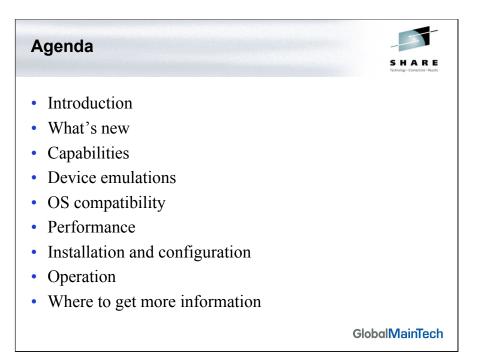


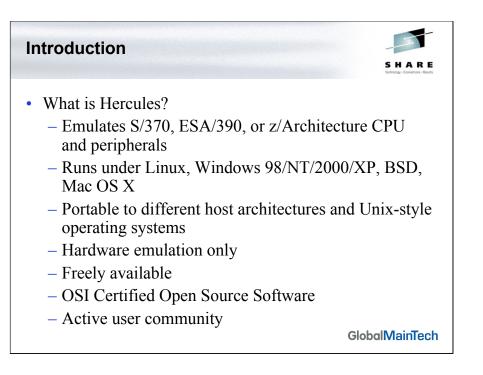
Mailing address: Jay Maynard Global MainTech Corporation 7836 Second Avenue South, Suite 1 Bloomington, MN 55420

(507) 235-3677 voice (507) 235-9787 fax

jmaynard@globalmt.com (work) jmaynard@conmicro.cx (home)

http://www.globalmt.com http://www.conmicro.cx



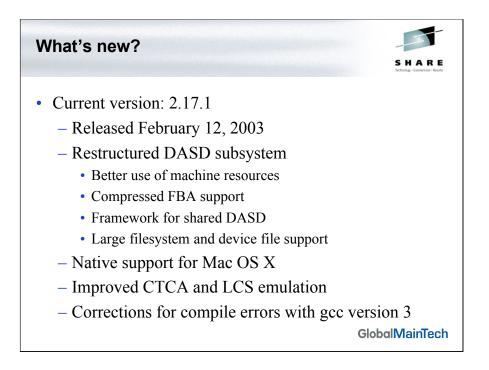


Hercules emulates the hardware of an IBM mainframe system. It doesn't provide any software services at all, any more than the bare iron does. You have to run some sort of OS on it.

One strength of Hercules is its user base. There are nearly 3000 members of the Hercules mailing list, and a large number of them are involved in running, fixing, and enhancing the package, and in getting OSes, old and new, running properly on it.

A lot of work has gone into portability. There are no intentional assumptions made about the host processor, and operating system-dependent features are automatically selected during the build process.

Hercules is OSI Certified Open Source Software. The QPL, the license under which Hercules is distributed, is accepted by the Open Source Initiative as an Open Source Definition compliant license. Since they own the trademark, they get to define what is and is not Open Source.

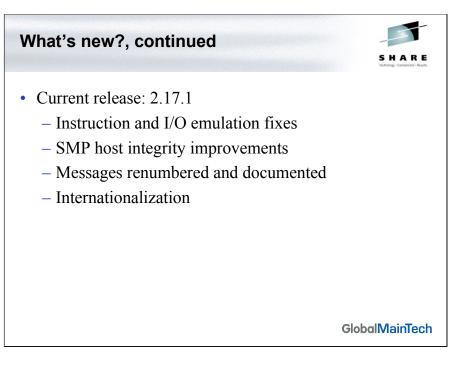


The new DASD subsystem makes better use of CPU and memory. It also allows individual files to be larger than 2 GB if the underlying OS allows it.

Previous versions had to be patched for Mac OS X. This version will run with no modifications. It requires the Jaguar (10.2) release or later.

The networking code has been reworked and improved.

No special options need to be supplied to build the code with any release of gcc from 2.91 to 3.2. Previous versions had some incompatibilities with gcc versions 3.1 and 3.2.

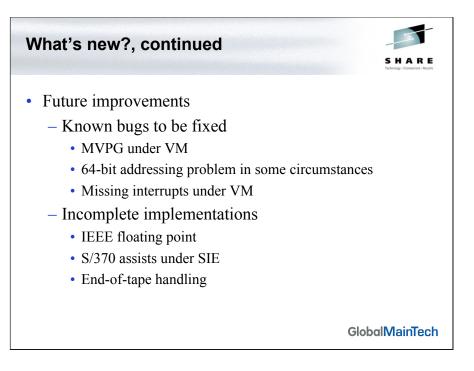


Many instructions have been fixed to work correctly in all known cases, including unusual ones not normally encountered.

The internal locking fixes greatly improve reliability for SMP host systems.

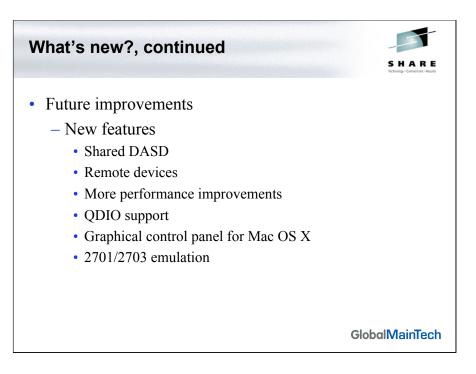
The message documentation is a work in progress; approximately half have been done.

The necessary framework for internationalization support has been completed, and translations are present for Dutch and German.



The known bugs here are not normally encountered in execution, but have been reported and verified.

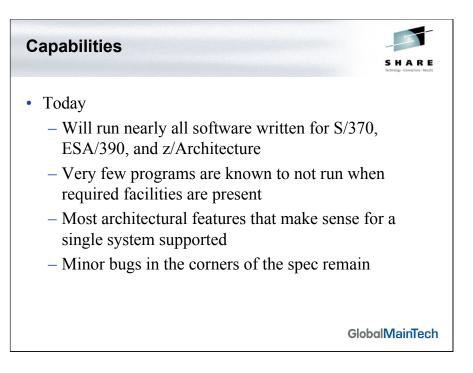
Work is progressing on the IEEE floating point and S/370 assist features. The design of how to handle the end of an emulated tape on write is being discussed in the Hercules developer community.



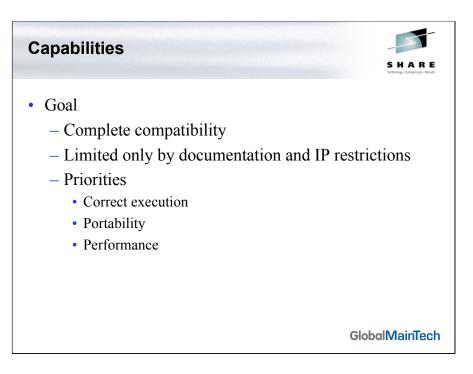
The new DASD subsystem has laid the groundwork for shared DASD, but the actual work has not been done yet. The same code that provides that function will also more than likely allow DASD images to be shared among Hercules images running on different hosts.

There are still performance improvements to be made. Several were originally planned for inclusion in 2.17, but backed out when they were found to cause execution errors.

There are some VMers working on 2701/2703 emulation. This support is currently planned for Hercules-to-Hercules connections only, for RSCS networking.

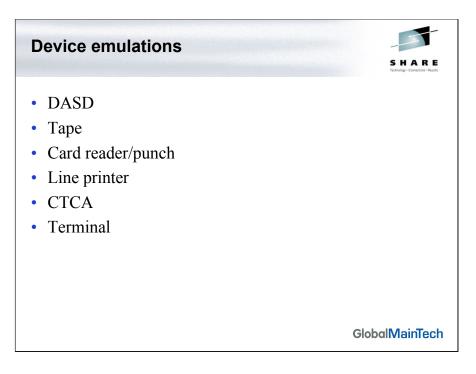


The biggest missing architecture features are the ones that aren't documented in the POO and the like in sufficient detail to implement. Some aren't likely to be implemented; others will be if they can be reverse-engineered. A complete list of features implemented and not is in the Hercules FAQ.

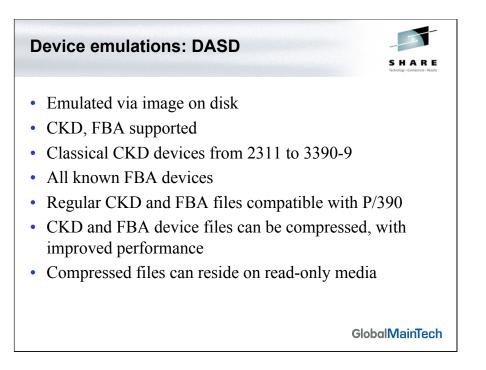


The goal is simple: make Hercules conform to the POO definition of a compatible architecture. If it works differently on Hercules from the real iron, we consider that a bug to be fixed.

Performance is definitely a lower priority than correct execution: getting a wrong answer fast is still getting a wrong answer. Performance will also not be gained at the expense of portability.



Other devices can be implemented, but some (like, say, the 3705) would require a large amount of work, and provide relatively little gain. Others, like the OSA network adapters, would be quite a bit of help to users, but aren't documented well enough to emulate properly.

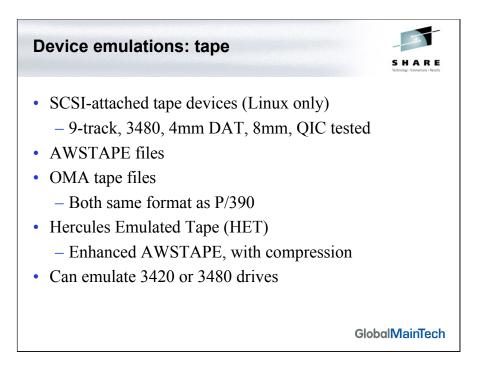


Regular CKD and FBA devices take up as much disk space as the actual device has; an emulated 3350 will take up 300 MB, for example.

Compressed devices can take up much less, and the actual space used depends not only on how well the data compresses, but also how much space is actually in use. Empty tracks take up no space at all. The smaller space used also means less real I/O needs to be done.

Compressed DASD, especially compressed FBA, is comparatively recent, and there may still be some deeply hidden lurking bugs. I recommend keeping good volume backups if you use it. The shadow file facility is also a great help here.

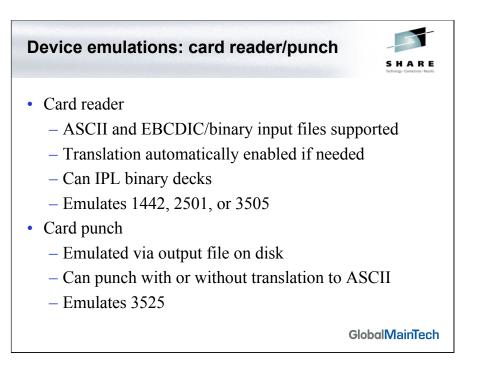
We believe that every DASD model since the 2311 is present. If not, let us know.



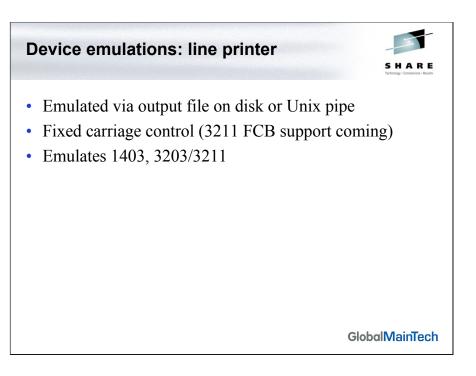
Other SCSI tapes should work, as long as they follow the SCSI tape specification.

SCSI tapes are only supported under Linux. The Cygwin library does not emulate Linux SCSI tape processing. Support under BSD is somewhat limited.

HET is a Hercules-specific format, upwardly compatible with AWSTAPE. It adds compression support, which typically gives 70-80 percent space savings. Utilities are provided to manipulate HETs, including converting to and from AWSTAPE.



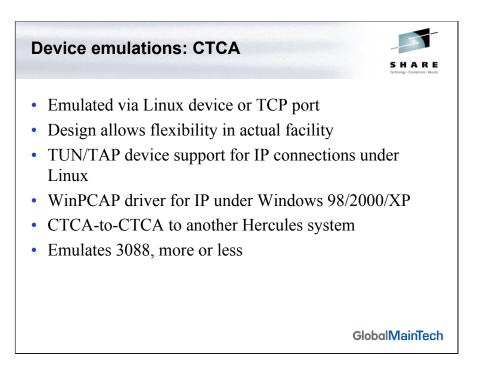
Reader file records don't have to be 80 characters long; they'll be padded if necessary, and an option is provided to truncate long lines if present. Multiple files can be sent to the reader either as separate decks or concatenated into one.



Printed output is usually sent to a file; the ability to send output to a printer directly (via a pipe to an external command) was added in 2.16.

3211 FCB support has been "coming" for two years... hopefully, will happen soon.

It should be possible to emulate a 3800 by generating HP-PCL or PostScript, but nobody's tackled that yet.

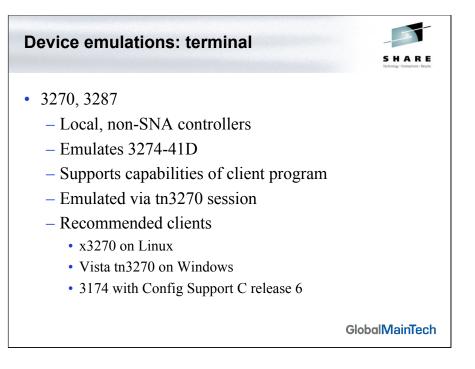


TUN/TAP is a facility provided in the 2.4 series of Linux kernels to allow software to tap directly into the network, below the IP stack. As 2.4 kernels become the standard, this is the recommended and most-used networking method.

The 3088 emulation is now mostly complete, but there are still a few holes, mainly due to a lack of documentation. (Heard that before?)

Under Windows 98, 2000, and XP, David "Fish" Trout has developed a driver to provide CTC networking. This driver is still in development, but the Hercules part of it is complete.

CTCA connections to another Hercules system use simple TCP sockets, and work in all environments.



The actual 3270 emulation provided is limited by the client's capabilities, not Hercules itself. The TN3270 protocol places almost all of the responsibility for interpreting the datastream on the client.

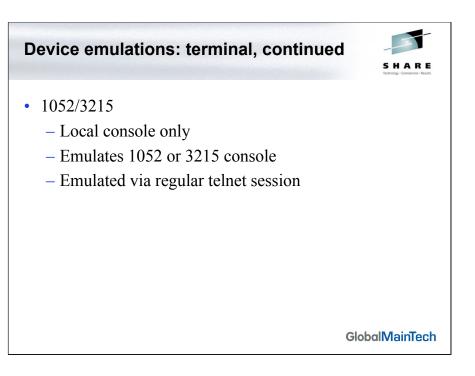
Hercules does not do TN3270E, though this support should be straightforward to add.

The most compatible clients for Hercules are x3270, on Linux, and Tom Brennan's Vista tn3270 on Windows.

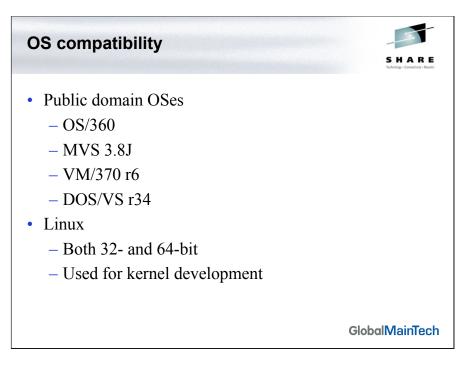
Yes, I've used a 3174-61R with Token Ring adapter and Config C r6 as an MVS console. Neat, in a geeky sort of way.



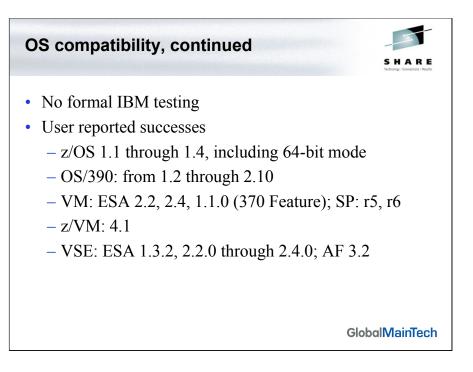
Think I'm kidding about the 3174, huh?



Line-mode terminals, except for local processor consoles, aren't supported because it would require going through the hoops to emulate a 270x or 37x5, a major undertaking for a small gain. If the 2701/2703 emulation comes to pass, it might be possible to add line-mode terminal emulation to that code.

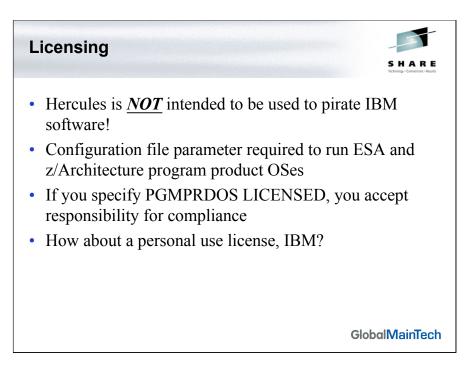


Hercules is in use for development of Linux/390 and Linux for the z/Series, since it allows developers to participate in work on those systems without access to an actual mainframe.



Hercules has not been certified officially as a plug-compatible system, mainly because that takes real money. Because of that, we can't officially claim that newer stuff will run.

That said, users have reported running these OSes with Hercules.



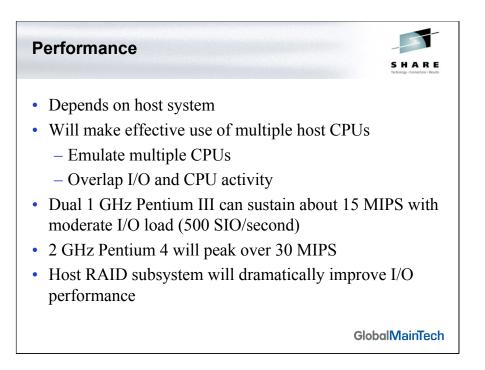
Hercules acts as an IFL (Integrated Facility for Linux) processor by default. The user can override this and make it appear as a regular CPU by specifying PGMPRDOS LICENSED in the config file. This will produce a warning message at Hercules startup:

```
HHCCF039W PGMPRDOS LICENSED specified.
Licensed program product operating systems are enabled.
You are responsible for meeting all conditions of your
software license.
```

This means you! The developers of Hercules are not responsible if you choose to violate the terms of any software license you may be subject to.

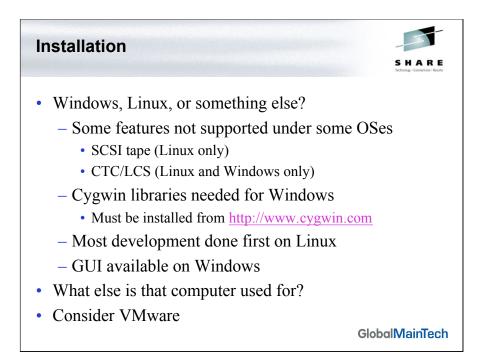
If you try to IPL OS/390 from 2.6 on, or z/OS, you'll get a wait state A7A with reason code 7 if you don't have that parameter specified. Some versions of VM/ESA will enter a 9030 wait state; others will run, but only allow Linux guests.

Hercules could be a real asset to IBM, giving the kinds of people who learn about systems by doing a way to get their hands on current technology. I hope they'll see the light.



The biggest boost in Hercules performance comes from using a multiprocessor host system. Hercules overlaps I/O and CPU by using separate threads of execution for I/O and CPU activity. These threads are dispatched on any available CPU in an SMP host.

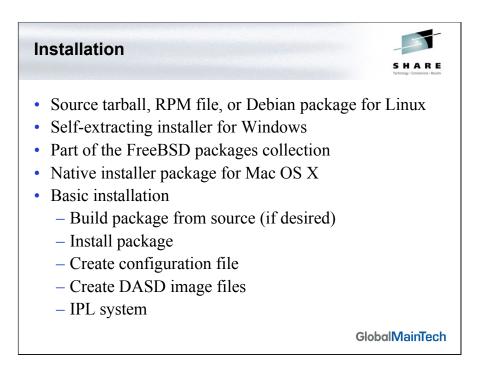
The next biggest boost comes from speeding up disk I/O. As with real hosts, the system spends a lot of time waiting on I/O, especially when one emulated I/O can take several real ones. Hardware RAID can overlap those I/Os, and caching can save I/Os entirely - a real win, just as with a mainframe.



Hercules was originally designed to make use of as much of the host system as it could. It shares its host fairly well, but it's happiest if it has a system all to itself. For best performance and ease of use, it may well be worth it to spend a few hundred bucks on a host to dedicate to it, especially if you're looking to run an older OS, where a very modest host will provide entirely acceptable performance. New 1 GHz Celeron/Duronclass systems can be had for well under \$500.

Due to the GPL license applied to the Cygwin base library, I cannot distribute it from the Hercules page without also making the version used for that release available in source form. Since I keep old releases of Hercules around, and since there are compatibility issues among Cygwin versions, this would cause consumption of large amounts of disk space for no good purpose.

If you run Hercules on Windows, I highly recommend checking out Fish's Hercules Page, where you can get his GUI for Windows, along with other useful utilities and mods. The URL is on the last page.



The host system architecture is detected automatically by the autoconf configuration system.

To compile on Linux: explode the source tarball with

```
tar xzvf hercules-version.tar.gz
```

and then

```
cd hercules-version
```

```
./configure
```

make

To install the compiled version: as the root user,

make install

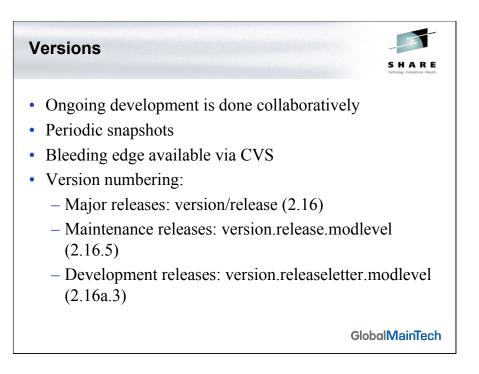
To install a packaged RPM:

```
rpm -Uvh hercules-version.architecture.rpm
```

where *version* is the version of Hercules, and *architecture* is the host system's CPU architecture (i386 for Intel systems).

On Debian:

```
apt-get install hercules
```

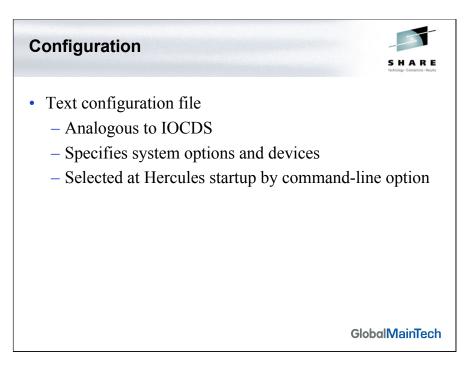


The Hercules development team uses the Concurrent Version System (CVS) to manage its source tree. It allows many developers spread out all over the world to share one view of the source, and to update that view in a consistent manner.

Anyone can get a copy of the current source tree via CVS. See http://www.cvshome.org for full details on CVS. The CVSROOT for the Hercules tree is:

:pserver:anonymous@cvs.conmicro.cx:/usr/cvs/hercules

Note that there is **no** connection between maintenance releases and development snapshots. Version 2.16.5 and 2.16a.5 are not connected at all except that they're both based on 2.16.

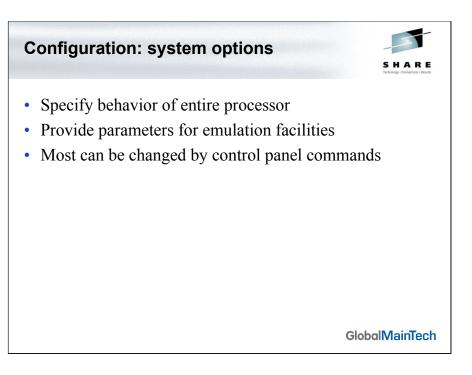


Hercules is started by:

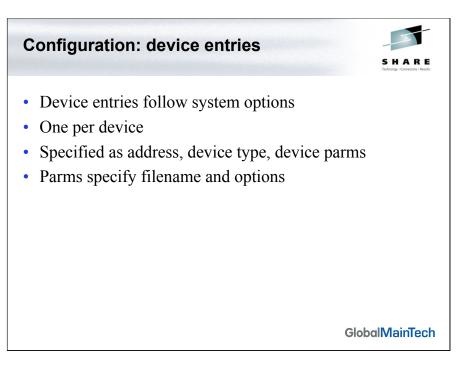
```
hercules -f config-file
```

You can have multiple configurations available by having more than one configuration file, but a single copy of Hercules can only use one configuration file. You can dynamically add and delete devices during execution, but some system-wide configuration parameters can only be changed by stopping and restarting Hercules.

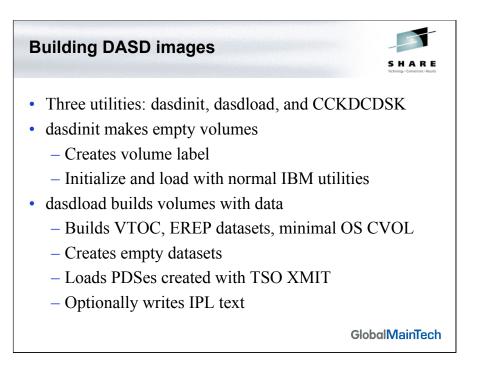
```
# Sample configuration file
CPUSERIAL 000611
CPUMODEL 3090
MAINSIZE 64
XPNDSIZE 0
CNSLPORT 3270
NUMCPU
         1
NUMVEC
         1
LOADPARM 0120....
OSTAILOR OS/390
SYSEPOCH 1900
TZOFFSET -0500
PANRATE FAST
ARCHMODE ESA/390
#
# Device definitions
000A 1442 adrdmprs.rdr
000C 3505 jcl.txt ascii trunc
000D 3525 pch00d.txt ascii
000E 1403 prt00e.txt
001F 3270
             mvsv5r.120
0120
       3380
0580 3420 ickdsf.ipl
```



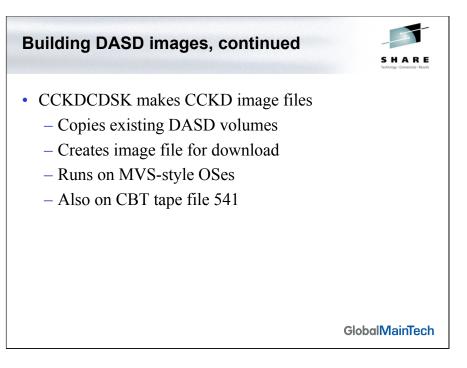
A more detailed discussion of the system parameters is in the appendix at the end of this handout.



Devices can be specified in any order. I recommend putting card readers and tape drives first, then the IPL disk, then any other devices, as this makes controlling them easier in the graphical control panel.



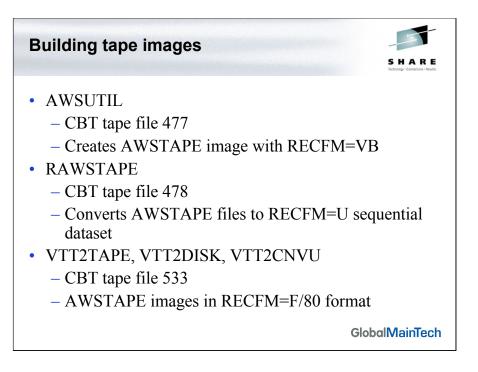
When creating a volume with dasdinit, include alternate tracks for any volume to be formatted with ICKDSF or IBCDASDI. These programs will seek to the alternate cylinders even if they don't intend to write there, just to make sure they're accessible.



Unlike the other utilities, CCKDCDSK runs on MVS. It reads a DASD volume and creates a CCKD image file, which you can transfer to your PC (via binary FTP or something similar) and point Hercules at directly.

CCKDCDSK is provided as a TSO TRANSMIT file. Upload it directly, again via binary FTP or the like, and then use TSO/s RECEIVE INDATASET() command to place the contents in a PDS.

The CCKDCDSK program must be installed in an APF-authorized library.

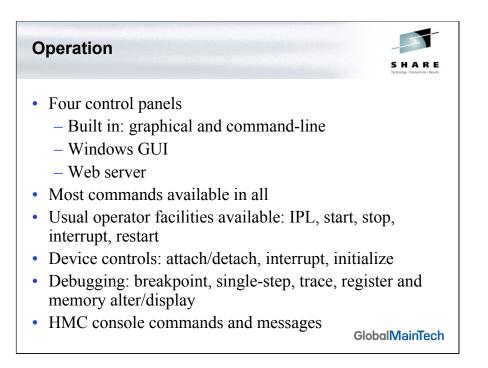


Brandon Hill's AWSUTIL was the first widely available program to deal with AWSTAPE images on MVS. It reads a real tape and creates an AWSTAPE image file.

Jan Jaeger's RAWSTAPE is intended to go the other direction: from an AWSTAPE image to a real tape. You can also write to DASD if you like.

Sam Golob's VTT utilities are part of a larger Virtual Tape Transportation System.

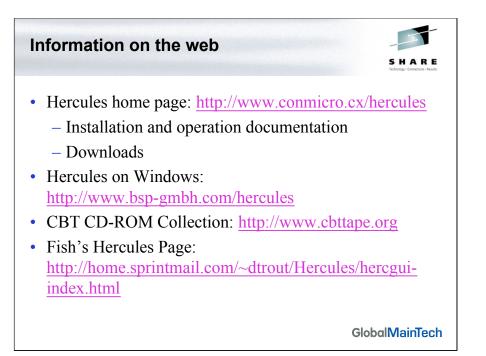
Which set of utilities to use is largely a matter of taste.

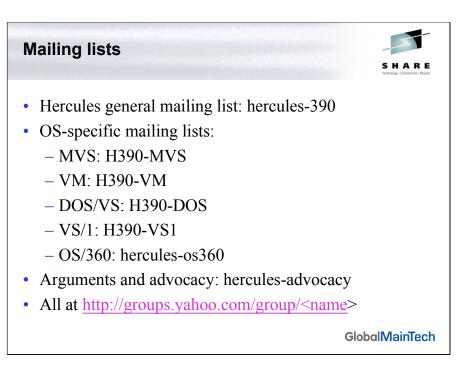


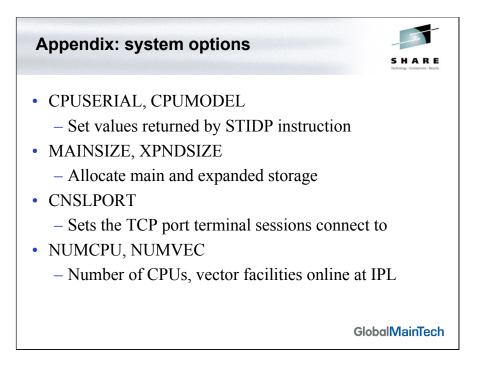
"Graphical" is a bit of a misnomer; it's not really graphical, just laid out on one screen in an easy-to-use manner.

Device initialization is also used to mount tapes and card decks on emulated devices during operation.

When devices are added to the configuration via attach, or removed via detach, the OS is notified as well; OS/390 will respond with the IOS150I DEVICE *nnnn* NOW AVAILABLE FOR USE message, just as it does on the real iron.





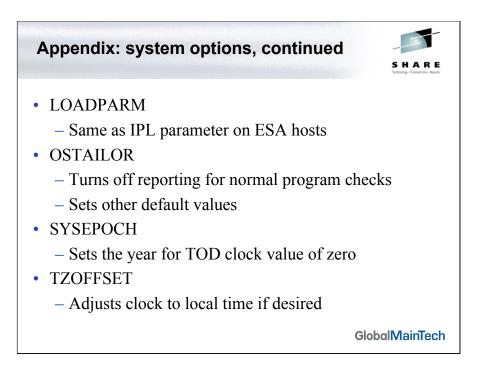


The CPUSERIAL and CPUMODEL parameters don't change anything in the emulator's behavior beyond the results of the STIDP (Store CPU ID) instruction; in particular, no attempt is made to implement any model's specific behavior.

MAINSIZE and XPNDSIZE allocate that much memory at Hercules startup and hold it as long as Hercules is up. Don't allocate more than your machine's physical RAM and swap space, and if possible don't allocate more than physical RAM for best performance. If you have to page, let Linux or Windows do the paging so page I/Os don't have to be emulated. XPNDSIZE is only valid if ARCHMODE ESA/390 is specified.

The same TCP port is used for both 3215 and 3270 connections. Connecting with a regular telnet client will connect to a 3215 device, and connecting with a tn3270 client will connect to a 3270 device.

The total number of CPUs and vector facilities available in the configuration is set at compile time. NUMCPU and NUMVEC set the number online at IPL. Use the facilities of the OS (such as CF CPU(x), ON in MVS) to bring others online as needed.

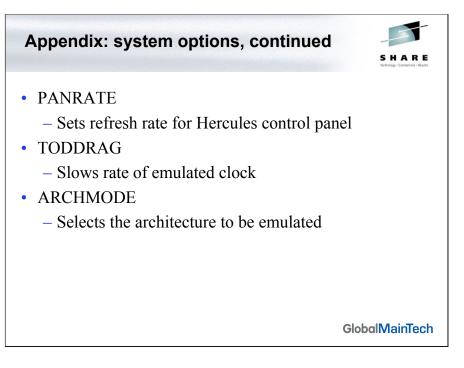


Without OSTAILOR, for example, VM will flood the Hercules console with messages about privileged operations exceptions, and OS/390 and Linux will do the same for other program checks. These are perfectly normal during operation for those OSes, and reporting them just wastes time and CPU cycles. OSTAILOR turns them off.

SYSEPOCH is intended to make older, non-Y2K OSes work properly by fooling the TOD clock. For OS/360, SYSEPOCH 1988 is recommended to set the emulated TOD clock to 1975 for dates in 2003, since OS/360 thinks dates beyond 1999 are invalid, and the calendars match between 2003 and 1975. For MVS 3.8 and VM/370, SYSEPOCH 1928 is recommended for the same reason. Only 1900, 1928, 1960, 1988, and 1970 are valid.

TZOFFSET is included in case you want to run your TOD clock on local time instead of GMT.

Both SYSEPOCH and TZOFFSET are present because the emulated TOD clock cannot be set directly; they're derived from the host system clock.

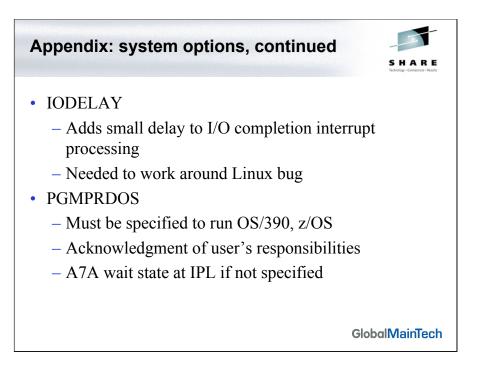


PANRATE FAST will make the control panel act more like a real mainframe by updating it every 50 milliseconds instead of every 500. This will cost some CPU speed, though. You can also specify a refresh interval in milliseconds directly.

TODDRAG slows down the emulated TOD clock by the specified factor. TODDRAG 2 will make the emulated TOD clock count one minute for every two minutes of real time. This helps with OS/390 performance on slower (below PIII-500 performance) host systems. TODDRAG 1 specifies no slowdown.

ARCHMODE can be one of S/370, ESA/390, or ESAME. ESAME was the original name for z/Architecture.

DEVTMAX defaults to 8 on Windows, and 0 on other platforms. 0 is no maximum; -1 causes a new thread to be launched for every I/O. Otherwise, the thread is kept until it has been idle for 5 minutes.



Linux/390 and Linux for z/Series, with kernels at least through version 2.4.17, cannot handle an I/O interrupt arriving immediately after the I/O is scheduled. IODELAY was added to work around this problem. A fix has been identified, but has not been added to the Linux source tree yet. IODELAY defaults to 0 unless OSTAILOR LINUX has been specified; if so, the default is 800.

Hercules acts as an IFL (Integrated Facility for Linux) processor by default. The user can override this and make it appear as a regular CPU by specifying PGMPRDOS LICENSED in the config file. This will produce a warning message at Hercules startup:

```
HHCCF039W PGMPRDOS LICENSED specified.
Licensed program product operating systems are enables.
You are responsible for meeting all conditions of your
software license.
```

This means you! The developers of Hercules are not responsible if you choose to violate the terms of any software license you may be subject to.