034(00000) EXPAN-002 E3-00000(00000)
 10 PROC 0000-038% PROC 0001-038%
 12 PROC 0002-038%
 14 LIMITED-00000
```

The `INDICATE LOAD` command gives a snapshot of current system performance. Except for the counts of virtual machines in various queues and the limited list, the values you see here are a smoothed average over the past 4 minutes. Areas where z/VM performance analysts tend to focus are the following:

- AVGPROC on line 1 gives the overall processor utilization, 38% in this example. The number following it is the number of on-line processors, 3 in this example. The individual processor utilization is shown on lines 10 and 11. Take a glance at these to see if they are somewhat balanced. There are cases where an imbalance is okay. This would include very low utilization scenarios or cases where there are not enough ready to run virtual processors to keep the physical processors busy. Line 2 describes paging to expanded storage. Most z/VM systems on z900 class machines can sustain 1000s of paging operations a second without any problems. The `MIGRATE` rate is the number of pages per second being moved from expanded storage out to paging space on DASD. A healthy system will have a `MIGRATE` rate significantly lower than the `XSTORE` rate. However, there are times the `MIGRATE` value will spike for brief periods of time.

- Minidisk cache (MDC) statistics are given on the third line. The effectiveness of MDC can be judged by the combination of the `READS` rate and the `HIT RATIO`. If both are high, then a large number of physical I/Os are avoided due to the MDC feature. However, a high `HIT RATIO` with a low value for the `READS` rate is not good (it doesn't matter much if you have a 100% hit ratio, but are doing only 1 I/O per second).

- Line 4 describes more storage (memory) management. The PAGING rate is important. Higher values will often impact performance. The STEAL percentage is often misleading. This is basically the percentage of pages taken from guests that z/VM believes are non-dormant. Since some guests have periodic timers going off, they appear to be active to z/VM even when relatively idle. Pages taken from these guests are considered stolen. So there are scenarios where a system only has active guests, in which case all pages
taken would be considered stolen. Bearing this in mind, if a high STEAL value is observed, the paging rate needs to be checked. If the paging rate is low, then the STEAL value is not important.

- On lines 5 through 8 you also see a series of counters that represent the users in various queues. The z/VM scheduler classifies work into 3 different classes (1 through 3) and a special class of zero. So the Column of Q_x values and E_x represent the virtual machines in the dispatch list and the eligible list. The most important value here to validate is that there are no virtual machines in the Eligible list: E1, E2, E3; this implies z/VM has stopped dispatching some virtual machines to avoid over committing resources. Do not worry about the values in parenthesis.

**INDICATE QUEUES EXP**

Another useful command to understand the state of the system is the `INDICATE QUEUES EXP`. Following is an example:

```
  ==> ind q exp
  DATAMGT1    Q3 AP  00000537/00000537 .... -2.025 A02
  BITNER      Q1 R00 00000785/00000796 .I... -1.782 A00
  EDLNX4       Q3 PS  00007635/00007635 .... -1.121 A00
  TCPIP        Q0 R01 00004016/00003336 .I... -.9324 A01
  APCTEST1     Q2 IO  00003556/00003512 .I... -.7847 A01
  EDLRWK20     Q3 AP  0001495/00001462 .... -6.996 A01
  EDL         Q3 IO  0000918/00000902 .... -2.409 A01
  EDLRWK11     Q3 AP  0002323/00022299 .... -0.183 A00
  EDLRWK18     Q3 IO  0001052/00000388 .... -0.047 A00
  EDLRWK4      Q3 AP  0004792/0002295 .... -0.055 A01
  EDLRWK8      Q3 AP  0004804/0004797 .... -0.089 A02
  EDLRWK16     Q3 AP  0002378/0002378 .... -0.017 A02
  EDLRWK2      Q3 AP  0005544/0002956 .... -0.360 A00
  EDLRWK12     Q3 AP  0004963/0002348 .... -0.077 A01
  EDLRWK6      Q3 IO  0000750/0000302 .... -0.069 A02
  EDLRWK3      Q3 AP  0005098/0005096 .... -0.099 A02
  EDLRWK17     Q3 AP  0004786/0004766 .... -0.106 A01
  EDLRWK9      Q3 AP  0002372/0002334 .... -0.1107 A02
  EDLRWK5      Q3 IO  0002376/0002376 .... -0.1205 A01
  EDLRWK14     Q3 AP  0002426/0002323 .... -0.1238 A02
  EDLLIB19     Q3 IO  0001226/00011100 .... -0.1309 A02
  EDLRWK19     Q3 AP  0002322/0002298 .... -0.1705 A00
  EDLRWK15     Q3 AP  0002839/0002781 .... -0.2205 A02
  EDLRWK1      Q3 AP  0002969/0002935 .... -0.2491 A02
```

This is another class E command and displays the virtual processors (a single virtual machine may have multiple virtual processors) what queue (dispatch list, eligible, limit list) they are in and what state they are. This is a snapshot in time. Again you want to make sure there are not any virtual machines in the eligible list. Normal virtual processors in the dispatch list will be Q_x (x=1,2,3). Eligible list would be marked as E_x. The third column in the example also gives state of virtual processor. This can be helpful to get a feel for how the virtual processors might be constrained. Virtual processors that are actually running at the snapshot are marked with and RNN where NN is the processor number they are on. An R without a number means the virtual processor is ready to run but there is not an available processor. *(Note: the virtual machine that issues the INDICATE command will always be one of the running machines).* Other states are documented in the help for `IND Q EXP`. One doesn't have to be concerned about the other columns unless detailed analysis is required or if IBM support requests it. Also, always remember that is just a snapshot in time so often repeating this command over time can give a more accurate picture of your z/VM system.
13.1.2 Using other basic commands

Some other useful basic commands are briefly mentioned. All examples are shown from the MAINT user ID. The results will be different for users with fewer privileges.

**Getting help**

To get help on the system use the `HELP` command. Sometimes it’s hard to find help for exactly the command you’re looking for. Some useful help commands are as follow

```
=> help       // for basic help
=> help cp menu // for a menu of all CP commands
=> help cpquery // for a menu of all CP QUERY command
=> help cpset  // for a menu of all CP SET commands
```

**Determining who is logged on**

To see who is logged on to the system use the `QUERY NAMES` command. For example

```
=> q n
LINUX06 - DSC, LINUX04 - DSC, LINUX03 - DSC, LINUX07 - DSC
LINUX01 - DSC, SLES9 - DSC, FTPSERVE - DSC, DTCVSW2 - DSC
DTCVSW1 - DSC, TCP/IP - DSC, OPERSYM - DSC, DISKACNT - DSC
EREP - DSC, OPERATOR - DSC, MAINT -L0005
VSM - TCP/IP
```

**Determining storage or memory**

To see how much central and expanded storage (memory) use the `QUERY STORAGE` and `QUERY XSTOR` commands. For example:

```
=> q stor
STORAGE = 3G
```

```
=> q xstor
XSTORE= 1024M online= 1024M
XSTORE= 1024M userid= SYSTEM usage= 97% retained= 0M pending= 0M
XSTORE MDC min=0M, max=1024M, usage=96%
XSTORE= 1024M userid= (none) max. attach= 1024M
```

**Determining processors or CPUs**

To see how many processors (CPs, IFLs, CPUs) you have, use the `QUERY PROCESSORS` command. For example:

```
=> q proc
PROCESSOR 00 MASTER
PROCESSOR 01 ALTERNATE
PROCESSOR 02 ALTERNATE
PROCESSOR 03 ALTERNATE
```

**Determining software level**

To determine what level of CP your system is at, use the `QUERY CPLEVEL` command. For example:

```
=> q cplevel
z/VM Version 5 Release 1.0, service level 0401 (64-bit)
Generated at 08/31/04 17:33:32 EST
IPL at 03/10/05 14:42:02 EST
```

**Determining system cylinder allocation**

The `QUERY ALLOC MAP` command shows you the system’s allocate of spool, paging and directory space. For example:

```
=> q alloc map
```
### Determining DASD, OSA and virtual resources

The `QUERY DASD` and `QUERY DASD FREE` commands will show you what DASD is assigned to the system and what DASD is free to be assigned. Similarly the `QUERY OSA` and `QUERY OSA FREE` commands will report on the OSA resources. Finally, the `QUERY VIRTUAL ALL` command can be useful. Following is the short form of these commands without any output shown:

```plaintext
===> q da
===> q da free
===> q osa
===> q osa free
===> q v all
```

### 13.2 The z/VM Performance Toolkit

To use the z/VM Performance Toolkit, the product must be ordered. You should only configure the product if you have ordered it.

Much more detail can be found in the following books:

- `z/VM Performance Toolkit`, SC24-6136, on the Web starting at the z/VM 5.2 bookshelf:
  

  Search for `Toolkit` on that page.

- *The Program Directory for Performance Toolkit for VM*, GI11-2854-00, on the Web at
  

- The IBM Redbook *Linux on IBM zSeries and S/390: Performance Toolkit for VM*, SG24-6059, on the Web at:
  

The section that follow describe how to set up and use the IBM Performance Toolkit very briefly:

- “Configuring the z/VM Performance Toolkit” on page 201
- “Using the z/VM Performance Toolkit” on page 205

### 13.2.1 Configuring the z/VM Performance Toolkit

The Performance Toolkit is installed with z/VM. Configuration is described in the Program Directory. Following is a summary of how to turn it on. Again, you should configure the product only if you have ordered it. You can query which priced products are enabled via the `QUERY PRODUCT` command:

```plaintext
===> q product
DMSVML2060I LNXMAINT 192 linked as 0120 file mode Z
Ready; T=0.01/0.01 11:35:43
Product  State    Description
5VMDIR10 Disabled 00/00/00.00:00:00.$BASEDDR DIRECTORY MAINTENANCE FL 510
```
To enable it, logon to MAINT and enter the following command:

```bash
=> service perftk enable
VMFSRV2760I SERVICE processing started
...
VMFSUT2760I VMFSUFTB processing started
VMFSUT2760I VMFSUFTB processing completed successfully
VMFSRV2760I SERVICE processing completed successfully
```

You should see a few screens of messages scroll by and finally the success messages shown above. This will enable the Performance Toolkit for the current z/VM session. To enable it across IPLs, modify the SYSTEM CONFIG file and uncomment the two lines changing DISABLED to ENABLED and reIPL your system:

```bash
=> cprel a
CPRELEASE request for disk A scheduled.
HCPZAC6730I CPRELEASE request for disk A completed.
=> link * cf1 cf1 mr
=> acc cf1 f
=> x system config f
====> /5vmptk10
PRODUCT PRODID 5VMPTK10 STATE ENABLED DESCRIPTION '00/00/00.00:00:00.$BASEDDR PERFORMANCE TOOLKIT FOR VM'
...
```

Check your changes for syntax errors and put the CP disk back online

```bash
=> acc 193 g
=> cpsyntax system config f
CONFIGURATION FILE PROCESSING COMPLETE -- NO ERRORS ENCOUNTERED.
=> rel f (det
DASD OCF1 DETACHED
=> cpacc * cf1 a
CPACCESS request for mode A scheduled.
HCPZAC6732I CPACCESS request for MAINT's OCF1 in mode A completed.
```

### 13.2.2 Configuring Web Browser support

Once the product is enabled, the TCPIP profile must be modified to add browser capabilities for the Performance Toolkit.

1. Logon to TCPMAINT. Edit the `<systemID>` TCPIP D file and search for the string reserve ports. This is where z/VM TCP/IP ports are reserved

```bash
=> x <systemID> tcpip d
====> /port
```

2. Add the following line under the PORT entries:

```bash
... PORT
  20  TCP FTPSERVE  NOAUTOLOG ; FTP Server
  21  TCP FTPSERVE ; FTP Server
  23  TCP INTCLIEN ; TELNET Server
 ; 25  TCP SMTP ; SMTP Server
 ; 53  TCP NAMESRV ; Domain Name Server
 ; 53  UDP NAMESRV ; Domain Name Server
 ; 67  UDP BOOTPD ; BootP Server
```
3. Save your changes. The TCPIP user ID needs to be recycled in order for our changes to take effect. You can FORCE and XAUTOLOG TCPIP from a console. Alternatively, if you are in position to reIPL the system, you can issue the command shutdown reipl iplparms cons=sysc.

When the system comes back, logon to TCPMAINT and check if everything was successful by issuing the NETSTAT CLIENTS command. You want to see that the service PERFSVM is a client (listening). This should be shown after a few screens of output:

```
===> netstat clients
... Client: PERFSVM Authorization: {none}
Notes Handled: none
Last Touched: 0:00:25
Vmcf error count: 0
```

The output of the command is lengthy, you want to make sure you have an entry for PERFSVM.

### 13.2.3 Configuring PERFSVM

The PERFSVM user ID is the Performance Toolkit service machine.

1. **Logon to PERFSVM.** If you successfully enabled the product, you should be put in a Performance Toolkit session and see the following text at the top of the screen:

```
FCX001             Performance Toolkit for VM
FCXBAS500I Performance Toolkit for VM FL510 BASE
Monitor event started -- recording is activated
Monitor sample started -- recording is activated
```

2. Press **PF12 twice** to get to a CMS prompt.

3. Copy the PROFILE XEDIT from the MAINT 191 disk so XEDIT sessions will have a common interface among user IDs.

   a. Use theVMLINK command to both link the disk read-only and access it as the highest available file mode. The default read password is read, however, if you changed your passwords as described in section 4.9.1, “Changing passwords in USER DIRECT” on page 60, then it will be lnx4vm.

   ```
   ==> vmlink maint 191
   ENTER READ PASSWORD:
   lnx4vm
   DMSVML2060I MAINT 191 linked as 0120 file mode Z
   ```

   b. Copy the PROFILE XEDIT to the A disk:

   ```
   ==> copy profile xedit z = = a
   ```

4. Copy the default configuration files, which are on PERFSVM's D disk, to your A disk:

   ```
   ==> copy * * d = = a
   ```

5. The main configuration file is FCONX $PROFILE. Edit that file and search for the string VMCF. This should take you to line 173 where the next 4 lines are comments starting with an *.

Perform the following changes:

- Uncomment the second and fourth line by changing *C to FC
- Change IDTEST PASSFILE to IDTEST CP on the fourth line
– Add the text FC MONCOLL LINUXUSR ON after the fourth line

The modified lines should be as follows:

```bash
===> x fconx $profile a
===> /vmcf
* Following command activates VMCF data retrieval interface
FC MONCOLL VMCF ON
* Following command activates Internet interface
FC MONCOLL WEBSERV ON TCPIP TCPIP 81 IDTEST CP
FC MONCOLL LINUXUSR ON
* Following command activates Internet interface with SSL
*C MONCOLL WEBSERV ON SSL TCPIP TCPIP 81 IDTEST RACF
...
```

6. Save your changes. The line you added tells the Performance Toolkit to collect Linux performance data.

7. Create a remote data retrieval authorization file – replace `<systemID>` with your system name:

```bash
===> x fconrmt authoriz
====> a 2
<systemID>  PERFSVM S&FSERV
<systemID>  MAINT    DATA CMD EXCPMSG
```

8. Create a system identification file – replace `<systemID>` with your z/VM system name:

```bash
===> x fconrmt systems
====> a
<systemID>  PERFSVM ESA N FCXRES00
```

9. Create a Linux system definition file. Add the TCP/IP addresses of your Linux system(s). The following example shows adding seven Linux virtual servers:

Use port 8803 for Linux performance data:

```bash
===> x fconx linuxusr
LINUX01 129.40.178.121:8803
LINUX02 129.40.178.122:8803
LINUX03 129.40.178.123:8803
LINUX04 129.40.178.124:8803
LINUX05 129.40.178.125:8803
LINUX06 129.40.178.126:8803
LINUX07 129.40.178.127:8803
```

10. Edit the PROFILE EXEC file and uncomment the five MONITOR SAMPLE and the two MONITOR EVENT statements:

**Before:**

```bash
...```
Chapter 13. Monitoring z/VM and Linux

11. Save your changes with the **FILE** subcommand.

You should now be ready to run the Performance Toolkit.

### 13.2.4 Starting the z/VM Performance Toolkit

To start the Performance Toolkit enter the following command:

```bash
===> perfkit
```

The Performance Toolkit should now be configured and running.

### 13.2.5 Using the z/VM Performance Toolkit

The Performance Toolkit can be used via a Web browser or 3270 interface.

**Using a Web browser interface**

To use the Web-enabled Performance Toolkit, perform the following steps:

1. Point a browser to your z/VM system at port 81. For example:

   ```plaintext
   http://129.40.178.120:81
   ```

2. You should see your system on the **Web Session Setup** screen. Click it and you will be presented with the Web Server Logon screen.
Remote Performance Monitoring Session Setup
Web Server Logon

You are connected to the data retrieval interface of the Performance Toolkit for VM on system LNVM52. Data retrieval authorization is based on your VM user identification on that system. Please enter your userid and password (CP).

VM UserID: maint  Password:  submit

Figure 13-1  Performance Toolkit logon screen

3. Enter any valid user ID and password. MAINT can be used.
4. You should see the Central Monitoring System Load Overview with your system name on the left side.
5. Click your system name and you should see the Initial Performance Data Selection Menu screen as shown in Figure 13-2 on page 206.

Figure 13-2  Browser interface to the Performance Toolkit
Using a 3270 interface
Logon to PERFSVM. Run the PROFILE EXEC and you should be put into the Performance Toolkit for z/VM environment. The subcommand monitor should present the panel shown below.

```plaintext
===> profile
FCXBA500I Performance Toolkit for VM FL510 BASE
Monitor event started -- recording is activated
Monitor sample started -- recording is activated

... 
FCX001 Performance Toolkit for VM
FCXBA500I Performance Toolkit for VM FL510 BASE
HCPMOF6229E Monitor event collection is already active.
HCPMOG6229E Monitor sample collection is already active.

Command ===> monitor
```

<table>
<thead>
<tr>
<th>FCX124</th>
<th>Performance Screen Selection (FL510 BASE )</th>
<th>Perf. Monitor</th>
</tr>
</thead>
<tbody>
<tr>
<td>General System Data</td>
<td>I/O Data</td>
<td>History Data (by Time)</td>
</tr>
<tr>
<td>2. Storage utilization</td>
<td>12. Control units</td>
<td>32. History data files*</td>
</tr>
<tr>
<td>3. Storage subpools</td>
<td>13. I/O device load*</td>
<td>33. Benchmark displays*</td>
</tr>
<tr>
<td>5. System counters</td>
<td>15. Cache extend. func.*</td>
<td>35. System summary*</td>
</tr>
<tr>
<td>6. CP IUCV services</td>
<td>16. DASD I/O assist</td>
<td>36. Auxiliary storage</td>
</tr>
<tr>
<td>7. SPOOL file display*</td>
<td>17. DASD seek distance*</td>
<td>37. CP communications*</td>
</tr>
<tr>
<td>8. LPAR data</td>
<td>18. I/O prior. queueing*</td>
<td>38. DASD load</td>
</tr>
<tr>
<td>A. Shared data spaces</td>
<td>1A. I/O config. changes</td>
<td>3A. Paging activity</td>
</tr>
<tr>
<td>B. Virt. disks in stor.</td>
<td></td>
<td>3B. Proc. load &amp; config*</td>
</tr>
<tr>
<td>C. Transact. statistics</td>
<td>User Data</td>
<td>3C. Logical part. load</td>
</tr>
<tr>
<td>D. Monitor data</td>
<td>21. User resource usage*</td>
<td>3D. Response time (all)*</td>
</tr>
<tr>
<td>E. Monitor settings</td>
<td>22. User paging load*</td>
<td>3E. RSK data menu*</td>
</tr>
<tr>
<td>F. System settings</td>
<td>23. User wait states*</td>
<td>3F. Scheduler queues</td>
</tr>
<tr>
<td>G. System configuration</td>
<td>24. User response time*</td>
<td>3G. Scheduler data</td>
</tr>
<tr>
<td>H. VM Resource Manager</td>
<td>25. Resources/transact.*</td>
<td>3H. SFS/BFS logs menu*</td>
</tr>
<tr>
<td>I. Exceptions</td>
<td>26. User communication*</td>
<td>3I. System log</td>
</tr>
<tr>
<td>K. User defined data*</td>
<td>27. Multitasking users*</td>
<td>3K. TCP/IP data menu*</td>
</tr>
<tr>
<td>L. User communication*</td>
<td>28. User configuration*</td>
<td>3L. User communication</td>
</tr>
<tr>
<td>M. User wait states</td>
<td>29. Linux systems*</td>
<td>3M. User wait states</td>
</tr>
</tbody>
</table>

Drilling down into report screens
You should now be able to use the active report screens. To drill down into these screens, move the cursor to any of the titles that are active (active titles display the number or letter in white, inactive titles are in green). Some of the more useful report screens to drill down into are:

21. User resource usage
22. User paging load
23. User wait states
24. User configuration
25. Linux systems
13.3 Monitoring Linux

To monitor Linux performance data, a data gatherer process must be running. There are different ways of gathering this data. Data can be gathered in the kernel and as a user application. SuSE SLES9, SLES10 and Red Hat RHEL 4 and RHEL 5 have been enabled for the kernel to gather performance data. There is a package called the Linux RMF PM Data Gatherer (also called rmfpms) that runs as a user application. Both of these data gatherers work in conjunction with the IBM z/VM Performance Toolkit.

13.3.1 Monitoring Linux with rmfpms

As a user application, the Linux RMF PM Data Gatherer (rmfpms) can be used. Currently it is not part of an IBM product and is intended for evaluation purposes only. A description of rmfpms is as follows:

"rmfpms is a modular data gatherer for Linux®. The gathered data can be analyzed using the RMF PM client application. The performance data is accessible through XML over HTTP so you can easily exploit it in your own applications."

The following Web site is a starting point:


To download the data gatherer, scroll down and look for the following text and links:

* 31 bit data gatherer (kernel24 - 630 KB, kernel26 - 1040 KB).
* 64 bit data gatherer (kernel24 - 650 KB, kernel26 - 666 KB).

You can download the appropriate gatherer via a browser, or, if you have access to the Internet you can use an FTP client. You will want one of two files depending on whether you have a 31-bit or 64-bit kernel:

rmfpms_s390_kernel26.tgz - for 31-bit distributions
rmfpms_s390x_kernel26.tgz - for 64-bit distributions

Following is an example of downloading the tar file for 31-bit distributions directly from the Internet. Get an SSH session on the controller or any other virtual server. Change to the home directory via the cd command with no argument. Download the appropriate tar file with the wget command. For a 64-bit SLES 9 and for SLES 10 use:

```bash
# cd
```

For RHEL 4 and RHEL 5 use:

```bash
```

Untar the file with the tar command and change to the rmfpms/ directory:

```bash
# tar xzf rmfpms_s390x_kernel26*.tgz
# cd rmfpms/
# ls -aF
./ .rmfpms_config README bin/ enable_autostart*
./ .rmfpms_config_autostart autostart_rmfpms doc/
```

You should now be able to start rmfpms in the bin/ directory with the following command:

```bash
# bin/rmfpms start
Creating /opt/IBM/rmfpms/.rmfpms ...
Starting performance gatherer backends ...
DDSRV: Functionality Level=2.008
```
DDSRV: Reading exceptions from gpmexsys.ini and gpmexusr.ini.

DDSRV: Server will now run as a daemon process.

done!

Once it is running, you can view the performance data from a browser pointing to the Linux image and port 8803 as shown in Figure 13-3.

![Browser view of rmfpms interface](image)

**Figure 13-3  Browser view of rmfpms interface**

You can also register Linux images with the Performance Toolkit - see section 13.4, “Registering Linux images with the Performance Toolkit” on page 210.

### 13.3.2 Monitoring Linux performance data from the kernel

To monitor Linux performance data directly from the kernel, the following must be true:

1. The **APPLMON** option must be set in the user directory.
2. Applmon data monitoring must be built into the kernel.

The first requirement should be true as the **OPTION APPLMON** was set for the controller and for Linux user IDs in earlier sections.

For the second requirement, SLES 9, SLES 10 and RHEL 5 now have this function built in. Details of this function are described in Chapter 15, *Linux monitor stream support for z/VM* in the manual *Device Drivers, Features, and Commands* for the October 2005 stream, on the Web at:


A quick description of how to use this built-in monitoring function follows.
There are three modules that are built into the kernel but are not loaded by default. They are named appldata_mem, appldata_os and appldata_net_sum. You can verify that they are not loaded with the `lsmod` and `grep` commands:

```bash
# lsmod | grep appldata
```

There is no output so no modules with the string `appldata` are loaded. Load those modules now via the `modprobe` command and verify they have been loaded:

```bash
# modprobe appldata_mem
# modprobe appldata_os
# modprobe appldata_net_sum
```

Now if you repeat the `lsmod` command, you should see the following:

```bash
# lsmod | grep appldata
appldata_net_sum       20064  0
appldata_os            21512  0
appldata_mem           20112  0
```

The directory in the virtual `/proc/` file system where the monitoring variables exist is `/proc/sys/appldata/`. In this directory there are five files as follow:

- **timer**: Controls whether any data gathering is in effect.
- **interval**: Sets the interval, in milliseconds, that samples will be taken.
- **mem**: Controls the memory data gathering module
- **os**: Controls the CPU data gathering module
- **net_sum**: Controls the net data gathering module

To turn on the built in kernel monitoring, use the `echo` command to send a non-zero value into four of the five monitoring variables in the `/proc/` virtual file system:

```bash
# echo 1 > /proc/sys/appldata/timer
# echo 1 > /proc/sys/appldata/mem
# echo 1 > /proc/sys/appldata/os
# echo 1 > /proc/sys/appldata/net_sum
```

Built-in kernel monitoring should now be turned on. You may only want to leave the monitoring on for specific periods of time. As Linux monitoring data is captured, the Performance Toolkit’s minidisk space can fill up relatively quickly.

### 13.4 Registering Linux images with the Performance Toolkit

To register Linux images that have performance data gathering enabled, `logon` to `PERFSVM` and create a file `FCONX LINUXUSR A` and add Linux user ID/IP address:port pairs. The following example shows adding the `SLES9X` and `LINUX01` user IDs to this file:

```bash
===> x fconx linuxusr a
====> a 3
*Linux-ID IP address for DDS Interface:Port
SLES9X <129.40.178.130>
LINUX01 <129.40.178.121>
```

Restart the performance toolkit via the `profile` command. After the system has had some time to collect data, you should be able to use the Performance Toolkit to monitor Linux systems that have both monitoring data being captured and an entry in the `FCONX LINUXUSR` file. To view that data, drill down into menu 29, Linux systems.
13.5 Monitoring z/VM and Linux via a Web application

A basic Web application is written to demonstrate one way to view data of both z/VM and the Linux virtual servers. The system is named DAZL, an acronym for Data About z/VM and Linux (yes, a name shamelessly force fit to come up with a cool acronym :)). To supply the DAZL information, a Web Server is run on the controller that makes available some HTML pages in the DocumentRoot directory and some bash shell scripts in the cgi-bin/ directory. If you have been progressing sequentially through the book, the Web server should be configured by now. If not, see chapter 7, especially the section “Configuring Apache for DAZL” in section 7.7.

A philosophy of DAZL is to keep common information grouped into pages about one screen in size, utilizing HTML tables to maximize the real estate of a browser screen. This application is not a heavily tested, nor is it a professional monitoring system. Rather it is a set of simple Web pages and scripts that can be a start of a home-grown monitoring tool, if you choose to investigate it.

This section shows some screen captures of the Web application, then the application’s relatively simple design is described in some detail in section 13.5.3, “DAZL design” on page 220. All source code is included in the controller RPM described in.

**Important:** DAZL is not a strategic direction for monitoring z/VM and Linux. A more strategic solution from IBM’s point of view is IBM Director and z/VM Center. See section 1.4, “IBM Director and z/VM Center Extension” on page 5.

### 13.5.1 Overall design

Following is a block diagram of the overall design of DAZL from an execution point of view:
13.5.2 DAZL screen captures

The main Web page is shipped with the file name index.html. If you followed all the steps in section It allows you to generate basic pages or reports about:

- **Linux systems**
- **z/VM DASD**
- **z/VM status**

Each Linux virtual server that was cloned
Each volume in the USER DISKMAP report on the MAINT 191 disk
Output of simple z/VM query commands

**DAZL home page**

Following is a screen capture of the DAZL home page:

![Figure 13-5 DAZL main page]

### z/VM status page

If you click the **z/VM status** button on the right, you will get some basic information about your z/VM system from the following commands:

- `IND`
- `Q ALLOC MAP`
- `Q NAMES`
- `Q STOR, Q XSTOR`
- `Q CLEVEL`
- `Q VSWITCH`

These values are obtained via the Linux `vmcp` interface to z/VM CP commands. Following is a screen capture of the z/VM status page:
Figure 13-6  z/VM status page

From this page, you can drill down to get more detailed information. From the Indicate Load table cell, you can drill down to view the output of INDICATE QUEUES EXP. From the QUERY VSWITCH table cell, you can drill down to view the output of the QUERY VSWITCH DETAILS and QUERY VSWITCH ACCESS commands.

**DASD report**

If you click on the DASD report button in the center of the main page, you will get a report on the z/VM system volumes that comprise your system. This report is another way of displaying the contents of the USER DISKMAP file. The accuracy of the report generated relies on the USER DISKMAP file being up to date under /backup/vm/USER.DISKMAP on the controller. There is a checkbox that allows update this file on Linux from the MAINT 191 disk. Clicking that checkbox will copy the USER DISKMAP file from z/VM, however, it still relies on the file being up to date. If you have modified the USER DIRECT file and run DIRECTXA but not run DISKMAP, then there is a good chance the data will be out of date (a good convention to adopt is to always run DISKMAP before running DIRECTXA).

Following is a screen capture of the DASD report
For each DASD in the report, you can drill down into the **Volume Label** column to get the text of that volume’s report in the **USER DISKMAP** file, and into the **Device Address** column to get the output of the CP command `QUERY DASD DETAILS <vaddr>` for that volume.

### Linux report

If you click on **Linux report** in the left side of the main page, you will get a report on the Linux clones. It is generated starting from data under the `/backup/linux/` directory. When the `clone.sh` script clones a new Linux image, it also creates a directory of the form `<userID>-on-<TCP.IP.address>/` under `/backup/linux/`. For example, if you clone a virtual server to **LINUX01** with an IP address of 129.40.178.121, the directory created will be `/backup/linux/LINUX01-on-129.40.178.121/`. A row is created in the report's table for each directory, assuming they are associated with Linux systems. If this directory structure does not reflect the existing Linux images, the report will not be correct.

For each directory found, the user ID and TCP/IP address are extracted from the name. The existence of these directories drive the main loop of the report generation. Each TCP/IP address that is extracted from the directory structure is pinged. If the ping is successful, the table cell entries for that virtual server are displayed in green and made to be drill down links. If the ping is not successful, the table cell entries are displayed as plain red text. The **Sitar Cron** entries are drawn in green if the `sitar.html` file is found in each virtual server's `/etc/` directory and in red if it is not.

Following is a screen capture of the Linux report:

---

For each directory found, the user ID and TCP/IP address are extracted from the name. The existence of these directories drive the main loop of the report generation. Each TCP/IP address that is extracted from the directory structure is pinged. If the ping is successful, the table cell entries for that virtual server are displayed in green and made to be drill down links. If the ping is not successful, the table cell entries are displayed as plain red text. The **Sitar Cron** entries are drawn in green if the `sitar.html` file is found in each virtual server's `/etc/` directory and in red if it is not.

Following is a screen capture of the Linux report:
Linux report drill down pages

From the Linux report page, you can drill down into the following columns:

- **z/VM User ID**: z/VM-related data
- **Linux host name**: Linux non-networking related data
- **IP address**: Linux networking related data
- **Sitar data**: Output of Sitar - either from the nightly cron run, or a new report in real time

In the right-most column, **Description/Log**, you can keep a record of each virtual server. The data is retrieved from and stored in the file `descLog.txt` in each of the virtual server's directories under `/backup/linux/`. These files can be viewed and edited directly from the Web application.

**z/VM User ID drill down**

Following is a screen capture of the z/VM related data when drilling down under the **z/VM user ID** column:
The first table cell in the upper left corner shows the directory entry of the specified user ID extracted from the USER DIRECT file. This file must be up to date on the controller. There is a checkbox, **Refresh /backup/vm**, on the DAZL home page that allows you to update all z/VM configuration files including the USER DIRECT file. The table cell in the upper right shows the output of the **INDICATE USER** command for the specified user ID. The bottom two table cells display the contents of the installation parameter files with file types of **PARMFILE** for SLES 9 and **PARM-S10** for SLES 10, or **PARM-RH5** and **CONF-RH5** for RHEL5.

**Linux host name drill down**

Following is a screen capture of the z/VM related data when drilling down under the **Linux host name** column:
Figure 13-10  Linux information drill down report

This report gives the output from the following commonly run commands to query the status of Linux systems.

- `df -h`  Display file system disk usage
- `mount`  Display mounted file systems
- `runlevel`  Display previous and current run levels
- `w`  Show who is logged on
- `uname`  Display system information about this distribution
- `free -m`  Display megabytes of free and used memory
- `ps tree`  Display a tree of processes
- `lsdasd`  List Direct Access Storage Devices (DASD)

### IP address drill down

Following is a screen capture of the z/VM related data when drilling down under the **IP address** column:
This report is generated from the output of the following four network-related commands:

- `ifconfig eth0`: Display information about the interface eth0
- `ip link show`: Display IP link information
- `netstat -tan`: Display TCP network information
- `iptables -L`: List IP tables (firewall) rules

**Sitar data drill down**

You can drill down into the **Cron** link which will use the Sitar output file that should be generated by a cron job each night. Also you can drill down into the **New** link which will generate a Sitar HTML file on demand. Following is a screen capture of the z/VM related data when drilling down under the **Sitar** column:
Another useful feature of this Web application is the ability to maintain a flat text file for each of the Linux virtual servers from a browser. The basic premise is that the first line of the file is an overall description of the virtual server and the remainder of the file can be used to log changes. This feature utilizes Asynchronous Java and XML (AJAX) for pushing and pulling changes between the browser and the Web server without having to update the entire page.

The flat text files are maintained on the controller under the directory `/backup/linux/` with one file named `descLog.txt` under the each Linux subdirectory (of the form `<userID>-on-<TCP.IP.address>`). For example, a Linux system running on the user ID LINUX01 with a TCP/IP address of 129.40.178.121 would have a flat text file named `/backup/linux/LINUX01-on-129.40.178.121/descLog.txt`.

You can access these files manually via an SSH session on the controller. That is still true, but editing the files via a Web interface is another, perhaps more convenient, way of viewing and editing a set of files. Please note there is no mechanism for locking files. If you are editing a file via the Web interface and in the file system (or from two different browsers), then it is possible to lose changes. Adding a locking mechanism is left as an exercise for the reader :)).

Following is a screen capture of one Linux description/log file being edited:

![Linux report log file editing](image-url)
13.5.3 DAZL design

This section discusses some of the design of DAZL.

The main page, `index.html`, is a static HTML page which invokes one of three Common Gateway Interface (CGI) bash scripts. These scripts are run on the server in real time to generate reports. Those scripts in turn, invoke other CGI scripts where the reports have drill down links. The ability to edit description/log files in place utilizes Java Script and AJAX.

The xhr class and motivation to use AJAX was obtained from the book *Build Your Own AJAX Web Applications*, by Matthew Eernisse, published by SitePoint Ltd in 2006. See the following Web site for details on this book:

`http://www.sitepoint.com/books/ajax1/`

The Java Script file, xhr.js, is included, unchanged, with permission. It is a part of the Fleegix.js JavaScript Toolkit, copyrighted and distributed under the Apache License, v2.0. It is available from:

`http://js.fleegix.org/`

The main page is named `index.html` which goes in Apache's DocumentRoot, `/srv/www/htdocs/`, by default in SLES 9 and 10 or `/var/www/html` in RHEL 5. In addition there is the AJAX Java Script file xhr.js, obtained from the book above, and the Java Script editLog.js written to edit the text files. The CGI scripts reside in Apache's CGI directory, `/srv/www/cgi-bin/`, by default in SLES 9 and 10. They are as follow:

- `dazd.sh` Create the DASD report from the USER DISKMAP file
- `dazl.sh` Create the Linux report from the directories under `/backup/linux/`
- `dazvm.sh` Create the z/VM status report
- `getDasdInfo.sh` Create a simple page from z/VM QUERY DASD DETAILS command
- `getDesc.sh` Return the first line (the description) of the file descLog.txt

---

Figure 13-13   Editing Linux description/log files in place
getLinuxData.sh  Create a table page from commonly used Linux commands
getLog.sh  Return the entire file descLog.txt
getNetworkData.sh  Create a table page from commonly used Linux network commands
indQ.sh  Create a simple page from z/VM QUERY DASD DETAILS command
qVswitch.sh  Create a simple page from the z/VM QUERY VSWITCH command
getVMdata.sh  Create a page of z/VM data related to one Linux virtual server
runSitar.sh  Create a Sitar HTML file for a Linux virtual server on demand
getVolMap.sh  Create a page with value from USER DISKMAP for a z/VM volume
saveLog.sh  Save in-place changes to descLog.txt file for a Linux virtual server

Following is a block diagram of the DAZL system from a programming point of view:

Figure 13-14  DAZL design block diagram
Chapter 14. Backup and restore

*I never think of the future. It comes soon enough.*

--Albert Einstein

Given the opening quote, Albert Einstein would probably have not have made a good z/VM and Linux system administrator :)).

Backup and restore can be divided into two fundamental categories, each of which must be able to answer a basic question:

Incremental back up  How do I quickly get back a file that was deleted or corrupted?

Disaster recovery  How do I restore my systems if my entire data center is destroyed?

A key attribute of incremental back up is self-service. Ideally, you as the system administrator should not have to restore individual files that have been deleted or corrupted, rather, you should be easily be able to tell end users how to do it themselves. Or at least you should be able to get access to the files quickly.

A key to disaster recovery is practice. There is no *silver bullet*; disaster recovery takes planning and hard work. One rule of disaster recovery to consider is “For every backup you make, you should do one restore”. This chapter only touches on disaster recovery in the final section.

Getting back to incremental backup and restore, both your z/VM system and your Linux virtual servers must be considered. The following sections address this:

- “Incremental backup of z/VM” on page 224
- “Incremental backup of Linux virtual servers” on page 227
14.1 Incremental backup of z/VM

The following configuration files on z/VM are important:

- SYSTEM CONFIG on the MAINT CF1 minidisk
- USER DIRECT on the MAINT 2CC minidisk
- <system_ID> TCPIP and SYSTEM DTCPARMS on the TCPMAINT 198 minidisk
- TCPIP DATA on the TCPMAINT 592 minidisk
- PROFILE EXEC on the AUTOLOG1 191 minidisk

In addition, it would be helpful to backup all files on the LNXMAINT 192 minidisk. Two bash scripts are written to back up these files and other z/VM data:

backupVM.sh Backs up the files listed above. This script should not be your primary backup procedure, but it may help as a secondary method.

getVMinfo.sh Runs some basic CP commands to capture information that may be helpful.

14.1.1 The getVMinfo.sh script

There are some CP commands which also capture important information about the systems. A small script, getVMinfo.sh is supplied to issue the CP commands. This file should have been copied to the directory /usr/local/sbin/:

```bash
#!/bin/bash
...
echo "output of CP QUERY PROC:"
echo "------------------------"
vmcp QUERY PROC
echo ""
echo "output of CP QUERY FILES:"
echo "-------------------------"
vmcp QUERY FILES
echo ""
echo "output of CP QUERY NSS ALL MAP:"
echo "-------------------------------"
vmcp QUERY NSS ALL MAP
echo ""
echo "output of CP QUERY CLEVEL:"
echo "---------------------------"
vmcp QUERY CLEVEL
```

Running getVMinfo.sh script

Try running the script. You should see output similar to the following:

```
# getVMinfo.sh
output of CP QUERY PROC:
------------------------
PROCESSOR 00 MASTER
PROCESSOR 01 ALTERNATE
...

output of CP QUERY FILES:
-------------------------
FILES: 0000133 RDR, 0000018 PRT, NO PUN

output of CP QUERY NSS ALL MAP:
-------------------------------
```
Chapter 14. Backup and restore

14.1.2 The backupVM.sh script

The `backupVM.sh` script backs up important z/VM files to the `/backup/vm` directory. The first command that it runs is the above `getVMinfo.sh` script.

```
# cd /usr/local/sbin/
# cat backupVM.sh
...
# main()

cd /backup/vm              # change directory to /backup/vm where data is saved
getVMinfo.sh > zVMinfo.txt # run getVMinfo.sh to save basic VM information

# get USER DISKMAP from MAINT 191
linkAccess MAINT 191 as 1191
getFile USER.DISKMAP from 1191
relDetach 1191

# get USER DIRECT from MAINT 2CC
linkAccess MAINT 2CC as 12cc
getFile USER.DIRECT from 12cc
relDetach 12cc

# get SYSTEM CONFIG from MAINT CF1
linkAccess MAINT CF1 as 1cf1
getFile SYSTEM.CONFIG from 1cf1
relDetach 1cf1

# get TCPIP files from TCPMAINT 198
systemID=`vmcp q userid | awk '{print $3}'`
linkAccess TCPMAINT 198 as 1198
getFile $systemID.TCPIP from 1198
getFile SYSTEM.DTCPARMS from 1198
relDetach 1198
```

Important: Please note two important aspects of this script

1. It copies the USER DIRECT file which contains all of the system’s passwords (unless you are also using an external security manager). So security to the controller becomes as important as security to the MAINT user ID.

2. It backs up important z/VM data to a Linux system running under the same z/VM. If your z/VM system cannot be brought back, then this Linux system cannot be started and you cannot get the important z/VM data. Therefore, you may consider copying the output of this script to a different LPAR or different physical server.
# get TCPIP DATA from TCPMAINT 592
linkAccess TCPMAINT 592 as 1592
getFile TCPIP.DATA from 1592
relDetach 1592

# get PROFILE EXEC from AUTOLOGI 191 - rename to different FT
linkAccess AUTOLOGI 191 as 2191
CMSdevice=`cat /proc/dasd/devices | grep 2191 | awk '{print $7}'`
cmsfsclst -a -d /dev/$CMSdevice PROFILE.EXEC > PROFILE.EXECAUT1
relDetach 2191

getFiles 191               # Get all files from LNXMAINT 192 which is our 191 disk

The linkAccess() function uses the CP LINK command via vmcp to link to the specified disk read-only, then the chccwdev command to enable the new disk. The getFile() function uses the cmsfsclst command to copy a single file to the /backup/vm/ directory.

Running backupVM.sh script
Try running it now. You may choose to first remove any existing files. You should see output similar to the following:

# cd /backup/vm
# rm *
# be sure you are in the right directory when you issue this!
# backupVM.sh
Backing up USER.DISKMAP seems to have succeeded
Backing up USER.DIRECT seems to have succeeded
Backing up SYSTEM.CONFIG seems to have succeeded
Backing up LNXVM52.TCPIP seems to have succeeded
Backing up SYSTEM.DTCPARMS seems to have succeeded
Backing up TCPIP.DATA seems to have succeeded
Backing up CHPW52.XEDIT seems to have succeeded
...
Backing up SLES-S10.EXEC seems to have succeeded

You should see a number of files copied. All text files are converted to ASCII so you should be able to read them from Linux. Your output will differ:

# ls -F
CHPW52.XEDIT LINUX04.PARMFILE SLES-S10.EXEC SYSTEM.CONFIG
CPFORMAT.EXEC LINUX05.PARMFILE SLES10.EXEC SYSTEM.DTCPARMS
LABEL520.EXEC LINUX06.PARMFILE SLES10.INITRD TCPIP.DATA
LABEL520.XEDIT LINUX07.PARMFILE SLES10.KERNEL USER.DIRECT
LINUX01.PARMFILE LNXVM52.TCPIP SLES10.PARM-S10 USER.DISKMAP
LINUX02.PARMFILE PROFILE.EXEC SLES10.PARMFILE zVMinfo.txt
LINUX03.PARMFILE PROFILE.EXECAUT1 SWAPGEN.EXEC

Running backupVM.sh daily via cron
The backupVM.sh script is run once a day on the controller via the cron daemon from the script /etc/cron.daily/nightly.sh:

# tail -4 /etc/cron.daily/nightly.sh
# run sitar, backupVM.sh and backupLinux.sh nightly on the controller
sitar --format=html --outfile=/srv/www/htdocs/sitar.html
backupVM.sh # backup z/VM files to /backup/vm/
backupLinux.sh # backup Linux files to /backup/linux/

If this does successfully run each night, you will always have backed up files that are no older than one day.
14.2 Incremental backup of Linux virtual servers

Your IT shop may have an existing backup/restore solution for Linux. For example:

- Computer Associates' Brightstor Enterprise Backup
- Innovation Data Processing's FDR/UPSTREAM
- IBM's Tivoli Storage Manager (TSM)
- SecureAgent Software's SecureBackup
- Veritas' Backup Exec

For details of these and other solutions, see the Software Developer Products for Linux on IBM System z and S/390 page on the Web at:


Descriptions of these products are outside the scope of this book, and they are a much more complete solution than what is described in the sections that follow.

Some simple `cron` scripts and configuration of `rsync` follow. These will result in the /etc/ directories and other information of the virtual servers to be backed up to the controller nightly.

14.2.1 Configuring rsync on the controller

Because the master image was set up to allow the controller to copy its /etc/ directory and for ssh key-based authentication, each of the virtual servers' /etc/ directories can be copied with a single `rsync` command. For example, the following commands will copy the /etc/ directory on LINUX01 at IP address 129.40.178.121:

```
# cd /backup/LINUX01-on-129.40.178.121
# rsync -r --timeout=30 129.40.178.121:/etc .
```

To perform this backup on each of the virtual servers, a script named `backupLinux.sh` is provided. It iterates through the directories created by the clone script under /backup/linux/ of the form <userID>-on-<IPaddress>:

```
#!bin/bash
...
BACKUP="/backup/linux"
cd "$BACKUP"
for i in "*"-on-"*"  # iterate through dirs with "-on-" in their names
do
  IP_addr=${i#*-on-}  # this chops the head off and grabs the IP address
  cd "$BACKUP/$i"  # change directory to next virtual server
  rsync -r --timeout=3 $IP_addr:/etc . > /dev/null 2>&1 # use rsync to backup
  rc=$?
  if [ $rc != 0 ]; then  # error
    echo "Error: rsync returned $rc while backing up $i"
  else
    echo "Backing up $i seems to have succeeded"
  fi
done
```

This script should backup the /etc/ directory of each Linux system cloned to the appropriate directory under /backup/linux/.
14.2.2 Configuring sitar to run nightly on the virtual servers

SITAR is a tool that creates documentation describing your system. It is an acronym for System InformaTion At Runtime. In addition to backing up the /etc/ configuration files, a document describing your system might be helpful. If sitar is run once a day and the output is put in a file in /etc/, then you will have additional information about your system backed up.

This command can easily be run nightly via cron:

```
# cd /etc/cron.daily
# vi run-sitar  // add two lines
#!/bin/bash
sitar --format=html --outfile=/etc/sitar.html
```

14.3 Disaster recovery of z/VM and virtual Linux servers

In addition to incremental backups, you should do regular system backups of your z/VM system. Sections 4.10, “Backing up your z/VM system to tape” on page 61 and 4.12, “Restoring your z/VM system from tape” on page 67 just touched on this.

You may choose to do a full volume backups of all DASD in the LPAR. Then in the event of a disaster, your entire system could be restored. This is the most thorough and best way to perform backups, and is one of the areas where the System z platform excels.

You may also choose to backup a subset of the volumes (or perhaps you may choose to backup all volumes once a quarter, but backup a subset once a month). Two critical user IDs are LNMXMAINT and the controller user ID. If you have followed all steps in this book, then each of the virtual servers /etc/ file system is being backed to /backup/ file system which is the controller user ID’s 203 minidisk in this book. The master image root file system (100 minidisk) and the controller root file system (200 minidisk) are also critical. Given that these minidisks are vital to your system, you may choose to back them up more regularly or in a different fashion. You can look at the USER DISKMAP file, created on the MAINT 191 disk by the DISKMAP command, to determine which volumes to back up. You will probably have many more critical volumes than these three, but consider these especially in your hierarchy of backed up data.
References

This book refers to additional material that can be downloaded from the Internet as described below.

A.1 Related books

These publications are also relevant as further information sources:

- *Linux for zSeries and S/390 Device Drivers, Features, and Commands*, LINUX-1403
- *SUSE LINUX Enterprise Server: INSTALLATION AND ADMINISTRATION*
- *SUSE LINUX Enterprise Server: ARCHITECTURE-SPECIFIC INFORMATION*
- *SUSE LINUX Enterprise Server: START-UP GUIDE*
- *z/VM Guide for Automated Installation and Service: Version 5 Release 1.0*, GC24-6099
- *z/VM System Messages and Codes — CP: Version 5 Release 1.0*, GC24-6119
- *z/VM TCP/IP Messages and Codes: Version 5 Release 1.0*, GC24-6124
- *The Program Directory for Performance Toolkit for VM*, GI11-4800
- *z/VM CP Commands and Utilities Reference: Version 5 Release 1.0*, SC24-6081
- *z/VM CP Planning and Administration: Version 5 Release 1.0*, SC24-6083
- *z/VM Getting Started with Linux on zSeries: Version 5 Release 1.0*, SC24-6096
- *z/VM TCP/IP Planning and Customization: Version 5 Release 1.0*, SC24-6125
- *z/VM Performance Toolkit*, SC24-6136
- *Communication Controller for Linux on zSeries V1.0 Implementation Guide*, SC31-6872
- *Linux on IBM eServer zSeries and S/390: Performance Toolkit for VM*, SG24-6059
- *Linux on IBM eServer zSeries and S/390: Application Development*, SG24-6807
- *IBM Lotus Domino 6.5 for Linux on zSeries Implementation*, SG24-7021
- *Printing with Linux on zSeries Using CUPS and Samba*, REDP-3864
A.2 Online resources

These Web sites and URLs are also relevant as further information sources:

- The Linux for zSeries and S/390 portal:
  http://linuxvm.org/
- The linux-390 list server:
  http://www2.marist.edu/htbin/wlvindex?linux-390
- Linux on System z and S/390 developerWorks®:
- SUSE LINUX Enterprise Server 9 evaluation:
  http://www.novell.com/products/linuxenterpriseserver/eval.html
- z/VM publications:
- z/VM performance tips:
  http://www.vm.ibm.com/perf/tips/

A.3 Important z/VM files

z/VM differs from Linux in regard to the location and number of configuration files. In Linux, there are many configuration files and most of them are in or under the /etc/ directory. On z/VM, there are relatively few configuration files. However, they are on many different minidisks. Table 14-1 provides a summary and the location of important z/VM configuration files.

<table>
<thead>
<tr>
<th>File</th>
<th>Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTEM CONFIG</td>
<td>MAINT CF1</td>
<td>This is the operating system’s main configuration file. It defines the system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>name, the CP volumes, User volumes and other settings.</td>
</tr>
<tr>
<td>USER DIRECT</td>
<td>MAINT 2CC</td>
<td>This file defines the user directory. All user IDs or virtual machines known</td>
</tr>
<tr>
<td></td>
<td></td>
<td>to the system are defined here (assuming a directory maintenance product is</td>
</tr>
<tr>
<td></td>
<td></td>
<td>not being used).</td>
</tr>
<tr>
<td>&lt;System_ID&gt; TCPIP</td>
<td>TCPMAINT 198</td>
<td>This file defines the resources for the primary z/VM TCP/IP stack, including</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TCP/IP address, OSA resources, subnet mask and gateway. It is initially</td>
</tr>
<tr>
<td></td>
<td></td>
<td>created by the IPWIZARD tool as PROFILE TCPIP.</td>
</tr>
<tr>
<td>SYSTEM DTCPARMS</td>
<td>TCPMAINT 198</td>
<td>This file is created to define the TCP/IP stacks on the system. It is initially</td>
</tr>
<tr>
<td></td>
<td></td>
<td>created by the IPWIZARD tool.</td>
</tr>
<tr>
<td>TCPIP DATA</td>
<td>TCPMAINT 592</td>
<td>This file defines the DNS server, the domain name and some other settings.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>It is initially created by the IPWIZARD tool.</td>
</tr>
<tr>
<td>PROFILE EXEC</td>
<td>AUTOLOG1 191</td>
<td>This file is a REXX EXEC that is run when the system starts up. It is</td>
</tr>
<tr>
<td></td>
<td></td>
<td>analogous to the /etc/inittab file in Linux.</td>
</tr>
</tbody>
</table>

A.4 Cheat sheets

This section contains quick references or “cheat sheets” for the XEDIT and vi editors
A.4.1 XEDIT cheat sheet

XEDIT has line commands which are typed on the command line (===>) and prefix commands which are typed over the line numbers on the left side of the screen.

**Line Commands**

- `a` Add a line
- `a<n>` Add 'n' lines
- `c/<old>/<new>/ <n>` Search for string 'old' and replace it with 'new' for 'n' lines below the current line and 'm' times on each line. '*' can be used for 'n' and 'm'
- `/<string>` Search for 'string' from the current line
- `-<string>` Search backwards for 'string'
- `all /<string>/` Show all occurrences of 'string' and hide other lines
- `bottom` Move to the bottom of the file
- `top` Move to the top of the file
- `down <n>` Move down 'n' lines
- `up <n>` Move up 'n' lines
- `file` Save the current file and exit XEDIT
- `ffile` Save the current file and exit but don’t warn of overwrite
- `save` Save the current file but don’t exit
- `quit` Exit XEDIT if no changes have been made
- `qquit` Exit XEDIT even if changes have not been saved
- `left <n>` Shift ‘n’ characters to the left
- `right <n>` Shift ‘n’ characters to the right
- `get <file>` Copy file and insert past the current line
- `:<n>` Move to line 'n'
- `?` Display last command
- `=` Execute last command
- `x <file>` Edit 'file' and put it into the XEDIT “ring”
- `x` Move to the next file in the ring

**Prefix Commands**

- `a` Add one line
- `a<n>` Add 'n' lines
- `c` Copies one line
- `cc` Copies a block of lines
- `d` Deletes one line
- `dd` Deletes a block of lines
- `f` Line after which a copy (c) or a move (m) is to be inserted
- `p` Line before which a copy (c) or a move (m) is to be inserted
- `i` Insert a line
- `i<n>` Insert 'n' lines
- `m` Move one line
- `mm` Move a block of lines
- `"` Replicate a line
- `"<n>` Replicate a line 'n' times
- `""` Replicate a block of lines

A.4.2 vi cheat sheet

Following is a small subset of vi commands, but those most commonly used. The vi editor has three modes:

1. Input mode - the **Insert** key, i, o (add a line below), O (add a line above) and other commands put you in this mode. When you are in this mode you will see the text `--INSERT--` in the last line.

2. Command mode - ‘Esc’ gets you out of input mode and into command mode
i brings you back to input mode
dd deletes a line and puts it in the buffer
<n>dd delete <n> lines
x delete a character
dw delete a word
p add the buffer past the current location
P add the buffer before the current location
o add a line and go into insert mode
/ string - search for string
n do the last command again (this can be powerful)
jkl; cursor movement
A add text at the end of the line
<n>G go to line <nn>
G go to the last line in the file
yy yank a line (copy into buffer)
<n>yy yank n lines

3. Command line mode - pressing the colon : key brings you to this mode
:wq save (write & quit)
:q! quit and discard changes
:<nn> go to line number <nn>
:r <file> read <file> into the current file
:1,$s/old/new/g globally replace <old> with <new>
:help give help
Common source code

This section lists source code for z/VM and Linux that is common regardless of Linux distribution. The common code is in the following categories:

- z/VM REXX EXECs and XEDIT macros
- Bash scripts on the master image
- Bash scripts on the controller

B.1 z/VM REXX EXECs and XEDIT macros

This section lists z/VM REXX EXECs and XEDIT macros.

B.1.1 The CPFORMAT EXEC

Following is the code for the EXEC that formats multiple disks via CPFMTXA (described in section 4.6.1, “Formatting the paging volumes” on page 47):

```/*+------------------------------------------------------------------*/
/*| EXEC: CPFORMAT - wrapper around CPFMTXA to format many DASD      |*/
/*|  retVal: 0 - success                                             |*/
/*|          1 - help was asked for or given                         |*/
/*|          2 - user is not sure                                    |*/
/*|          3 - DASD (minidisk) range is not valid                  |*/
/*|          4 - at least one DASD (minidisk) is reserved to MAINT   |*/
/*+------------------------------------------------------------------*/

THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS
firstChar = 'M' /* change this for an LPAR ID other than 'M' */
parse upper arg dasds "AS " type
if ((dasds = '') | (dasds = '?')) then call help
labelPrefix = getLabelPrefix(firstChar type)
umDasd = parseDasd(dasds)
answer = areYouSure(type)
if (answer = 'Y') then /* the user is sure */
do
  retVal = doFormat(labelPrefix numDasd type)
call doReport
end
else
  retVal = 2
exit retVal

help: procedure
/*+------------------------------------------------------------------*/
parse source .. fn .
say ''
say 'Synopsis:'
say ''
say ' Format one or a range of DASD as page, perm, spool or temp disk space'
say ' The label written to each DASD is V<t><xxxx> where:'
say '   <t> is type - P (page), M (perm), S (spool) or T (Temp disk)'
say '   <xxxx> is the 4 digit address'
say ''
say 'Syntax is:'
say ''
say ' CPFORMAT -rdev--------.--AS---+-PERM-+---------><
                 | <---------------< |       '-SPOL-''
                 '-rdev1-rdev2-------'
exit 1

areYouSure: procedure
/*| Show minidisks, ask are you sure                                 |
/*|  parm 1: type - PERM, PAGE, or SPOL                              |
/*|  retVal: firstChar - LPAR identifier, 'V' by default             |
/*+------------------------------------------------------------------*/
arg type
say ''
say 'WARNING - this will destroy data!'
say 'ARE YOU SURE you want to format the DASD as' type 'space (y/n),'
parse upper pull answer
return substr(answer, 1, 1) /* from areYouSure */

getLabelPrefix: procedure
/*| Return first two characters of label                            |
/*|  parm 1: firstChar - LPAR identifier, 'V' by default             |
/*|  retVal: the two character label prefix                          |
/*+------------------------------------------------------------------*/
arg firstChar type
select
  when (type = PERM) then
labelPrefix = firstChar||'M' /* for VM Minidisk */
when (type = PAGE) then
  labelPrefix = firstChar||'P' /* for VM Page */
when (type = SPOL) then
  labelPrefix = firstChar||'S' /* for VM Spool */
otherwise
  do
    say 'Error: "AS" must be present and type must be PERM, PAGE or SPOL'
    call help
  end /* otherwise */
end /* select */
return labelPrefix /* from getLabelPrefix */

 parseDasd: procedure expose dasdList.
 /*| parse all dasd into an array verifying all are attached |*/
 /*| parm 1: dasds - the list of dasd passed in          |*/
 /*| retVal: number of DASD in dasdList                  |*/
 /*|-------------------------------------------------------------------*/
arg dasds
numDasd = 0
say ''
say 'Format the following DASD:'
do while (dasds <> '')
  parse upper var dasds dasd dasds
  dashPos = pos('-', dasd)
  if (dashPos = 0) then /* there is just one DASD */
    do
      numDasd = numDasd + 1
      dasdList.numDasd = dasd
      'CP Q MDISK' dasdList.numDasd 'LOCATION'
      if (rc <> 0) then
        do
          say 'Return code from Q MDISK =' rc
          say 'Are all DASD ATTrach?'
          exit 3
        end
        call checkReserved(dasdList.numDasd)
      end /* do */
    end /* do */
  else /* process the range of DASD */
    do
      startRange = substr(dasd, 1, dashPos - 1)
      endRange = substr(dasd, dashPos + 1, length(dasd) - dashPos)
      do i = x2d(startRange) to x2d(endRange)
        numDasd = numDasd + 1
        dasdList.numDasd = d2x(i)
        'CP Q MDISK' dasdList.numDasd 'LOCATION'
        if (rc <> 0) then
          do
            say 'Return code from Q MDISK =' rc
            exit 3
          end
          call checkReserved(dasdList.numDasd)
        end /* do i */
      end /* do */
    end /* do while */
return numDasd /* from parseDasd */

doFormat: procedure expose dasdList.
/* Format all DASD specified using CPFMTXA                        */
/* parm 1: labelPrefix - the two character label prefix          */
/* parm 2: numDasd - number of DASD in the array dasdList        */
/* parm 3: type - the type of DASD format                        */
/* retVal: 0 = success                                          */
/*------------------------------------------------------------------*/
arg labelPrefix numDasd type
'CP TERM MORE 1 1'
do i = 1 to numDasd
   label = getLabel(labelPrefix dasdList.i)
   call formatOne(dasdList.i type label)
end /* do i = */
'CP TERM MORE 50 10'
return 0 /* from doFormat */

/*------------------------------------------------------------------*/
checkReserved: procedure
/* Try copying an already formatted DASD then relabelling it     */
/* parm 1: source                                               */
/* parm 2: target                                              */
/* parm 3: label                                               */
/*------------------------------------------------------------------*/
arg dasd
/* create a list of reserved dasd - this is somewhat hokey to be sure
   but it's better to be hokey than to format system minidisks! */
   resvd1 = "0122 0123 0124 0125 0190 0191 0193 0194 019D 019E 0201 02A2"
   resvd2 = "02A4 02A6 02C2 02C4 02CC 02DD 032D 03A2 03A4 03A6 03B2 03C2"
   resvd3 = "03C4 03D2 0400 0402 0405 040A 040B 040F 0490 049B 049E 0442"
   resvd4 = "04A6 04B2 04C2 04D2 0500 051D 05A2 05A4 05A6 05B2 05C2"
   resvd5 = "05C4 05D2 05E5 05E6 05F6 05F8 05FA 05FB 05FC 05FD 05FE 0602"
   reserved = resvd1 resvd2 resvd3 resvd4 resvd5 resvd6
if (index(reserved, dasd) <> 0) then /* MAINT minidisk - ABORT! */
do
    say 'Minidisk' dasd 'is a reserved MAINT minidisk'
    say 'This must be formatted manually using a different vaddr'
    exit 4
end /* if dasd is reserved */
return /* from checkReserved */

/*------------------------------------------------------------------*/
doReport: procedure expose dasds
/* Report on the newly labelled DASD                             */
/* retVal: 0 = success                                           */
/*------------------------------------------------------------------*/
'DETACH' dasds
'ATTACH' dasds '*'
say ''
say 'DASD status after:'
'CP Q MDISK' dasds 'LOCATION'
return 0 /* from doReport */

/*------------------------------------------------------------------*/
formatOne: procedure
/* Format a DASD via DDR                                         */
/* parm 1: disk - the vaddr to be formatted                      */
/* parm 2: type - PAGE, SPOL or PERM                            */
/* parm 3: label - the six character label                      */
/*------------------------------------------------------------------*/
arg disk type label
queue 'FORMAT'
queue disk
queue '0 END'
queue label
queue 'YES'
queue type '0 END'
'CPFMTXA'
return /* from formatOne */

getLabel: procedure
/*| Compose the six character label of a minidisk          |*/
/*| parm 1: labelPrefix - first two characters of label   |*/
/*| parm 2: disk - vaddr of length 1, 2, 3 or 4          |*/
/*| return: the 6 character label                        |*/
/*+------------------------------------------------------------------+*/
arg labelPrefix disk
diskLen = length(disk)
select
when (diskLen = 1) then /* insert 3 zeros */
    label = labelPrefix||'000'||disk
when (diskLen = 2) then /* insert 2 zeros */
    label = labelPrefix||'00'||disk
when (diskLen = 3) then /* insert a zero */
    label = labelPrefix||'0'||disk
otherwise /* it must be length 4 or query would have failed */
    label = labelPrefix||disk
end /* select */
return label /* from getLabel */

B.1.2 The CHPW52 XEDIT macro

Following is the code for the XEDIT macro that changes all passwords in the z/VM 5.2 USER DIRECT file:

/* CHPW52 XEDIT - change all passwords in z/VM 5.2 USER DIRECT file */

parse arg fn ft fm '(' options ')' newPass .
if (length(newPass) > 8) then
do
    say "Error: new password must be 8 characters or fewer"
    exit
end
say ''
say 'Changing all passwords to:' newPass
say ''

/* set some values */
'command set stay on'
'command set num on'
'command set nulls on'
'command set serial off'
'command set cmdline bottom'
'command set serial on 3'
'command set serial off'
'command set scale off'
'command set case m i'
'command set pre off'
'command set v 1 80'

/* change user ID passwords */
'command c/CMS1 CMS1/CMS1' newPass'/*'
'command c/LGLOPR LGLOPR/LGLOPR' newPass'/*'
'command c/MAINT MAINT/MAINT' newPass'/*'
'command c/CMSBATCH CMSBATCH/CMSBATCH' newPass'/*'
'command c/AVSMV AVSMV/AVSMV' newPass'/*'
'command c/TSAFVM TSAFVM/TSAFVM' newPass'/*'
'command c/VMSERV1 VMSERV1/VMSERV1' newPass'/*'
'command c/VMSERV2 VMSERV2/VMSERV2' newPass'/*'
'command c/VMSERV3 VMSERV3/VMSERV3' newPass'/*'
'command c/GCS GCS/GCS' newPass'/*'
'command c/GCSXA GCSXA/GCSXA' newPass'/*'
'command c/SYSMAINT SYSMAINT/SYSMAINT' newPass'/*'
'command c/OPERATOR OPERATOR/OPERATOR' newPass'/*'
'command c/OPI OPI/OPI' newPass'/*'
'command c/EREP EREP/EREP' newPass'/*'
'command c/OPERATNS OPERATNS/OPERATNS' newPass'/*'
'command c/AUTOLOGI AUTOLOGI/AUTOLOGI' newPass'/*'
'command c/DISKACNT DISKACNT/DISKACNT' newPass'/*'
'command c/OPERSYM OPERSYM/OPERSYM' newPass'/*'
'command c/VMUTIL VMUTIL/VMUTIL' newPass'/*'
'command c/SYSDUMP1 SYSDUMP1/SYSDUMP1' newPass'/*'
'command c/5684042J 5684042J/5684042J' newPass'/*'
'command c/4OSASF40 4OSASF40/4OSASF40' newPass'/*'
'command c/OSASF OSASF/OSASF' newPass'/*'
'command c/OSAMAINT OSAMAINT/OSAMAINT' newPass'/*'
'command c/OSADMINI OSADMINI/OSADMINI' newPass'/*'
'command c/OSADMIN2 OSADMIN2/OSADMIN2' newPass'/*'
'command c/OSADMIN3 OSADMIN3/OSADMIN3' newPass'/*'
'command c/P684096K P684096K/P684096K' newPass'/*'
'command c/RSCS RSCS/RSCS' newPass'/*'
'command c/RSCDNS RSCDNS/RSCDNS' newPass'/*'
'command c/TCPMAINT TCPMAINT/TCPMAINT' newPass'/*'
'command c/5VMTCP0 5VMTCP0/5VMTCP0' newPass'/*'
'command c/TCPPIP TCPPIP/TCPPIP' newPass'/*'
'command c/IMAP IMAP/IMAP' newPass'/*'
'command c/FTPSERVE FTPSERVE/FTPSERVE' newPass'/*'
'command c/SMTP SMTP/SMTP' newPass'/*'
'command c/NAMESRV NAMESRV/NAMESRV' newPass'/*'
'command c/REXEC REXEC/REXEC' newPass'/*'
'command c/RXAGENT1 RXAGENT1/RXAGENT1' newPass'/*'
'command c/X251PI X251PI/X251PI' newPass'/*'
'command c/PORTMAP PORTMAP/PORTMAP' newPass'/*'
'command c/NDBPMGR NDBPMGR/NDBPMGR' newPass'/*'

'command c/NDBSRV01 NDBSRV01/NDBSRV01' newPass'/*
'command c/SNMPQE SNMPQE/SNMPQE' newPass'/*
'command c/SNMPD SNMPD/SNMPD' newPass'/*
'command c/ROUTED ROUTED/ROUTED' newPass'/*
'command c/LPSERVE LPSERVE/LPSERVE' newPass'/*
'command c/SNALNKA SNALNKA/SNALNKA' newPass'/*
'command c/VNMFS VNMFS/VNMFS' newPass'/*
'command c/VMKERB VMKERB/VMKERB' newPass'/*
'command c/ADMSERV ADMSERV/ADMSERV' newPass'/*
'command c/UFTD UFTD/UFTD' newPass'/*
'command c/BOOTPD BOOTPD/BOOTPD' newPass'/*
'command c/TFTPD TFTPD/TFTPD' newPass'/*
'command c/DHCPD DHCPD/DHCPD' newPass'/*
'command c/MROUTE MROUTE/MROUTE' newPass'/*
'command c/SSL SERV SSL SERV/SSL SERV' newPass'/*
'command c/MONWRITE MONWRITE/MONWRITE' newPass'/*
'command c/5VMDIR10 5VMDIR10/5VMDIR10' newPass'/*
'command c/BLDNUC BLDNUC/BLDNUC' newPass'/*
'command c/AUDITOR AUDITOR/AUDITOR' newPass'/*
'command c/SYSMON SYSMON/SYSMON' newPass'/*
'command c/5767002P 5767002P/5767002P' newPass'/*
'command c/RACFVM RACFVM/RACFVM' newPass'/*
'command c/RACFSMF RACFSMF/RACFSMF' newPass'/*
'command c/RACMAINT RACMAINT/RACMAINT' newPass'/*
'command c/AUTOLOG2 AUTOLOG2/AUTOLOG2' newPass'/*
'command c/IBMUSER IBMUSER/IBMUSER' newPass'/*
'command c/SYSA DM SYSA DM/SYSA DM' newPass'/*
'command c/BLDRACF BLDRACF/BLDRACF' newPass'/*
'command c/5VMP TK20 5VMP TK20/5VMP TK20' newPass'/*
'command c/PERFSVM PERFSVM/PERFSVM' newPass'/*
'command c/CBDIODSP CBDIODSP/CBDIODSP' newPass'/*
'command c/VSMSERVE VSMSERVE/VSMSERVE' newPass'/*
'command c/WIMAPAUTH WIMAPAUTH/WIMAPAUTH' newPass'/*
'command c/RCAT ALOG WCATALOG/'newPass newPass'/*
'command c/RCONTROL WCONTROL/'newPass newPass'/*
'command c/READ WRITE MULTIPLE/'newPass newPass newPass'/*
'command c/RFTPSERV WFTPSERV MFTPSERV/'newPass newPass newPass'/*'
'command c/RGCS WGCS MGCS/'newPass newPass newPass'/*'
'command c/RIMAP WIMAP MIMAP/'newPass newPass newPass'/*'
'command c/RLOG1 WLOG1/'newPass newPass newPass'/*'
'command c/RLOG2 WLOG2/'newPass newPass newPass'/*'
'command c/RLPSPRVE WLPSERVE MLPSPRVE/'newPass newPass newPass'/*'
'command c/RMAINT WMMAINT MMMAINT/'newPass newPass newPass'/*'
'command c/RMPROUTE WMMPROUTE MMPROUTE/'newPass newPass newPass'/*'
'command c/RNAMESRV WNAMESRV MNAMESRV/'newPass newPass newPass'/*'
'command c/RDBSMGR WDBSMGR MDBSMGR/'newPass newPass newPass'/*'
'command c/RNDBSRV0 WNDBSRV0 MNDBSRV0/'newPass newPass newPass'/*'
'command c/RPORTMAP WPRTMAP MPRTMAP/'newPass newPass newPass'/*'
'command c/RREXECD WREXECD MRREXECD/'newPass newPass newPass'/*'
'command c/RROUTE WROUTE MROUTE/'newPass newPass newPass'/*'
'command c/RSNMPD WSNMPD MSNMPD/'newPass newPass newPass'/*'
'command c/RNSMPQ E RNSMPQ M RNSMPQ/'newPass newPass newPass'/*'
'command c/RNSSERV WNSSERV MNSSERV/'newPass newPass newPass'/*'
'command c/RPORTMAP WPRTMAP MPRTMAP/'newPass newPass newPass'/*'
'command c/RRTFTPD WRTFTPD MRTFTPD/'newPass newPass newPass'/*'
'command c/RRTFTPD WRTFTPD MRTFTPD/'newPass newPass newPass'/*'
'command c/RRTSBNP WBRTSBNP MBRTSBNP/'newPass newPass newPass'/*'
'command c/RUFTD WUFTD MUFTD/'newPass newPass newPass'/*'
'command c/RVMKKB WVMKKB MVMKKB/'newPass newPass newPass'/*'
'command c/RVMNS B WVMNS B MVMNS B/'newPass newPass newPass'/*'
'command c/RCRRLOG2 WCRRLQ2 MCRRLQ2/'newPass newPass newPass'/*'
'command c/MR READ/"MR newPass"/*'

B.1.3 The LABEL520 EXEC

Following is the code for the EXEC that changes the system labels of a z/VM 5.2 system:

/*+------------------------------------------------------------------+*/
/*| EXEC: LABEL520  wrapper around CPFMTXA to LABEL and ALLOC DASD   |*/
/*|  retVal: 0 - success                                             |*/
/*|          1 - help was asked for or given                         |*/
/*|          2 - user is not sure                                    |*/
/*|          3 - DASD (minidisk) range is not valid                  |*/
/*|          4 - at least one DASD (minidisk) is reserved to MAINT   |*/
/*+------------------------------------------------------------------+*/

THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR
CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT
LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT,
MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED
AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS
GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES
parse upper arg res spl pag w01 w02.
if (w02 = '') then call help
/* Construct the two character label prefix */
firstChar = 'M' /* change this for an LPAR ID other than 'M' */
labelPrefix = firstChar'V'
/* Construct the 5 labels */
resLabel = getLabel(labelPrefix res)
splLabel = getLabel(labelPrefix spl)
pagLabel = getLabel(labelPrefix pag)
w01Label = getLabel(labelPrefix w01)
w02Label = getLabel(labelPrefix w02)
/* Ask "Are you sure?" */
say 'The volumes are:
   CP Q' res spl pag w01 w02
say ''
say 'The system volume labels will become:
   say resLabel splLabel pagLabel w01Label w02Label
say ''
say 'ARE YOU SURE you want to relabel the DASD (y/n)?'
parse upper pull answer
ansFirstChar = substr(answer, 1, 1)
if (ansFirstChar ^= 'Y') then exit 2
/* Label the 4 volumes: RES is 123, W01 is 124, W02 is 125, SPL is 122 */
'CP TERM MORE 1 1'
'CPFMTXA 123' resLabel 'LABEL'
'CPFMTXA 124' w01Label 'LABEL'
'CPFMTXA 125' w02Label 'LABEL'
'CPFMTXA 122' splLabel 'LABEL'
/* LINK the 520PAG volume which is $PAGE$ A03, label it, DETACH it */
'CP LINK $PAGE$ A03 A03 MR'
'CPFMTXA A03' pagLabel 'LABEL'
'CP DET A03'
'CP TERM MORE 50 10'
exit
/*+------------------------------------------------------------------+*/
help: procedure
/*+------------------------------------------------------------------+*/
parse source .. fn .
say **
say "Synopsis:"
say **
say "Relabel the five system volumes (520RES, 520W01, ...) to VV<xxxx>"
say " where <xxxx> is the 4 digit address"
say **
say "Syntax is:"
say **
say " >>>-LABEL520--res--spl--pag--w01--w02------------------------><"
say **
say " where res, spl, pag, w01 and w02 are 4 digit virtual addresses"
say " of the volumes that z/VM 5.2 is installed onto"
say **
exit 1
getLabel: procedure
/* Compose the six character label of a minidisk */
/* parm 1: labelPrefix - first two characters of label */
/* parm 2: disk - vaddr of length 1, 2, 3 or 4 */
/* return: the 6 character label */

arg labelPrefix disk
if (DATATYPE(disk, 'X') = 0) then
  do
    say "Error:" disk "is not a hexadecimal number"
    call help
  end
end

diskLen = length(disk)
select
  when (diskLen = 1) then /* insert 3 zeros */
    label = labelPrefix||'000'||disk
  when (diskLen = 2) then /* insert 2 zeros */
    label = labelPrefix||'00'||disk
  when (diskLen = 3) then /* insert a zero */
    label = labelPrefix||'0'||disk
  otherwise /* it must be length 4 or query would have failed */
    label = labelPrefix||disk
end /* select */

return label /* from getLabel */
B.2 Linux source code

This section lists source code that is common regardless of Linux distribution.

B.2.1 Bash scripts on the controller

Following is the source code for the bash scripts that run on the controller.

B.2.1.1 The nightly.sh script

Following is the content of the bash script /etc/cron.daily/nightly.sh. It runs scripts to backup important z/VM and Linux data.

```bash
#!/bin/bash
#
# nightly.sh - do the following on the controller each night
# run backupVM.sh to backup VM files
# run backupLinux.sh to backup /etc/ directories on Linux
#
/sbin/backupVM.sh
/sbin/backupLinux.sh
```

B.2.1.2 The backupVM.sh script

Following is the content of the bash script /etc/cron.daily/nightly.sh. It does THIS.

```bash
#!/bin/bash
#
# backupVM.sh - do a backup of z/VM files by linking to important disks:
```
# MAINT 191 - USER DISKMAP
# MAINT 2CC - USER DIRECT
# MAINT CF1 - SYSTEM CONFIG
# TCMPMAINT 198 - <systemID> TCPIP, SYSTEM DTCPPARMs
# TCMPMAINT 592 - TCPIP DATA
# AUTOLOG1 191 - PROFILE EXEC
# LNKMMAINT 192 - * PARMFILE, * PARM-S10, SLES* *
# The system that this is run on should have the LNKNOPAS directory option so
# all the above minidisks can be linked via vmcp with no password necessary
#
#+--------------------------------------------------------------------------+
# function cpCmd()
# echo a CP command and invoke it via the vmcp module/command
# Arg 1-n: the command to issue
# Return: the command's return code
#+--------------------------------------------------------------------------+
#parse output to get return code: awk -F# splits line at '#' with rc at end
output=`vmcp $@ 2>&1`
ret_val=0
ret_val=`echo $output | grep "Error: non-zero CP response" | awk -F# '{print $2}'`
return $ret_val
#
#+--------------------------------------------------------------------------+
# function linkAccess()
# link and access a minidisk
# Arg 1: the source disk user ID
# Arg 2: the source disk virtual device number
# Arg 3: placeholder for code readability
# Arg 4: the virtual device number to link the minidisk as
#+--------------------------------------------------------------------------+
{
  userID=$1
  vaddr=$2
  linkVaddr=$4
cpCmd QUERY VIRTUAL $linkVaddr
queryRC=$?
if [ $queryRC = 40 ]; then # linkVaddr is free
  cpCmd LINK $userID $vaddr $linkVaddr RR
  rc=$?
  if [ $rc != 0 ]; then # can't link - punt
    echo "Error: unable to link $userID $vaddr read-only"
    echo "LINK $userID $vaddr $linkVaddr RR returned $rc"
    exit
  fi
else # linkVaddr is not free
  echo "Error: Virtual address $linkVaddr is not free?"
  echo "QUERY VIRTUAL $linkVaddr returned $queryRC"
fi
# activate new disk then sleep 1 second for CP to catch up
chccwdev -e $linkVaddr > /dev/null 2>&1
sleep 1
#
#+--------------------------------------------------------------------------+
# function getFile()
# get a file on a CMS disk
# Arg 1: the file or file pattern to get
# Arg 2: placeholder for code readability

244 z/VM and Linux on IBM System z: The Virtualization Cookbook
# Arg 3: the virtual device number to detach
{  
  file=$1  
vaddr=$3  
CMSdevice=`cat /proc/dasd/devices | grep $vaddr | awk '{print $7}'``
cmsfscat -a -d /dev/$CMSdevice $file > $file  
rc=$?  
if [ $rc != 0 ]; then  
echo "Error: cmsfscat returned $rc backing up $file"  
else  
echo "Backing up $file seems to have succeeded"  
fi  
}

function getFiles()  
# get all files on a CMS disk
# Arg 1: the virtual device number to copy files from
{  
vaddr=$1  
CMSdevice=`cat /proc/dasd/devices | grep $vaddr | awk '{print $7}'``
let i=0  
cmsfslst -d /dev/$CMSdevice | while read oneline  
do # first record is headers, second is "DIRECTOR", third is "ALLOCMAP"
  if ((i > 2)); then # we are into the CMS files
    fn=`echo $oneline | awk '{print $1}'``
    ft=`echo $oneline | awk '{print $2}'``
    cmsfscat -a -d /dev/$CMSdevice $fn.$ft > $fn.$ft  
    rc=$?  
    if [ $rc != 0 ]; then # error
      echo "Error: cmsfscat returned $rc backing up $fn.$ft"  
    else  
      echo "Backing up $fn.$ft seems to have succeeded"  
    fi  
    fi  
  let i=$i+1  
done  
}

function relDetach()  
# link and access a minidisk
# Arg 1: the virtual device number to detach
# Return: the command's return code
{  
  chccwdev -d $1 > /dev/null 2>&1  
  cpCmd DET $1  
}

# main()  

cd /backup/vm  

# run the getVMinfo script to save some basic VM information
getVMinfo > zVMinfo.txt  

# get USER DISKMAP from MAINT 191
linkAccess MAINT 191 as 1191
getFile USER.DISKMAP from 1191
relDetach 1191

# get USER DIRECT from MAINT 2CC
linkAccess MAINT 2CC as 12cc
getFile USER.DIRECT from 12cc
relDetach 12cc

# get SYSTEM CONFIG from MAINT CF1
linkAccess MAINT CF1 as 1cf1
getFile SYSTEM.CONFIG from 1cf1
relDetach 1cf1

# get TCPIP files from TCPMAINT 198
systemID=`vmcp q userid | awk '{print $3}'`
linkAccess TCPMAINT 198 as 1198
getFile $systemID.TCPIP from 1198
getFile SYSTEM.DTCPARMS from 1198
relDetach 1198

# get TCPIP DATA from TCPMAINT 592
linkAccess TCPMAINT 592 as 1592
getFile TCPIP.DATA from 1592
relDetach 1592

# get PROFILE EXEC from AUTOLOG1 191 - rename to different FT
linkAccess AUTOLOG1 191 as 2191
CMSdevice=`cat /proc/dasd/devices | grep 2191 | awk '{print $7}'`
cmsfscat -a -d /dev/$CMSdevice PROFILE.EXEC > PROFILE.EXECAUT1
relDetach 2191

# Get all files from LNXMAINT 192 which is our 191 disk
getFiles 191

B.2.1.3 The backupLinux.sh script
Following is the content of the bash script /sbin/backupLinux.sh. It loops through the directories under /backup/linux/ and updates each /etc/ directory via rsync:

#!/bin/bash

# For details on this code see the books "z/VM and Linux on IBM System z: The Virtualization Cookbook for <x>" on the Web at: http://linuxvm.org/present
# where <x> is "SLES 10" or "RHEL 5"

BACKUP="/backup/linux"
cd $BACKUP
for i in *-on-*            # iterate through dirs with "-on-" in their names
doi
IP_addr=${i#LINUX*-on-}  # this chops the head off and grabs the IP address
cd $BACKUP/$i             # change directory to next virtual server
rsync -r --timeout=3 $IP_addr:/etc . > /dev/null 2>&1 # use rsync to backup
rc=$?
if [ $rc != 0 ]; then      # error
    echo "Error: rsync returned $rc while backing up $i"
else
    echo "Backing up $i seems to have succeeded"
fi
done
Source code specific to Linux

C.1 Obtaining and using the Web material

The PDF of this book is on the Internet at:


The files associated with this book are in a gzipped tar file at:

http://linuxvm.org/present/misc/virt-cookbook-S10.tgz

Download the tar file to your NFS server and use it as is described in section 6.1. After untarring the file, you will have a directory named virt-cookbook-3/ for SLES 9, or virt-cookbook-S10/ for SLES 10. Under that directory are the following file and directories:

- README.txt: The main README file
- linux-controller/: Files used on Linux controller
- linux-master/: Files used on Linux master image which will become cloned servers
- nfs-server/: Files used on the temporary NFS server
- vm/: Files used on z/VM

This section lists the source code specific to Linux distributions.

C.2 The clone.sh script

This section lists the clone.sh script:

```bash
#!/bin/bash
#
# clone.sh <LinuxUserID> - clone a Linux server running under z/VM
#
# For details on how this script works see the book:
# "z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES 10"
#
# THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS
# OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY
```
# WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY
# OR FITNESS FOR A PARTICULAR PURPOSE.
# NEITHER RECIPENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY
# DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY
# OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
# NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR
# DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED
# HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES
# ----------------------------------------------------------------------------

#+--------------------------------------------------------------------------+
# function help()
# give help
#+--------------------------------------------------------------------------+
{
  echo "Usage: clone [options] target_linux_ID"
  echo "  Clone Linux system from minidisk $sdisk to target_linux_ID"
  echo "  options:"
  echo "    -v or --verbose: verbose"
  echo "    -w or --was: enable WebSphere server"
  echo "    -d or --db2: enable DB2 server"
  echo "    -m or --mqs: enable MQ Series server"
  exit 1
}

#+--------------------------------------------------------------------------+
# function process_arguments()
# give help
#+--------------------------------------------------------------------------+
{
  verbose="off"
  if [ $# = 0 ]; then # user did not pass an argument
    echo "whoops: at least one argument is needed"
    echo ""
    help
    fi
  for arg in $*
  do
    case $arg in
    -v|--verbose)
      verbose="on"
      ;;
    -w|--was)
      was="yes"
      ;;
    -d|--db2)
      db2="yes"
      ;;
    -m|--mqs)
      mqs="yes"
      ;;
    *)
      clone_id=`echo $arg | tr '[a-z]' '[A-Z]'` # fold target user ID to upper
      case
      ;;
      esac
      done
  esac
  if [ $clone_id = "none" ]; then # new linux user ID was not passed
Appendix C. Source code specific to Linux

```bash
echo "Error: new Linux user ID must be an argument"
echo ""
help
fi

# Function to check Linux user ID
check_target_id()
#
#
{ cp_cmd QUERY $clone_id
  case $? in
  0) # user ID is logged on or disconnected
      echo "$clone_id user ID must be logged off"
      exit 2
  ;;
  3) # user ID does not exist
      echo "$clone_id user ID does not exist"
      exit 3
  ;;
  45) # user ID is logged off - this is correct
  ;;
  *) # unexpected
      echo "$clone_id user ID must exist and be logged off"
      exit 4
  esac
}

function prepare_ipaddr()
#
#
{ new_ipaddr=`echo $1 | sed -e 's:\.:\\.:g'`
}

function prepare_vaddr()
#
#

```
# to make it 4 digits
# Arg 1: The vaddr to be modified
# Return:
# The new value is written to the global variable new_vaddr
#
{
    new_vaddr=`echo $1 | tr \[A-Z\] \[a-z\]`  # fold to lower case
    let leadingZeros=4-${#1}                  # determine number of zeros to add
    let i=0
    while [ $i -lt $leadingZeros ]; do
        new_vaddr="0$new_vaddr"
        i=$[i+1]
    done
}

function copy_disk()
#
# Try to use z/VM FLASHCOPY to copy one disk to another.
# Arg 1: Source minidisk vaddr
# Arg 2: the word "to"
# Arg 3: Target user ID
# Arg 4: Target virtual address
# Return code
# 0: success with FLASHCOPY
# 1: success with dasdfmt and dd
# 2: Target vaddr already in use?
# 3: CP LINK failed
# 4: unable to enable target minidisk
# 5: can't find source disk in /dev/dasd/devices
# 6: can't find target disk in /dev/dasd/devices
# 7: dasdfmt failed
# 8: dd failed
#
{
    ret_val=0
    source_mdisk=$1
    target_userid=$3
    target_vaddr=$4
    cp_cmd QUERY VIRTUAL $target_vaddr
    rc=$?
    if [ $rc != 40 ]; then
        echo "Error: CP QUERY VIRTUAL $target_vaddr failed with $rc"
        echo "The target device address is already in use"
        return 2
    fi
    cp_cmd LINK $target_userid $source_mdisk $target_vaddr MR
    rc=$?
    if [ $rc != 0 ]; then # LINK failed
        echo "Error: CP LINK $target_userid $source_mdisk $target_vaddr failed with $rc"
        return 3
    fi
    cp_cmd FLASHCOPY $source_mdisk 0 END $target_vaddr 0 END
    rc=$?
    if [ $rc != 0 ]; then # FLASHCOPY failed
        echo "FLASHCOPY $source_mdisk $target_vaddr failed with $rc"
        echo "Falling back to dasdfmt and dd copy"
        # enable target disk
        sleep 1
        chccwdev -e $target_vaddr
        rc=$?
if [ $rc != 0 ]; then # unable to enable target disk
  echo "Error: unable to enable $target_disk, rc from chccwdev = $rc"
  cp_cmd DETACH $target_vaddr
  return 4
fi

# get device name of source disk
source_dev=`cat /proc/dasd/devices | grep "$source_mdisk(ECKD)" | awk '{ print $7 }'`
if [ "$source_dev" = "" ]; then
cat /proc/dasd/devices
  echo "Error: can't find $source_mdisk(ECKD) in /proc/dasd/devices"
  chccwdev -d $target_vaddr # clean up
  cp_cmd DETACH $target_vaddr
  return 5
fi

# get device name of target disk
target_dev=`cat /proc/dasd/devices | grep "$target_vaddr(ECKD)" | awk '{ print $7 }'`
if [ "$target_dev" = "" ]; then
cat /proc/dasd/devices
  echo "Error: can't find $target_vaddr(ECKD) in /proc/dasd/devices"
  chccwdev -d $target_vaddr # clean up
  cp_cmd DETACH $target_vaddr
  return 6
fi

dasdfmt target disk
  echo "Invoking command: dasdfmt -b 4096 -y -f /dev/$target_dev"
  dasdfmt -b 4096 -y -f /dev/$target_dev
  rc=$?
  if [ $rc != 0 ]; then # dasdfmt failed
    echo "Error: dasdfmt -b 4096 -y -f /dev/$target_dev failed with $rc"
    chccwdev -d $target_vaddr # clean up
    cp_cmd DETACH $target_vaddr
    return 7
  fi

  # copy from source disk to target via dd
  echo "Invoking command: dd bs=4096 if=/dev/$source_dev of=/dev/$target_dev"
  dd bs=4096 if=/dev/$source_dev of=/dev/$target_dev
  rc=$?
  if [ $rc != 0 ]; then # dd failed
    echo "Error: dd bs=4096 if=/dev/$source_dev of=/dev/$target_dev failed with $rc"
    chccwdev -d $target_vaddr # clean up
    cp_cmd DETACH $target_vaddr
    return 8
  fi
  chccwdev -d $target_vaddr
  echo "Copying disk via dasdfmt/dd succeeded ..."
  ret_val=1 # success with dasdfmt and dd
fi # if FLASHCOPY failed
sync # sync disks
  cp_cmd DETACH $target_vaddr
  echo "Copying disk via FLASHCOPY succeeded ..."
  return $ret_val
} # copy_disk()

#-------------------------------------
# function get_PARMFILE_info()
# Bring 191 minidisk online to be read by cmsfs and check for two PARMFILE files
#-------------------------------------

}
# recycle 191 to pick up latest changes
chccwdev -d 191
chccwdev -e 191

# sleep 1 # timing problem?
rc=$?
if [ $rc != 0 ]; then # unable to enable 191 disk
echo "unable to enable 191, rc from chccwdev = $rc"
exit 5
fi
CMSdisk=`lsdasd | grep 0191 | awk '{ print $7 }'`
cmsfslist -d /dev/$CMSdisk | grep -i $clone_id | grep PARM-S10
rc=$?
if [ $rc != 0 ]; then
echo "Error: $clone_id PARM-S10 not found on 191 minidisk. Exiting"
exit 6
fi

# get information about target
export local $(cmsfscat -a -d /dev/$CMSdisk $clone_id.PARM-S10)
target_hostname=$Hostname
target_IP=$HostIP
target_DNS=$Nameserver
target_GW=$Gateway
target_mask=$Netmask
target_broadcast=$Broadcast
prepare_vaddr $ReadChannel
target_readdev=$new_vaddr
prepare_vaddr $WriteChannel
target_writedev=$new_vaddr
prepare_vaddr $DataChannel
target_datadev=$new_vaddr

# get information about source
source_guestID=`cat /proc/sysinfo | grep "VM00 Name" | awk '{ print $3 }'`
export local $(cmsfscat -a -d /dev/$CMSdisk $source_guestID.PARM-S10)
source_hostname=$Hostname
prepare_ipaddr $HostIP
source_IP=$new_ipaddr
prepare_ipaddr $Nameserver
source_DNS=$new_ipaddr
prepare_ipaddr $Gateway
source_GW=$new_ipaddr
prepare_ipaddr $Netmask
source_mask=$new_ipaddr
prepare_ipaddr $Broadcast
prepare_vaddr $ReadChannel
source_readdev=$new_vaddr
prepare_vaddr $WriteChannel
source_writedev=$new_vaddr
prepare_vaddr $DataChannel
source_datadev=$new_vaddr

# ask_are_you_sure() # Ask "Are you sure?" - if not, then exit
function ask_are_you_sure()
{ #+--------------------------------------------------------------------------+
    # Ask "Are you sure?" - if not, then exit
    #+--------------------------------------------------------------------------+
}

Appendix C. Source code specific to Linux

```bash
# Appendix C. Source code specific to Linux

# This code is used for cloning a Linux system.

echo ""  # Output a blank line.

if [ $was = "yes" ]; then
  echo "A WebSphere system will be cloned"
fi

if [ $db2 = "yes" ]; then
  echo "A DB2 system will be cloned"
fi

if [ $mqs = "yes" ]; then
  echo "An MQ Series system will be cloned"
fi

read ans  # Read user input.

if [ $ans != "y" ]; then
  exit 7
fi

read descr  # Read a brief description of the server.

function copy_system()
{
  # Copy master image on $sdisk (default 100) to target user ID

  echo "Copying $sdisk root file system to $clone_id ..."
  copy_disk $sdisk to $clone_id 1100 # copy /sles10master to target root file system
  rc=$?
  if [ $rc -gt 1 ]; then # both FLASHCOPY and dasdfmt/dd failed
    echo "Copying disk failed with $rc"
    exit 9
  fi
}

function modify_cloned_image()
{
  # Mount newly copied system over /mnt/sles10cloned and modify networking info
  # Arg 1: target userid
  # Arg 2: target minidisk
  # Arg 3: target virtual device address
  # Return code
  #   0: success

  target_userid=$1
  target_mdisk=$2
  target_vaddr=$3
  echo "Mounting newly cloned image over $cloned_mnt_pt ..."
  cp_cmd LINK $target_userid $target_mdisk $target_vaddr MR
  rc=$?
  if [ $rc != 0 ]; then # LINK failed
    echo "Fatal error: CP LINK $target_userid $target_mdisk $target_vaddr MR failed with $rc"
    cp_cmd DET $target_vaddr
    exit 10
  fi

  # Further modifications to the cloned system...
}

# Further source code...
```

Appendix C. Source code specific to Linux  255
if [ ! -d $cloned_mnt_pt ]; then
  mkdir $cloned_mnt_pt
  rc=$?
  if [ $rc != 0 ]; then
    echo "Fatal Error: mkdir $cloned_mnt_pt failed with $rc"
    cp_cmd DET $target_vaddr
    exit 10
  fi
fi
chccwdev -e $target_vaddr
sleep 2
rc=$?
if [ $rc != 0 ]; then
  echo "Fatal error: chccwdev -e $target_vaddr failed with $rc"
  cp_cmd DET $target_vaddr
  exit 10
fi
cloned_DASD=`cat /proc/dasd/devices | grep "$target_vaddr(ECKD)" | awk '{ print $7 }'`
if [ ""cloned_DASD" = "" ]; then
  cat /proc/dasd/devices
  echo "Fatal error: can't find $target_vaddr(ECKD) in /proc/dasd/devices"
  chccwdev -d $target_vaddr
  exit 10
fi
cloned_fs="/dev/${cloned_DASD}1"
echo "Mounting $cloned_fs over $cloned_mnt_pt ..."
mount $cloned_fs $cloned_mnt_pt
rc=$?
if [ $rc != 0 ]; then
  echo "Fatal error: mount $cloned_fs $cloned_mnt_pt failed with $rc"
  lsdsad
  cp_cmd DET $target_vaddr
  exit 10
fi
echo "Modifying cloned image under $cloned_mnt_pt ..."
  sed --in-place -e "s/$source_hostname/$target_hostname/g" "
$cloned_mnt_pt/etc/HOSTNAME
  sed --in-place -e "s/$source_IP/$target_IP/g" "
  -e "s/$source_guestID/$clone_id/g" "$cloned_mnt_pt/etc/hosts
  sed --in-place -e "s/$source_GW/$target_GW/g" "
$cloned_mnt_pt/etc/sysconfig/network/routes
  sed --in-place -e "s/$source_IP/$target_IP/g" "
  -e "s/$source_mask/$target_mask/g" "
  -e "s/$source_broadcast/$target_broadcast/g" "
$cloned_mnt_pt/etc/sysconfig/network/ifcfg-qeth-bus-ccw-$target_readdev
  sed --in-place -e "s/$source_DNS/$target_DNS/g" "
$cloned_mnt_pt/etc/resolv.conf
# Regenerate SSH keys
  echo "Regenerating SSH keys in $cloned_mnt_pt/etc/ssh/ ..."
  rm $cloned_mnt_pt/etc/ssh/ssh_host*
  ssh-keygen -t rsa -N "" -q -f $cloned_mnt_pt/etc/ssh/ssh_host_rsa_key
  ssh-keygen -t dsa -N "" -q -f $cloned_mnt_pt/etc/ssh/ssh_host_dsa_key
  ssh-keygen -t rsa1 -N "" -q -f $cloned_mnt_pt/etc/ssh/ssh_host_key
# Remove any old entry, then copy clone's public key to known_hosts file
  echo "Adding $target_IP to known_hosts file"
  cd /root/.ssh
  grep -v $target_IP known_hosts > known_hosts.temp
  mv known_hosts.temp known_hosts
Appendix C. Source code specific to Linux

```bash
# construct known_hosts entry with three fields: IP@ "ssh-rsa" clones_public_key
new_key=`cat $cloned_mnt_pt/etc/ssh/ssh_host_rsa_key.pub | awk '{ print $2 }'`
 echo "$target_IP ssh-rsa $new_key" >> known_hosts

# see if IBM middleware servers need to be created
if [ $was = "yes" ]; then # create a WebSphere server
    clone_was $cloned_mnt_pt
fi
if [ $db2 = "yes" ]; then # create a DB2 server
    clone_db2 $cloned_mnt_pt
fi
if [ $mqs = "yes" ]; then # create a MQ Series server
    clone_mqs $cloned_mnt_pt
fi

# clean up
sync # sync disks
umount $cloned_mnt_pt
chccwdev -d $target_vaddr
```

```bash
#+--------------------------------------------------------------------------+
function make_backup_dir()
# Create a directory /backup/<VM-ID>-on-<IP.addr> if it doesn't exist
# e.g. LINUX04 with IP@ 129.40.178.44 would create /backup/LINUX04-on-129.40.178.44/
#+--------------------------------------------------------------------------+
{
    if [ -d $backup_dir ]; then # main backup directory exists
        umask 002
        echo "Creating a directory under $backup_dir"
        cd $backup_dir
        backup_subdir=$clone_id-on-$target_IP
        if [ ! -d $backup_subdir ]; then # backup subdirectory does not exist
            mkdir $backup_subdir
            if [ ! $? = 0 ]; then # there was an error from mkdir command
                echo "Error creating directory $backup_dir/$backup_subdir/"
            else
                echo "Created directory $backup_dir/$backup_subdir/"
                echo $descr > $backup_subdir/descLog.txt
            fi
        else # directory exists - create descLog.txt file
            echo $descr > $backup_subdir/descLog.txt
        fi
    fi
}

#+--------------------------------------------------------------------------+
function clone_was()
#   Arg 1: Mount point of the newly cloned server
# Clone a WebSphere Application Server
#+--------------------------------------------------------------------------+
{
    echo "Cloning WebSphere ..."
    echo "Modifying zipl.conf and running zipl ..."
    cd $1/etc
    cp zipl.conf zipl.conf.orig
    sed -i -e 's:dasd=100-102:dasd=100-102,300(ro),400(ro):g' zipl.conf
    chroot $1 zipl
```
echo "Modifying fstab ..."
cp fstab fstab.orig
sed -i -e '4a LABEL=was-prod /opt/IBM/WebSphere ext2 ro, acl, user_xattr' fstab
0 0' fstab
echo "making symlink to /etc/init.d/wasprofile ..."
cd $1/etc/init.d/rc3.d
ln -s ../wasprofile S99wasprofile
cd
}

}+--------------------------------------------------------------------------+
function clone_db2()
# Arg 1: Mount point of the newly cloned server
# Clone a DB2 Server
}+--------------------------------------------------------------------------+
{
  echo "Cloning DB2 ..."
  echo "Modifying zipl.conf and running zipl ..."
  cd $1/etc
  cp zipl.conf zipl.conf.orig
  sed -i -e 's:dasd=100-102:dasd=100-102,301(ro),401(ro):g' zipl.conf
  chroot $1 zipl
  echo "Modifying fstab ..."
  cp fstab fstab.orig
  sed -i -e '4a LABEL=db2-prod /opt/IBM/db2 ext2 ro, acl, user_xattr' fstab
  0 0' fstab
  echo "Modifying /etc/inittab"
  echo "fmc:2345:respawn:/opt/IBM/db2/V8.1/bin/db2fmcd #DB2 Fault Monitor Coordinator"
  >> $1/etc/inittab
  echo "Making symlink to /etc/init.d/db2instance ..."
  cd $1/etc/init.d/rc3.d
  ln -s ../db2instance S98db2instance
  cd
}

}+--------------------------------------------------------------------------+
function clone_mqs()
# Clone a MQ Series Server
}+--------------------------------------------------------------------------+
{
  echo "Cloning MQ Series ..."
  echo "Modifying zipl.conf and running zipl ..."
  cd $1/etc
  cp zipl.conf zipl.conf.orig
  sed -i -e 's:dasd=100-102:dasd=100-102,302(ro),402(ro):g' zipl.conf
  chroot $1 zipl
  echo "Modifying fstab ..."
  cp fstab fstab.orig
  sed -i -e '4a LABEL=mqs-prod /opt/mqm ext2 ro, acl, user_xattr' fstab
  0 0' fstab
  echo "Making symlink to /etc/init.d/mqmuser ..."
  cd $1/etc/init.d/rc3.d
  ln -s ../mqmuser S97mqmuser
  cd
}

# main() # some important global variables
clone_id="none"
was="no"                                # middleware cloning - not in Feb 07
db2="no"                                 # book but hopefully coming
mqs="no"
# modify the following if you want a master image at a different minidisk
sdisk=100 # source minidisk to clone
cloned_mnt_pt="/mnt/sles10cloned" # directory of temporary mount point
backup_dir="/backup/linux" # directory of Linux backups

# begin function calls
process_arguments $@ # process arguments passed by user
if [ $verbose = "on" ]; then set -vx; fi # turn on debug
check_target_id # user ID must exist and be logged off
get_PARMFILE_info # get info from parm files
ask_are_you_sure # confirm disks will be overwritten
copy_system # copy disk $sdisk to target ID
modify_cloned_image $clone_id $sdisk 1100 # modify newly copied system
cp_cmd XAUTOLOG $clone_id # bring new clone to life
make_backup_dir # make a backup directory
if [ $verbose = "on" ]; then set +vx; fi # turn off debug
echo "Successfully cloned minidisk $sdisk to $clone_id"
echo "You should be able to ping $target_IP within one minute"
exit 0
Index

Symbols
$ALLOC$, z/VM user ID   54, 91, 139, 150
/etc/apache2/httpd.conf file   157
/etc/apache2/vhosts.d/ directory   157
/etc/cron.daily/   111
/etc/exeports file   132
/etc/fstab   183
/etc/init.d/boot.local script   112, 127
/etc/ntp.conf file   125
/etc/passwd file   164
/etc/samba/smb.conf file   168
/etc/sysconfig/apache2 file   158
/etc/zipl.conf   113
/proc/sys/appldata/ directory   210
/proc/sys/kernel/hz_timer   113
/root/.ssh/authorized_keys file   127
/usr/share/doc/ directory   169

Numerics
3270 emulators   27, 41
520RES   36

A
Aduva's Onstage   4
Albert Einstein   4, 223
Alt-Tab sequence   30
Apache   155
    Enabling for SSL   156
    Virtual hosts   157
Apache resources   159
appldata_mem module   210
appldata_net_sum module   210
appldata_os module   210
appletviewer command   173
application development resources   174
application development server   171
APPLMON option   209
ATTACH command   34
Attachmate Extra!   27
AUTOLOG1, z/VM user ID   230

B
backup_vm script   224
Bill of materials   9
boot.local   112

C
C/C++ compiler   172
C/C++ development   172
Certificate

Self-signed   156
signing request   157
Certificate Authority   156
CHANGEPW XEDIT   60
CHANGEPW XEDIT macro   60

 chapters
    in the book   7
chkconfig command   86, 108, 170
choices made
    in the book   4
CHPW52 XEDIT   60
    source code   237
chroot command   134
close.sh script   145, 168, 171, 227
    description   146
    source code   249
cloning   139
    Six more virtual servers   148
cmsfs package   123
cmsfsclat command   123
cmsfslist command   123
Common UNIX Printing System (CUPS)   167
Computer Aided Design (CAD)   172
Computer Associates
    VM Direct   4
    VM Monitor   197
    VM Secure   59
Computer Associates'
    Brightstor Enterprise Backup   227
Control Program   4
Controller   7, 89, 148
    applying service   122
    changing source of installation   135
    Configuration   121
    Copying files to   121
    Disk partitioning   115
    Installing   114
    Modifying master image   134
    NTP server configuration   124
    Remove unnecessary RPMs   122
    Turn off unneeded services   122
    Verify your changes   130
conventions
    command retrieval   12
    in the book   10
    naming backup files   11
    volume labeling   11
CP   4
CPACCESS command   43
CPFMTXIA command   47
CPFFORMAT EXEC   48
    using it   48
CPSYNTAX   52
CPSYNTAX command   43
CUPS   171
D
DASD
  3390  5
  activating in a SLES10 install  98
  ECKD  5
  formatting in a SLES10 install  98
  labelling convention  11
dasd= parameter
  in zipl.conf  113
dasdfmt command
  using in a for loop  180
dd command  146
 DEFINE STORAGE command  96, 160
desktop  21
df command  114
 DIRECTXA command  55, 60, 91, 150
 DirMaint  4
 Disaster recovery  223
 Disconnect Timeout  42
 Disconnect_Timeout  43
 DISKMAP command  53
distributions
  31-bit and 64-bit  ix

E
egrep command  168
evelope files  72
exportfs command  133
ext3 file system
  creating  183

F
FCONX LINUXUSR file  210
fdasd command  180
Features statement  42
FILELIST command  93
findutils-locate package  104
FLASHCOPY command  90, 146
FORMAT command  55
formatting DASD fast  48
FTP
  site fix 80 subcommand  93
FTP server  45

G
gcc  172
gdb  173
GeoTrust  157
getVMinfo script  224
GNU compiler collection (gcc)  172
GNU debugger (gdb)  173
gq package  164

H
hardware resources  9
HELP command  200
HMC  29

CD-ROM/DVD  31
  Load icon  35
  Operating System Messages  46
  Single Object Operations  30
  SYSG  36
  System Console (SYSC)  46
  Hummingbird Host Explorer  27
  hypervisor  4
  hz_timer  112, 127

I
IBM Personal Communications  27
IBM RACF  59
IBM Tivoli Provisioning Manager  4
IBM z/VM Performance Toolkit  197
  monitor subcommand  207
IBM’s Tivioli Storage Manager (TSM)  227
id command  165
Incremental back up  223
INDICATE command  198
INDICATE LOAD command  198
INDICATE QUEUES EXP command  199
infrastructure
  block diagram  6
design  5
initial RAMdisk  92
Innovation Data Processing’s FDR/UPSTREAM  227
INSTPLAN  33
INSTVM EXEC  38
Integrated 3270 Console  30, 32
IPL CMS  37, 96, 160
IPWIZARD command  38

J
jar command  174
Java  173
Java development
tool  173
Java Runtime Environment (JRE)  173
jdb command  174

K
kernel  92
kernel.hz_timer  112

L
LABEL520 EXEC
  source code  240
LAMP  171
LDAP client  162, 165
LDAP RPMs  166
LDAP server
  manual install  160
ldapadd command  164
ldapsearch command  164–165
Levanta  4
Lightweight Directory Access Protocol (LDAP)  160
Linux
Disaster recovery 223
Incremental backup 223
manual installation 5
monitoring 197, 209
monitoring from the kernel 209
Linux distribution 160, 171–172
Linux PC 83
Linux RMF PM Data Gatherer (rmfpms) 208
LINUX01 139, 155
LINUX02 160
LNXDFLT profile 90
LNXMAINT 53
   copying files to 56
LNXMAINT 192 224
Load icon 35
locate command 104
logical partition ix, 3
logical volume 91
   adding 179
LPAR
   requirements 9
lsmod command 210
lvcreate command 182
lvdisplay command 183
LVM
   block diagram 181
   physical extents 182
MAINT 2CC 230
MAINT CF1 230
make command 173
Map Network Drive 170
Master image 7, 89, 148
   Adding additional RPMs 104
   Changing source of installation 133
   Configuration 104
   Create nightly.sh 104
   Installation 94
   Modifying while controller is running 134
   Remove unnecessary RPMs 107
   Turn off unneeded services 108
   Verifying changes 114
MDISK statement 53
Minidisk cache (MDC) 198
Minidisk passwords 59
mke2fs command 183
modprobe command 210
monitor stream support 209
monitoring Linux 208–209
mount command 120
MySQL 171
N
Network File System (NFS) 83, 131
Network Time Protocol (NTP) 124
Network Time Protocol daemon (NTPD) 111
networking resources 10
NFS server
   configuration on controller 131
   configuration on Linux PC 86
nmbd daemon 169
nss_ldap 162, 166
NTP
   setting clock accurately 111
   ntpd command 111
   ntpdate command 125
   ntpq command 126
   ntptrace command 126
OpenLDAP
   configuration 163
   pointing virtual server to LDAP server 165
OpenLDAP server 161
openmotif 104
openssl command 156
OPTION APPLMON 139, 149
overlaps
   searching for 54
P
paging space 47
paging volumes 47
   location 49
passwd command 169
passwords
   planning 13
Performance Toolkit 201
PERFSVM user ID 203
Perl 172
philosophy of the book 4
PHP 171–172
physical extents 182
   planning 9
Printing
   configuration 171
PROFILE EXEC
   AUTOLOG1’s 57, 92, 140, 230
   LNXMAINT’s 56
   PERFSVM’s 207
PROFILE LNXDFLT 90
PROFILE TCPIP 45
Programming 69
PTF 70
PUT2PROD command 38
PuTTY 21
   customizing 21
   larger screen 24
   potential security breach warning 120
   pvcreate command 181
   pvdisplay command 181
Index 263
TCPIP Data 230
TCPMAINT 198 230
TCPMAINT 592 230
Thawte 157
timer pop 113
Tool Command Language (TCL) 172

U
UM97510 75
UM97520 75
Universal Naming Convention (UNC) 170
updatedb command 104
usability tests
  on the book 6
USER DIRECT 230
USER DIRECT file 53, 90, 149
  APPLMON option 209
  INCLUDE statement 90
  profile 90
USER DISKMAP file 53
User_Volume_Include 51, 149
User_Volume_List 51
useradd command 169
userid() function 93

V
VDISK 42
Vdisk
  allowing creation of 43
Velocity Software’s ESALPS 197
VeriSign 157
Veritas’ Backup Exec 227
vgcreate command 182
virtual disk (VDISK) 42
Virtual hosts 157
virtual NIC 90
Virtual server
  application development 171
  File and print 167
  LDAP 160
  LDAP configuration 163
  Web server 155
virtual servers 4
virtual switch (VSWITCH) 3, 42
vmpc command 130
VMLINK command 50
VNC 96
  configuring on controller 122
  configuring on master image 105
volume group 182
VSWITCH 3, 44, 47, 151

W
which command 145
Windows workstation 83
worksheets
  examples in the book 14

X
XAUTOLOG command 44, 57, 146
XEDIT
  " prefix command 90, 149
  ? subcommand 12
  add subcommand 43
  BOTTOM subcommand 43
  CHANGE subcommand 61
  FILE subcommand 43
  PREFIX OFF subcommand 53
  search subcommand (/) 42
XEDIT command 41
XEDIT PROFILE 41
xntpd daemon 111

Y
yast
  changing source of installation 134
  Installing RPMs 104
  Setting LDAP client 166, 188

Z
z/VM
  adding paging volumes 47
  backing up to tape 61
  cons=sysg 36
  conventions 10
  CPFMTXA command 47
  customizing shutdown 57
  customizing startup 57
  DIRECTXA command 55
  Disaster recovery 223
  Disconnect Timeout 42
  DISKMAP command 53
  envelope files 72
  FORMAT command 55
  FTP server 45
  HELP command 200
  HIT RATIO 198
  Incremental back up 223
  INDICATE command 198
  INDICATE QUEUES EXP command 199
  INSTPLAN EXEC 33
  INSTVM EXEC 38
  Integrated 3270 Console 30
  IPWIZARD command 38
  latest PTFs 75
  MIGRATE rate 198
  monitoring 197
  paging efficiency 49
  PAGING rate 198
  PUT2PROD command 38
  QUERY ALLOC MAP command 200
  QUERY CPLEVEL command 200
  QUERY DASD command 201
  QUERY NAMES command 200
  QUERY OSA command 201
  QUERY PROCESSORS command 200