Linux for S/390
Installation Hands-on Workshop

Richard Lewis - rflewis@us.ibm.com
Chuck Morse - morsec@us.ibm.com

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

ESCON*	PowerPC*
HiperSockets	S/390*
IBM*	VM/ESA*
IBM eServer	z/VM
IBM logo*	zSeries
Multiprise*	FICON

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Intel is a trademark of the Intel Corporation in the United States and other countries.

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

Lotus, Notes, and Domino are trademarks or registered trademarks of Lotus Development Corporation.

Linux is a registered trademark of Linus Torvalds.

Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.

SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

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

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the 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.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.

Agenda

- **Introduction**
  - ► **Brief History of Linux®**
  - ► **Hardware Requirements, Connectivity Options and Distributions**
  - ► **Linux File System and Common Linux Commands**
  - ► **Intro to Lab and Installation Overview**
- **Basic Lab Exercises**
  - ► **Installation of Linux for S/390®**
    - – Marist (2.2.16)
    - – SuSE SLES7 (2.4.7)
    - – Red Hat 7.2 (2.4.9)
  - ► **Basic Linux for S/390 System Administration**
- **Elective Lab Exercises**
  - ► **Rebuild the Linux Kernel**
  - ► **Using Linux as a Firewall with Ipchains**
  - ► **Using Linux as a DNS with BIND-8**
  - ► **File serving with Samba**
  - ► **Apache Web Server Installation and Customization**
  - ► **KDE Installation**

Brief History of Linux

What is Linux ?

- **Linux is the kernel of a UNIX® technology (-like) operating system, originally developed by Linus Torvalds**

- **It was developed / tested by the Open Source community**
  - ► **Highly disciplined / structured**
  - ► **High quality**
  - ► **Secure**
  - ► **Stable**

- **Not just for Intel® processor-powered PCs**
  - ► **PowerPC®, Sparc, Alpha, S/390**
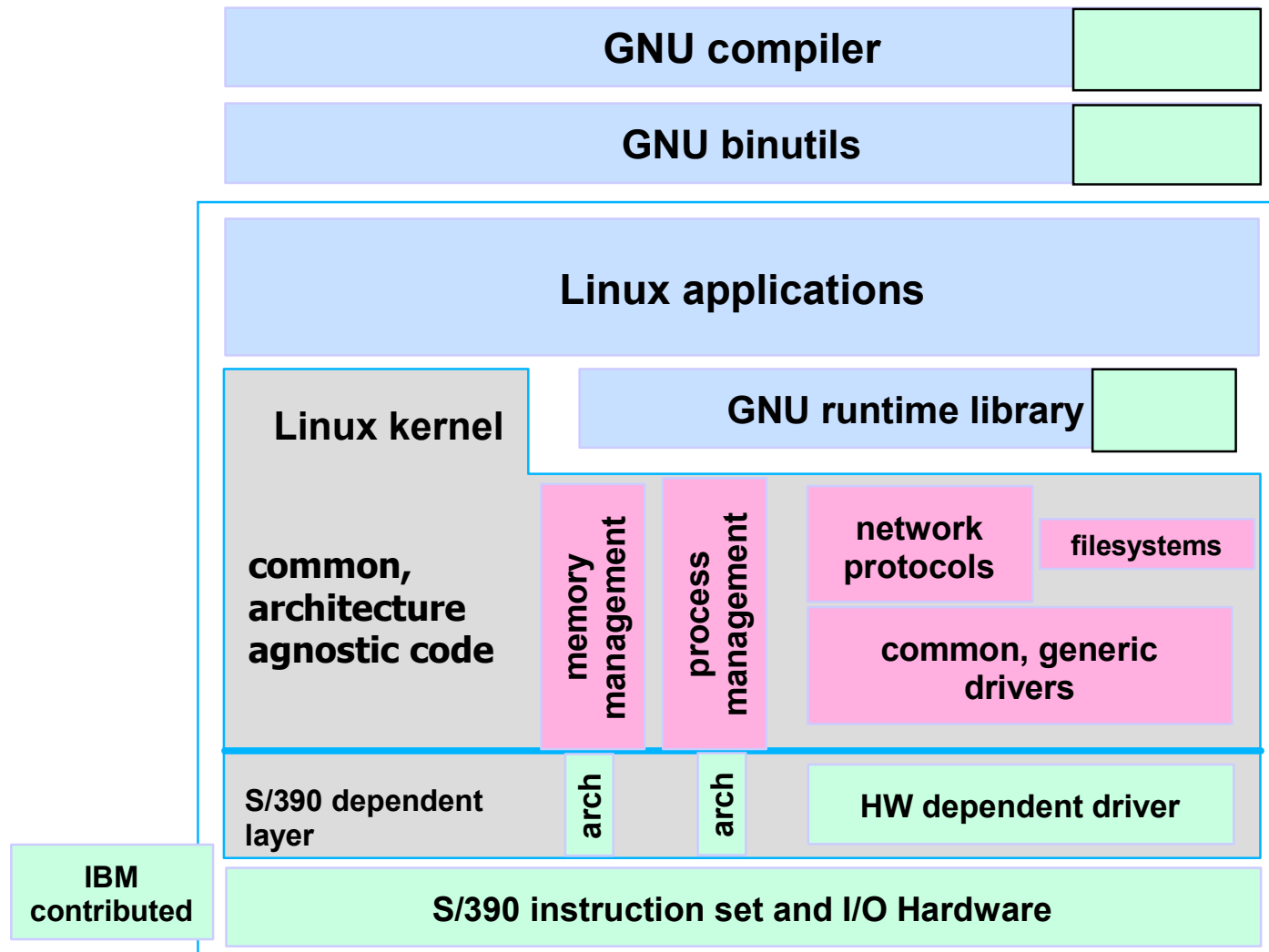 - ► **Over 100 platforms supported today**

What is Linux on zSeries?

- **A native IBM $e$server zSeries™ operating environment**
  - ► **Pure Linux, an ASCII environment**
  - ► **Exploits IBM S/390 hardware, including IEEE floating point**
  - ► **Linux for S/390 - 32-bit**
  - ► **Linux for zSeries - 64-bit**

- **Not a unique version of Linux or other operating system**

- **Not a replacement for other IBM $e$server zSeries operating systems**

Linux Architecture

GNU compiler

GNU binutils

Linux applications

Linux kernel

GNU runtime library

common, architecture agnostic code

memory management

process management

network protocols

filesystems

common, generic drivers

S/390 dependent layer

arch

arch

HW dependent driver

IBM contributed

S/390 instruction set and I/O Hardware

The look and feel of Linux does not change on S/390 and zSeries

Hardware Requirements, Connectivity Options and Distributions

IBM

Hardware Exploitation

- **Multiple Supported Environments**
  - ► **ESA/390 single image**
  - ► **S/390 or zSeries Logical Partition**
  - ► **z/VM™ (or VM/ESA®) Guest**

- **Exploits IBM S/390 and zSeries Hardware**
  - ► **Hardware Management Console (HMC)**
  - ► **3380/3390 ECKD DASD**
  - ► **FBA (9336 or VDISK)**
  - ► **FICON / ESCON® / Parallel Channels**
  - ► **FCP attached SCSI devices**
  - ► **OSA/2 or OSA-Express Adapters**
  - ► **IEEE Floating Point**
  - ► **Expanded Storage**
  - ► **Magnetic Tape**
  - ► **HiperSockets™**

IBM

Hardware Requirements

- **Processors**
  - ► **9672 G2 - G6 (IBM only supports G5+)**
  - ► **zSeries**
  - ► **Multiprise® 2000 (not supported by IBM)**
  - ► **Multiprise 3000**
  - ► **P/390, R/390, Integrated Server (not supported by IBM)**
  - ► **64 MB central storage (128 MB recommended)**

- **Connectivity**
  - ► **Network connectivity is required to aquire installation materials**

- **VM Linux guest support**
  - ► **z/VM V4 Recommended**
    - – Can be run under VM/ESA Version 2 Release 4

Hardware Requirements

- **Devices**
  - ► **DASD support via ECKD driver**
    - – One 3380 / 3390 / Multiprise internal disk volume
    - – One 500 cylinder minidisk (VM)
  - ► **System console function via**
    - – Hardware Management Console (LPAR or basic mode)
    - – Virtual 3215 console (VM)
  - ► **Network connection**
  - ► **Workstation with CD-ROM for installation**

Network Connectivity

- **LPAR or VM**
  - ► **Channel-to-Channel Adapter (CTCA)**
    - − ESCON and Parallel channels
  - ► **LAN Channel Station (LCS)**
    - − OSA/2 adapter (Ethernet and Token-Ring)
  - ► **Gigabit Ethernet**
    - − OSA-Express adapter on G5, G6 and zSeries
  - ► **HiperSockets**
    - − zSeries

- **VM Only**
  - ► **Virtual Channel-to-Channel Adapter (VCTCA)**
  - ► **Inter User Communication Vehicle (IUCV)**
  - ► **Guest LAN**

Available Distributions

- **Marist College — linux390.marist.edu**
  - ▶ 2.2.16 kernel
- **SuSE — suse.de/en/**
  - ▶ Linux Enterprise Server 7 for S/390 and zSeries (31-bit 2.4 kernel)
  - ▶ Linux Enterprise Server 7 for zSeries (64-bit 2.4 kernel)
  - ▶ Linux Enterprise Server 8 for IBM Mainframes (31-bit and 64-bit 2.4 kernels)
- **Red Hat — www.redhat.com**
  - ▶ Red Hat Linux 7.2 for S/390 - (31-bit 2.4 kernel)
  - ▶ Red Hat Linux 7.1 for zSeries - (64-bit 2.4 kernel)
- **ThinkBlue — linux.s390.org/**
  - ▶ ThinkBlue Linux for S/390 (31-bit)
  - ▶ ThinkBlue/64 7.1a (64-bit) Linux for zSeries
- **Debian — www.debian.org/ports/s390/**
  - ▶ Debian GNU/Linux Version 3 for S/390 (31-bit 2.4 kernel)
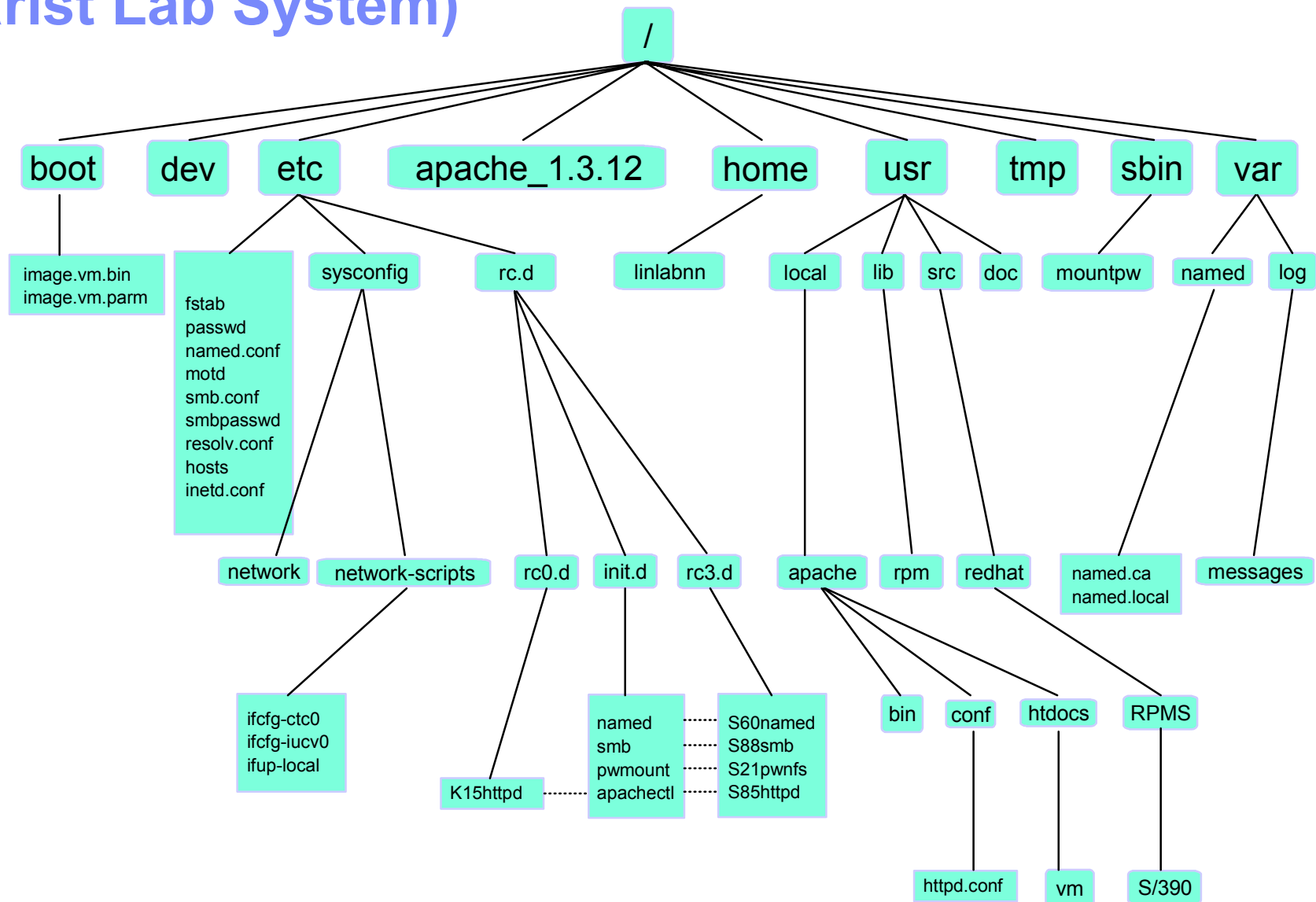
One Important Web Site

www.linuxvm.org

Linux File System Structure
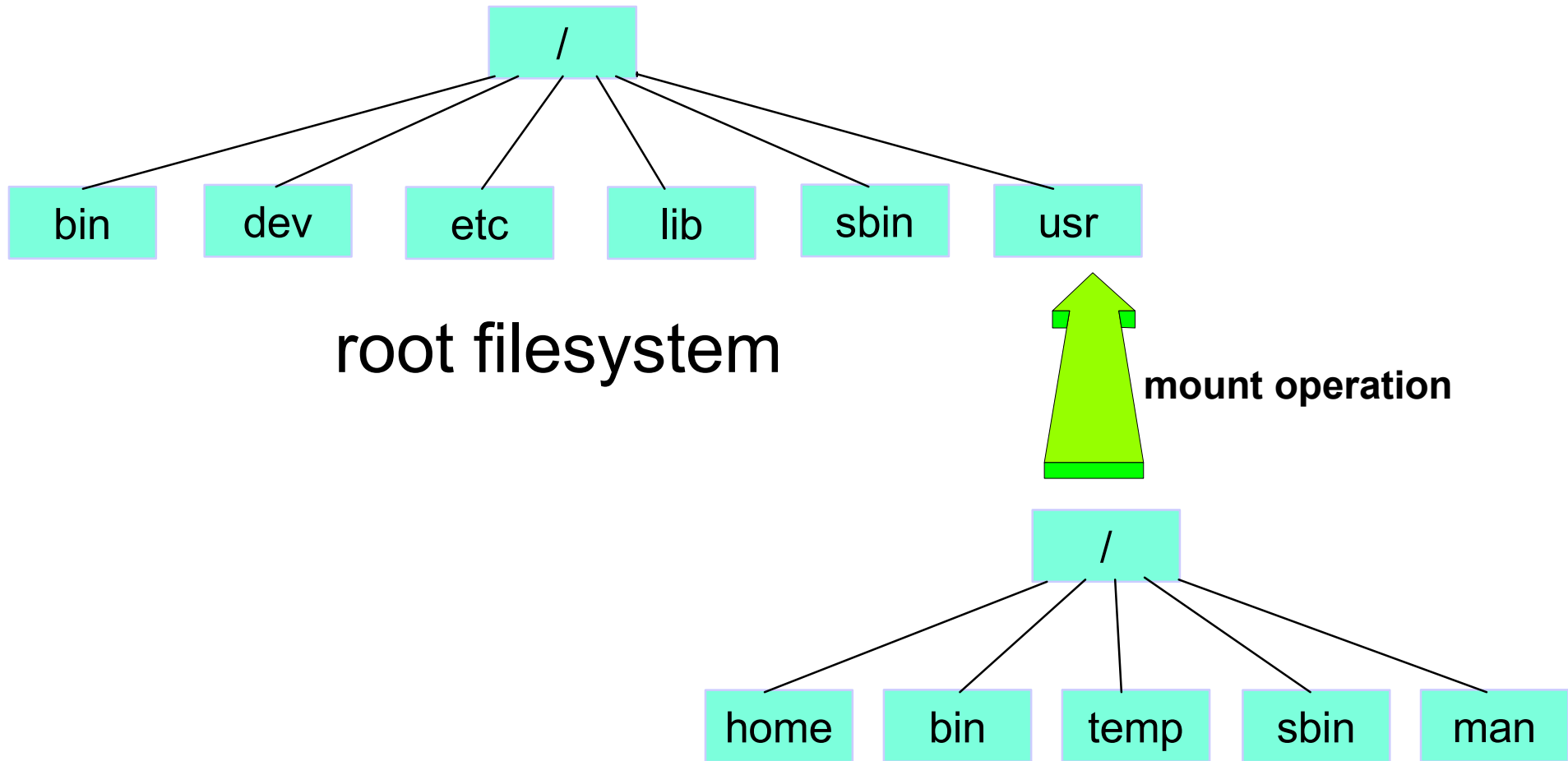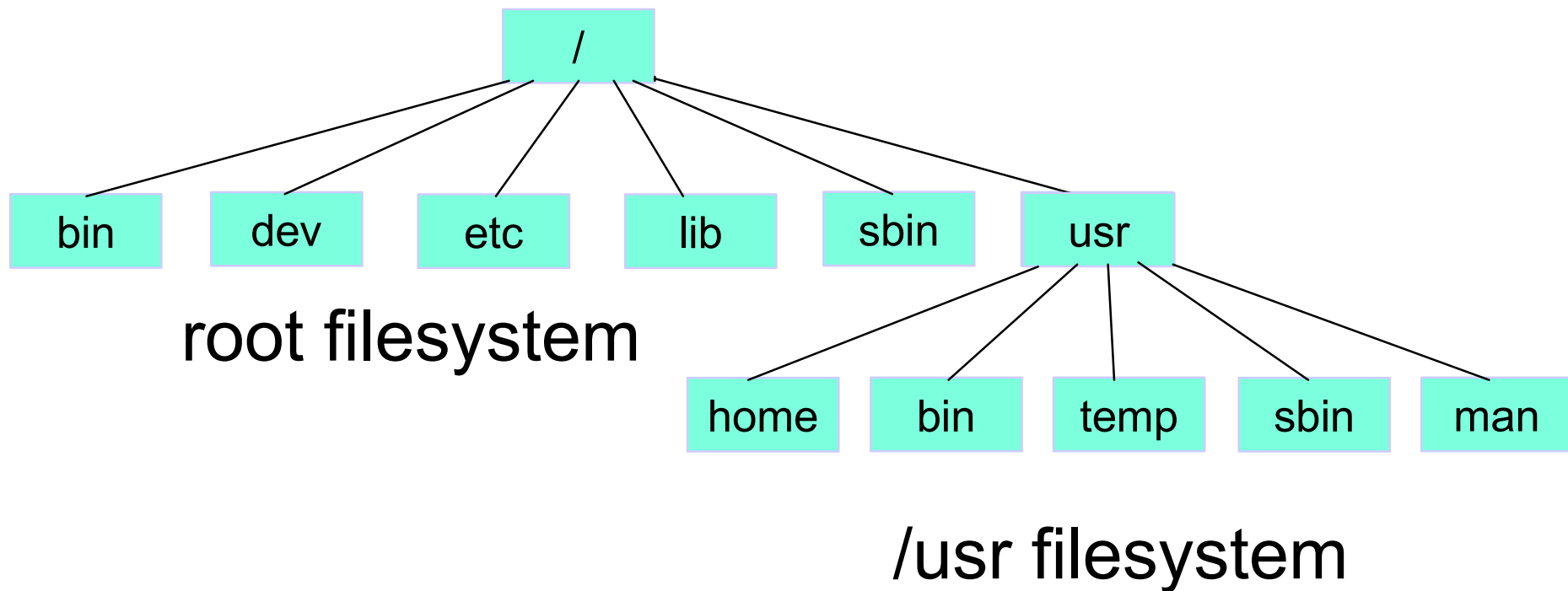
File System Structure (Marist Lab System)

File system tree diagram:

- **/** (root)
  - **boot**
    - image.vm.bin
    - image.vm.parm
  - **dev**
  - **etc**
    - fstab
    - passwd
    - named.conf
    - motd
    - smb.conf
    - smbpasswd
    - resolv.conf
    - hosts
    - inetd.conf
    - **sysconfig**
      - network
      - network-scripts
        - ifcfg-ctc0
        - ifcfg-iucv0
        - ifup-local
    - **rc.d**
      - rc0.d
        - K15httpd
      - init.d
        - named
        - smb
        - pwmount
        - apachectl
      - rc3.d
        - S60named
        - S88smb
        - S21pwnfs
        - S85httpd
  - **apache_1.3.12**
  - **home**
    - linlabnn
  - **usr**
    - local
      - apache
        - bin
        - conf
          - httpd.conf
        - htdocs
          - vm
        - RPMS
          - S/390
    - lib
    - src
      - rpm
    - doc
      - redhat
  - **tmp**
  - **sbin**
    - mountpw
  - **var**
    - named
      - named.ca
      - named.local
    - log
      - messages

Mounting a File System

root filesystem

mount operation

Mounting a File System (cont.)

root filesystem

/usr filesystem

complete hierarchy after mounting /usr

Directory Usage

/	root directory
/boot	boot files (kernel, parm file, system map)
/home	user directories
/dev	device files that represent system hardware
/etc	important system configuration files
/bin	commands needed to start the system
/sbin	critical system binaries, commands reserved for the superuser
/usr/doc	documentation files
/usr/man	manual files
/usr/src	source code for the system software
/usr/src/linux	the kernel sources
/tmp	temporary files
/var	configuration files (linked from /usr)
/lib	shared libraries
/proc	the process file system
/mnt	mount point for temporarily mounted filesystems
/usr	additional utilities and applications

Basic Configuration Files

/etc/fstab   - tells Linux what filesystems to mount when it starts

/etc/inittab   - parameters for the init process

/etc/rc.config   - primary configuration file for SuSE Linux (includes network config)

/etc/sysconfig/network - general network configuration for Red Hat and Marist Linux

/etc/sysconfig/network-scripts/ifcfg-iucv0

/etc/sysconfig/network-scripts/ifcfg-ctc0

                             - interface specific configuration files for Red Hat and Marist Linux

/etc/motd   - "Message of the Day" file, contents are displayed when users login

/etc/passwd   - Contains user names, numbers, home directories, and login shell

/etc/group   - Contains user groups

/etc/shadow   - Contains passwords

/etc/hosts   - Contains hostname to IP address assignments.

Common Commands

IBM

Linux Commands Used in Class

Command		Syntax	Example
adduser	Creates a directory and an entry in the passwd file for a new user	**adduser userid**	**adduser linlab01**
cat	"Concatenate" View, create, and concatenate files	**cat [options] [inputfile] [outputfile]**	**cat ifup-local** Print file ifup-local on the display.
cd	"Change Directory" Used to change from your current working directory to another directory	**cd directory | ~username**	**cd /mnt/etc** Change directory to /mnt/etc **cd ~linlab01** Change directory to /home/linlab01
cp	"Copy" Copy a file	**cp source destination**	**cp fstab fstab.save**

dasdfmt	"DASD Format" Formats a device to be managed by the LINUX dasd driver	**dasdfmt [-tvy] [-s start_track] [-e end_track][-b blocksize]**
		-f devicename | -n 390_devno
		 where: -v = verbose, to
		           display more
		           messages
		       -y = omits the prompt
		           to reconfirm the
		           format request
		       -t = test mode (the
		           device will not
		           be formatted)
			dasdfmt -f /dev/dasda -b 4096
			
			Formats device /dev/dasda with a blocksize of 4096

Linux Commands Used in Class

Command		Syntax	Example
df	"disk free" Reports file system disk space usage	`df [-h]` where -h = display output in more human readable form	`df -h`
du	"disk used" Reports the space occupied by the current (or named) directory and all directories within it	`du [directory] [-sh]` where -h = display output in more human readable form -s = display summaries only	`du -h`
ed	"edit" Invokes the ed text editor	`ed filename`	

find	Locate files in a directory based on search criteria	`find [/directory]`
		`[-name filename]`
		`[-atime (+\-)`
		`#days_since_last_access]`
		`[-ok command {}]`
		`[-mtime`
		`#days_since_last_modified]`
		`[-print]`
			`find /home -name temp size +100 -atime +5 ok rm {}`
			
			Finds files named temp in the home directories larger than 100 blocks that have not been accessed in the last 5 days. When a file is located, you are asked if you want to delete it.

free	Display amount of free and used memory	`free`	

Linux Commands Used in Class

Command		Syntax	Example
gcc	"GNU C Compiler" Used to compile a source file into a binary executable file	**gcc [-o output_filename] [options] source_filename**	**gcc -o mountpw mountpw.c** Compile the mountpw.c file into a binary executable file called mountpw.
ifconfig	"Interface Configuration" Used to activate or shut down an interface, eg. channel-to-channel, IUCV, Token-Ring, Ethernet, PPP and loopback devices	**ifconfig [interface options | address]**	**ifconfig iucv0 9.130.240.161 pointopoint 9.130.240.101 mtu 9216** Activate the iucv0 interface at IP address 9.130.240.161 with a point-to-point connection to IP address 9.130.240.101 using a Maximum Transmission Unit size of 9216 bytes.
kill	Stop a process	**kill [PID] [-options]**	**kill 93 -HUP** Stop process number 93 and restart
last	Show listing of last logged in users	**last**	
lastlog	Format and print contents of the last login file	**lastlog**	

Linux Commands Used in Class

Command		Syntax	Example
ln	"link" Creates a link between one file and another. This allows the file to be located in one place and referenced in another.	ln [-s] source linkname where: -s = symbolic link	ln -s init.d/named S60named Creates a symbolic link which allows you to reference the file "named" in the "init.d" directory by the linkname of "S60named".
ls	"list" Displays the contents of a directory	ls [-al] where: -a = all l = long format	ls -al Lists all files in the current directory in the long format
mkdir	"Make directory" Creates a sub-directory under the current working directory	mkdir directory_name	mkdir boot Creates an empty directory called "boot".
mke2fs	"make ext2 file system" Creates a native LINUX ext2 file system.	mke2fs devicename [-b blocksize]	mke2fs /dev/mnda -b 4096 Creates a file system of type ext2 on device mnda with a blocksize of 4096.
mkswap	"make a swap partition" Used to create a LINUX swap partition	mkswap partitionname	mkswap /dev/mndb Makes minidisk device /dev/mndb a swap partition.

Linux Commands Used in Class

Command		Syntax	Example
mount	Tells the system that a device is available for use and specifies where in the filesystem you want it to be located.	mount [-t type] [-o accesstype] device mountlocation	mount -t ext2 -o ro /dev/mnda /mnt Makes device mnda, which contains an ext2 file system, accessible to the Linux system at location (directory) mnt, with read-only access.
mv	"move" Move or rename a file	mv source destination	mv ifcfg-ctc0 ifcfg-iucv0 Renames the file ifcfg-ctc0 to ifcfg-iucv0
nslookup	"name services lookup" Tool provided with BIND-8. Interactively queries Internet domain name servers.	nslookup	nslookup Usage: Enter "nslookup" to begin an interactive session with the tool. Enter a host name. nslookup will respond with the fully qualified name of the host and it's IP address. To end the interactive session, enter "exit".
passwd	Create/set a password for a userid	passwd userid	password linlab01 Enter the password when prompted.

Linux Commands Used in Class

Command		Syntax	Example
ps	"process" Displays the processes running on your system. Often used in conjunction with the kill command.	ps [-efl] where: e = select all processes f = provide full output listing l = display in the long form	ps -ef
rm	"remove" Erase a file	rm filename	rm apache_1.2.12.tar.Z
route	Used to manipulate the Linux kernel's routing table.	route [interface options]	route add -net default iucv0 Add the iucv0 interface to the routing table as the default interface.
rpm	"Redhat Package Manager" Installs products packaged by the Redhat Package Manager	rpm [-ivh --nodeps] [-qlp] packagename.rpm where: --nodeps = no dependency checking i = install a new package v = verbose h = display a progress indicator (hash marks) during installation q = query package info l = list all files in the package p = queries the packagefile	rpm -ivh --nodeps bind-8.2.2p3-1_s390.rpm Installs a new package with no dependency checking. A progress indicator and additional messages will be displayed.

Linux Commands Used in Class

Command		Syntax	Example
shutdown	Shut the system down	shutdown [-r | -h] [now] where: h = halt the system after it shuts down r = reboot after shutdown now = start the shutdown process immediately without warnings to users	shutdown -h now Shuts down the system immediately.
silo	Create an IPL/boot record on a device	silo [-f image_file] [-d boot_device] [-p parmfile] [-b boot_sector_file] -t2 Note: -t2 indicates "test level 2". Although this is not a parameter you would expect to use, it is still necessary at the current kernel level to write the IPL record.	silo -f image.vm.bin -d /dev/dasda -p image.vm.parm -b ipleckd.boot Creates an IPL record on device /dev/dasda using the image.vm.bin kernel image, the image.vm.parm kernel parameter file, and the ipleckd.boot boot sector file.
swapon	"swap on" Activates or displays usage for a swap partition	swapon partitionname [-s] where: s = display usage information	swapon /dev/mndb Tells Linux to begin using the swap partition /dev/mndb
tail	View the end of a file	tail [-number_of_lines] filename The default number of lines shown is 10	tail -20 /var/log/messages Displays the last 20 lines of the file "messages"

Linux Commands Used in Class

Command		Syntax	Example

tar	"tape archive"
	Combines files and directory structure in one archive file or recreates files and directory structure from previous tar operations
		tar [-xzvfc] input_fn | output_fn
		 where: x = extracts files and directories from an archived file
		        z = zip (compress) or uncompress files
		        v = verbose - tells tar to list the files being archived or unarchived
		        f = specifies a filename for the archive file
		        c = creates an archive file
			tar -xzvf /tmp/initfs_big_Marist.tgz
			
			Extracts and uncompresses the files and directory structure from the file named initfs_big_Marist.tgz, listing all files as it works.

top	Display top CPU processes	top	
umount	"unmount" Unmount a mounted file system	umount mountlocation	umount /tmp
uptime	Tell how long the system has been running	uptime	
w	Show who is logged on, and resource usage	w	
who	Show who is logged on	who	
whoami	Show effective userid	whoami	

ed Editor

- **The ed editor has two modes:**
  - ► **Command mode - everything you type in is considered to be a command. Some commands you will be using are:**
    - – *number* positions the editor at line number
    - – a   append (add) text after the current line
    - – c   change a line
    - – i   insert text before the current line
    - – d   delete the current line
    - – p   display (print) lines
    - – w   save (write) lines
    - – q   end (quit) the editing session
    - – .   refers to the current line
    - – $ refers to the last line

  - ► **Input mode - after you have entered the a, c, or i subcommands, everything that follows will be text, until a period (.) is entered on a line by itself.**

Sample ed Session

```
ed fstab
1
.c
/dev/mnda      /      ext2  defaults,errors=remount-ro 0 1
.
1,$p
1,$w
q
```

- **This sequence of commands will:**
  - ► **begin editing on the file "fstab"**
  - ► **position the editor at line 1 in the file**
  - ► **indicate that the line is to be changed**
  - ► **enter the exact text that should replace the current line of text**
  - ► **indicate the end of changes**
  - ► **position the editor at line 1 and display (print) the file**
  - ► **position the editor at line 1 and save (write) the file**
  - ► **end (quit) the editing session**

Need Help?

- **The Linux equivalent of HELP is *man* (manual)**

  ► **Use *man <command>* to display help for that command.**

    – Output is presented a page at a time. Use *b* to scroll backward, *f* or a space to scroll forward, and *q* to quit.

Installation Overview

Basic Installation Steps

- **Acquire Linux Distribution**
  - ► **Kernel image**
  - ► **Ram disk**
  - ► **File system**

- **Prepare Environment**
  - ► **Configure virtual machine or LPAR**
  - ► **Gather network parameters**
  - ► **Create a boot parameter file**

- **Load the kernel, parm file and ram disk into storage**

- **Build the file system and configure system**

Create Parameter File

- **The parameter file provides information needed by the kernel at boot time**

- **Basic parameters include**
  - ► **mem=   defines the amount of storage to be used by Linux**
  - ► **mdisk= specifies the devices to be used by the minidisk driver(VM)**
  - ► **dasd=   specifies the devices to be used by the dasd driver**
  - ► **iucv=   identifies the virtual machine(s) to be connected via IUCV(VM)**
  - ► **root=   specifies the device containing the root file system**

Initial System Build
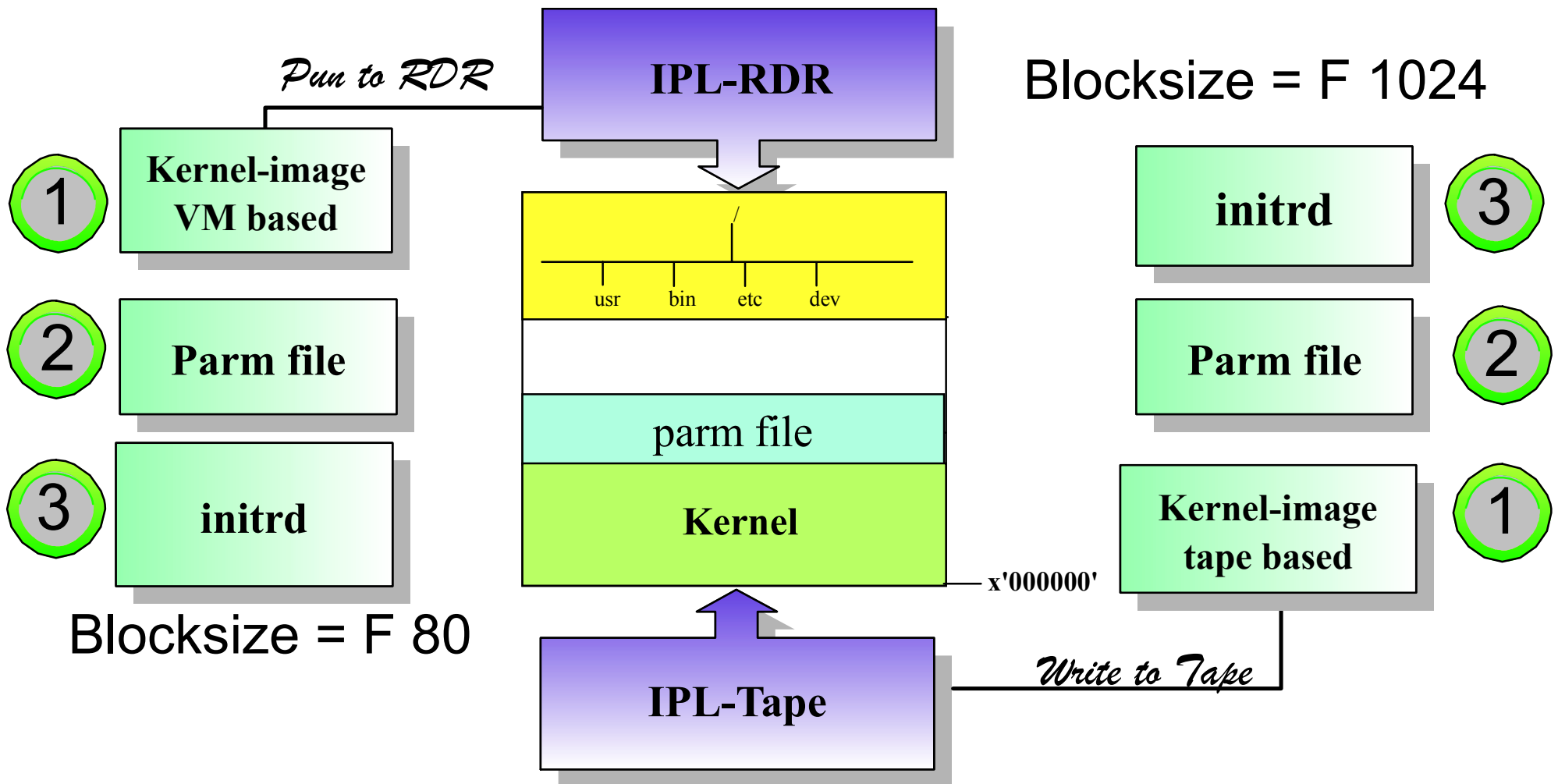
Build and Configure File System

- **Now that Linux is up and running you can**
  - ▶ **Create the file system**
    - – FTP the tar file to your Linux system
    - – Uncompress using the tar command
  - ▶ **Create a swap volume**
  - ▶ **Make the system bootable**
    - – Format a boot device
    - – Put boot files on the device
      - • kernel image
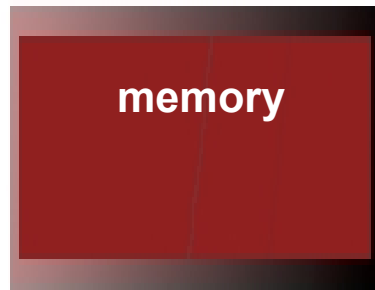   - • parameter file
      - • IPL text
    - – Run Silo (2.2.16) or zipl (2.4)

IBM

Hands-On Lab - Virtual Machine Configuration

```
mem=128m
mdisk=200,202,400
dasd=300
root=/dev/mnda ro
```
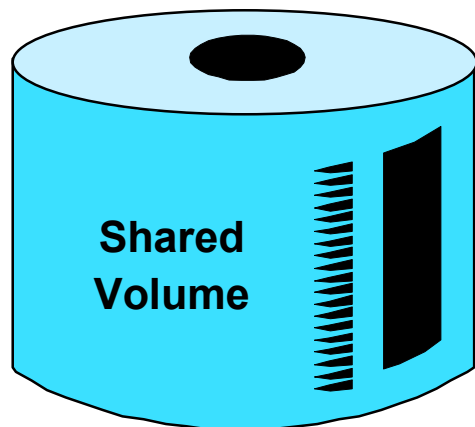
memory
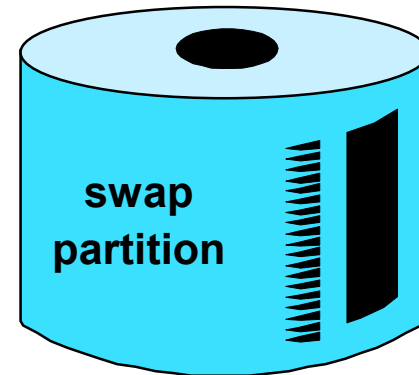
swap partition

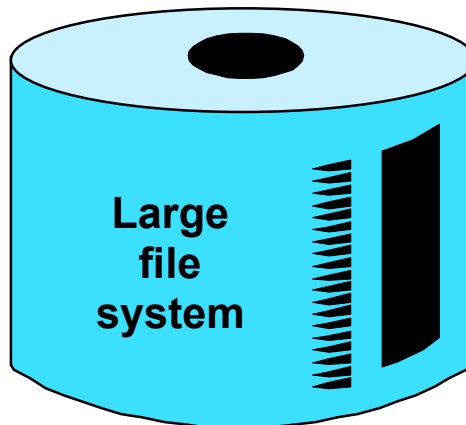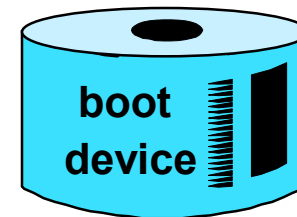202 mdisk
/dev/mndb

iucv

Shared Volume

Large file system

boot device

400 mdisk
/dev/mndc

200 mdisk
/dev/mnda

300 dasd
/dev/dasda
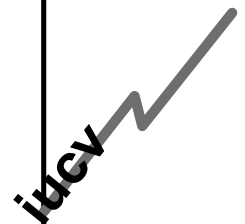
Hands-on Lab - Network Configuration

LINLAB01

x.x.x.x

OSA

x.x.x.x

9.82.56.1   9.82.56.131

TCP/IP   LINLAB01

CP

Multiprise 3000