z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES 11

A cookbook for installing and customizing z/VM 6.1 and Linux SLES 11 on the mainframe

Marian Gasparovic
Michael MacIsaac
## Contents

**Preface** ................................................................. ix
   Chapters and Appendices ............................................. ix
   History ........................................................................ x
**Summary of changes in the February 2010 version** ...................... xi
**Conventions** ................................................................ x
**The team that wrote this book** ....................................... xii
**Comments welcome** .................................................... xii

**Chapter 1. Introduction to z/VM and Linux** .......................... 1
   1.1 What is virtualization? ............................................. 3
   1.2 A philosophy adopted in this book ......................... 3
   1.3 Choices and decisions made in this book ................ 3
   1.4 Infrastructure design ............................................. 4
   1.5 Usability tests performed for this book .................. 5

**Chapter 2. Planning** ..................................................... 7
   2.1 Bill of materials ..................................................... 7
      2.1.1 Hardware resources ........................................ 7
      2.1.2 Software resources ......................................... 8
      2.1.3 Networking resources ..................................... 8
   2.2 z/VM conventions .................................................. 9
      2.2.1 Volume labeling convention ............................. 9
      2.2.2 Backup file naming convention ....................... 9
      2.2.3 The command retrieve convention .................. 10
   2.3 Disk planning ........................................................ 10
   2.4 Memory planning ................................................... 11
   2.5 Password planning ................................................ 12
   2.6 Planning worksheets .............................................. 13
      2.6.1 z/VM resources used in this book .................. 13
      2.6.2 z/VM DASD used in this book ....................... 14
      2.6.3 Linux resources used in this book .................. 14
      2.6.4 Linux user IDs used in this book .................. 15
   2.7 Blank worksheets .................................................. 16
      2.7.1 z/VM resources worksheet ............................. 16
      2.7.2 z/VM DASD worksheet .................................. 17
      2.7.3 Linux resources worksheet ....................... 18
      2.7.4 Linux user ID worksheet ............................. 18

**Chapter 3. Configuring a desktop machine** ......................... 19
   3.1 PuTTY: a free SSH client for Windows .................... 19
   3.2 Setting up a VNC client ......................................... 23
      3.2.1 Downloading and running RealVNC ................ 23
      3.2.2 Customizing RealVNC .................................. 24
   3.3 3270 emulators ..................................................... 25

**Chapter 4. Installing and configuring z/VM** ......................... 27
   4.1 Installing z/VM from DVD ...................................... 27
      4.1.1 Booting z/VM from DVD ............................... 28
      4.1.2 Copying a vanilla z/VM system to DASD .......... 31
Chapter 5. Servicing z/VM

5.1 Applying a PTF

5.2 Applying a Recommended Service Upgrade or RSU

4.1 Completing the z/VM installation

4.2 Configuring TCP/IP

4.3 Configuring the XEDIT profile

4.4 Customizing the SYSTEM CONFIG file

4.5 Configuring TCP/IP to start at IPL time

4.6 Adding paging volumes

4.7 Creating a user ID for common files

4.8 Customizing system startup and shutdown

4.9 Addressing z/VM security issues

4.10 Backing up your z/VM system to tape

4.11 Relabeling system volumes

4.12 Restoring your z/VM system from tape

5.1 Applying a PTF

5.2 Applying a Recommended Service Upgrade or RSU

5.2.1 Getting service from the Internet

5.2.2 Downloading the service files

5.2.3 Creating a new MAINT minidisk

5.2.4 Receiving, applying, and building the service

5.2.5 Putting the service into production

5.1.1 Getting service using ShopzSeries

5.1.2 Determining if a PTF has been applied

5.1.3 IPL the vanilla z/VM from DASD

5.1.4 Completing the z/VM installation

5.2.6 Checking for APARMEMO files

5.3 Configuring the FTP server

5.4 Shutting down and reIPLing the system

5.5.1 Renaming the TCPIP configuration file

5.5.2 Copy the PROFILE XEDIT file

5.5.3 Configuring the FTP server

5.5.4 Checking for APARMEMO files

5.6.1 Formatting the paging volumes

5.6.2 Formatting DASD for minidisks

5.6.3 Updating the SYSTEM CONFIG file

5.6.4 Testing the changes

5.7.1 Define the user in the USER DIRECT file

5.7.2 Logging and customizing the new user ID

5.7.3 Copying a PROFILE XEDIT

5.7.4 Creating a PROFILE EXEC

5.7.5 Copying files associated with this book to LNXMAINT

5.8.1 Configuring the AUTOLOG1 PROFILE EXEC

5.8.2 Testing the changes

5.9.1 Changing passwords in USER DIRECT

5.10.1 IPL the vanilla z/VM from DASD

5.10.3 IPL the vanilla z/VM from DASD

5.10.5 IPL the vanilla z/VM from DASD

5.10.6 IPL the vanilla z/VM from DASD

5.11.1 Modifying labels in the SYSTEM CONFIG file

5.11.2 Modifying labels in the USER DIRECT file

5.11.3 Changing the labels on the five volumes

5.11.4 Shutting down your system and restarting it

5.12.1 Completing the z/VM installation
Chapter 9. Cloning SLES 11 .......................................................... 145
 9.1 Formatting DASD for minidisks ............................................ 145
 9.2 Defining a new user ID for a virtual server ......................... 146
    9.2.1 Adding LINUX01 to AUTOLOG1’s PROFILE EXEC ............. 147
    9.2.2 Creating a parameter file for the new LINUX ID ............ 148
 9.3 Cloning a virtual server manually ...................................... 149
 9.4 Cloning a virtual server automatically ............................... 154
 9.5 Creating three more virtual servers .................................. 156
    9.5.1 Defining three more user IDs ................................. 156
    9.5.2 Creating three new parameter files ......................... 157
    9.5.3 Granting user IDs access to VSITCH ....................... 158
    9.5.4 Testing logging on to a new user ID ....................... 159
 9.6 Reviewing system status .................................................. 160

Chapter 10. Cloning open source virtual servers ......................... 163
 10.1 Creating a virtual Web server ......................................... 163
    10.1.1 Installing Apache RPMs .................................... 163
    10.1.2 Testing Apache ............................................. 164
    10.1.3 Populating your Web site ................................... 164
    10.1.4 Apache resources .......................................... 164
 10.2 Creating a virtual LDAP server ....................................... 164
    10.2.1 Cloning a Linux ............................................. 165
    10.2.2 Configuring the LDAP server ................................ 165
    10.2.3 Adding an LDAP user ....................................... 167
    10.2.4 Setting another virtual server to use the LDAP server .... 168
 10.3 Creating a virtual file and print server ............................ 171
    10.3.1 Cloning a Linux virtual server ............................ 171
    10.3.2 Installing necessary RPMs ................................ 171
    10.3.3 Configuring Samba configuration file ...................... 172
    10.3.4 Adding a Samba user ...................................... 172
    10.3.5 Starting Samba at boot time ................................ 173
    10.3.6 Testing your changes ...................................... 173
    10.3.7 Configuring printing ....................................... 175
 10.4 Creating a virtual application development server .............. 175
    10.4.1 Cloning a Linux virtual server ............................ 175
    10.4.2 Scripting Languages ...................................... 175
    10.4.3 C/C++ development tools .................................. 176
    10.4.4 Java development tools ................................... 177
    10.4.5 Additional resources ...................................... 178

Chapter 11. Miscellaneous recipes .......................................... 181
 11.1 Adding DASD ......................................................... 181
    11.1.1 Adding minidisks to LINUX02 ............................. 181
    11.1.2 Making the new minidisks available ...................... 182
    11.1.3 Formatting and partitioning the minidisks ............... 182
 11.2 Adding a logical volume .............................................. 183
    11.2.1 Creating a logical volume and file system ............... 183
    11.2.2 Updating the file system table .......................... 186
 11.3 Extending an existing logical volume .............................. 187
 11.4 The X Window System ............................................... 189
    11.4.1 VNC Server ................................................ 190
    11.4.2 X Server on workstation .................................. 191
 11.5 Centralizing home directories for LDAP users .................... 194
    11.5.1 Recommendations for centralizing home directories .... 194
<table>
<thead>
<tr>
<th>Chapter 12. Monitoring z/VM and Linux</th>
<th>197</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.1 Using INDICATE and other commands</td>
<td>197</td>
</tr>
<tr>
<td>12.1.1 Using the INDICATE command</td>
<td>197</td>
</tr>
<tr>
<td>12.1.2 Using other basic commands</td>
<td>200</td>
</tr>
<tr>
<td>12.2 The z/VM Performance Toolkit</td>
<td>201</td>
</tr>
<tr>
<td>12.2.1 Configuring the z/VM Performance Toolkit</td>
<td>202</td>
</tr>
<tr>
<td>12.2.2 Configuring Web Browser support</td>
<td>203</td>
</tr>
<tr>
<td>12.2.3 Configuring PERFSVM</td>
<td>203</td>
</tr>
<tr>
<td>12.2.4 Increasing the size of the MONDCSS DCSS</td>
<td>205</td>
</tr>
<tr>
<td>12.2.5 Starting the z/VM Performance Toolkit</td>
<td>206</td>
</tr>
<tr>
<td>12.2.6 Using the z/VM Performance Toolkit</td>
<td>206</td>
</tr>
<tr>
<td>12.3 Monitoring Linux</td>
<td>209</td>
</tr>
<tr>
<td>12.3.1 Monitoring Linux with rmfpms</td>
<td>210</td>
</tr>
<tr>
<td>12.3.2 Monitoring Linux performance data from the kernel</td>
<td>211</td>
</tr>
<tr>
<td>12.4 Viewing Linux data in the Performance Toolkit</td>
<td>212</td>
</tr>
<tr>
<td>Appendix A. Reference Material</td>
<td>215</td>
</tr>
<tr>
<td>A.1 Related books</td>
<td>215</td>
</tr>
<tr>
<td>A.2 Online resources</td>
<td>216</td>
</tr>
<tr>
<td>A.3 Important z/VM files</td>
<td>216</td>
</tr>
<tr>
<td>A.4 Cheat sheets</td>
<td>216</td>
</tr>
<tr>
<td>A.4.1 XEDIT cheat sheet</td>
<td>217</td>
</tr>
<tr>
<td>A.4.2 vi cheat sheet</td>
<td>217</td>
</tr>
<tr>
<td>Appendix B. Source code</td>
<td>219</td>
</tr>
<tr>
<td>B.1 Obtaining and using the Web material</td>
<td>219</td>
</tr>
<tr>
<td>B.2 z/VM REXX EXECs and XEDIT macros</td>
<td>219</td>
</tr>
<tr>
<td>B.2.1 The CPFORMAT EXEC</td>
<td>219</td>
</tr>
<tr>
<td>B.2.2 The CHPW610 XEDIT macro</td>
<td>220</td>
</tr>
<tr>
<td>B.2.3 PROFILE EXEC for Linux user IDs</td>
<td>223</td>
</tr>
<tr>
<td>B.3 Linux code</td>
<td>223</td>
</tr>
<tr>
<td>Index</td>
<td>231</td>
</tr>
</tbody>
</table>
Preface

“The search for truth is more precious than its possession.”
— Albert Einstein

This book describes how to 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 6.1 media, and a Linux distribution.

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.

Chapters and Appendices

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

► Chapter 2, “Planning” on page 7 describes how to plan hardware, software and networking resources. It discusses DASD labeling conventions used in the book and 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 19 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 27 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 67 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 a NFS server for SLES 11” 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 SLES11 on the cloner” on page 89, describes how to install and configure the first two Linux images onto the first Linux user ID: the golden image, which it is cloned from, and the controller, which does the cloning among other tasks.

► Chapter 8, “Installing SLES11 on the golden image” on page 117, illustrates how to move the Linux install tree from the Linux PC server to the controller under z/VM.

► Chapter 9, “Cloning SLES 11” on page 145 explains how to prepare z/VM user IDs and clone your first virtual server.

► Chapter 10, “Cloning open source virtual servers” on page 163, 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, “Miscellaneous recipes” on page 181 describes how to add a logical volume and how to extend it if more space is needed.
▷ Chapter 12, “Monitoring z/VM and Linux” on page 197, describes some basic steps to begin monitoring z/VM and your new Linux virtual servers.
▷ Appendix A, “Reference Material” on page 215, provides references Web sites, books and other pertinent information.
▷ Appendix B, “Source code” on page 219 lists the z/VM and Linux source code associated with this book.

History

There have been quite a number of copies of Virtualization Cookbooks, so a short history is provided:

2010
In February of 2010 this book was published.
There are REXX EXECs and Linux scripts associated with this book to help with the installation and configuration of z/VM and Linux. These tools are not IBM products nor formally supported. However, they will be available on the Web and informally supported.

2008
In August 2008, two IBM Redbooks were published, one targeting Novell SLES 10 SP2 distribution and the other RHEL 5.2:
▷ This book: z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES 10 SP2, SG24-7493
▷ z/VM and Linux on IBM System z: The Virtualization Cookbook for RHEL 5.2, SG24-7492

2007
In March 2007, two books were published on linuxvm.org/present, each book targeting a different distribution:
▷ z/VM and Linux on IBM System z: The Virtualization Cookbook for SLES 10
▷ z/VM and Linux on IBM System z: The Virtualization Cookbook for RHEL 5

2006
In September of 2006 a Redbook was published that addressed both 31-bit and 64-bit RHEL 4:
▷ IBM z/VM and Linux on IBM System z: Virtualization Cookbook for Red Hat Enterprise Linux 4, SG24-7272

In August of 2006 a book was published on linuxvm.org/present that addressed both 31-bit and 64-bit SLES 9:
▷ z/VM and Linux on IBM System z: The Virtualization Cookbook 2
Summary of changes in the February 2010 version

There are both significant and smaller changes in this book:

- The z/VM sections are updated for V6.1.
- The Linux sections are updated for SLES 11.
- The REXX EXEC and XEDIT macro to help modify z/VM system labels have been removed as it is recommended to change the system labels at z/VM install time. For reference, the section on relabeling system labels is slightly modified to describe performing the steps manually, without the REXX EXEC and XEDIT macro.
- The Servicing z/VM chapter has been updated for z/VM 6.1 and now describes IBM ShopzSeries, not IBMLink.
- The Linux system that does the cloning is called the cloner, not the controller.
- A section on the X Window System has been added: see “The X Window System” on page 189.

Some sections are removed in the release of this book:

- In Miscellaneous Recipes chapter, sections on LDAP, NTP, rsync and CMM were removed in the interests of time.
- Details on creating a travelling /home/ have been removed, however, a section based on a linux-390 append by Patrick Spinler has been added: see “Recommendations for centralizing home directories” on page 194.

Conventions

The following font conventions are used in this book:

<table>
<thead>
<tr>
<th>Monospace and bold</th>
<th>Commands entered by the user on the command line</th>
</tr>
</thead>
<tbody>
<tr>
<td>MONOSPACEx, bold, italics</td>
<td>Values in italics and bold are example values to be replaced with values correct for your enterprise.</td>
</tr>
<tr>
<td>monospace</td>
<td>File, directories, user ID and minidisk names</td>
</tr>
</tbody>
</table>

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 running as non-root are usually prefixed with $
The team that wrote this book

This book was updated by Marian Gasparovic and Mike MacIsaac, both of IBM.

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

Bill Bitner, Jim Switzer, Tami-Zebrowski Darrow
IBM Endicott

Jeng-Fang Wang
IBM Poughkeepsie

Martin Schwiedfsky
IBM Boeblingen

Patrick Spinler
Mayo Clinic

Mark Post
Novell

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 and Roy Costa from IBM Poughkeepsie for always finding another free LPAR and more DASD.

Comments welcome

Your comments are important to us. You can send comments or questions directly to the authors:

marian.gasparovic at sk.ibm.com
mikemac at us.ibm.com
Introduction to z/VM and Linux

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

Virtualization is hot in the IT industry. The IBM mainframe, z/VM and its predecessors have been doing virtualization for five decades. Today, it is the most functionally rich virtualization platform available. When Linux came to the IBM 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 in production mode.

With a z/VM and Linux infrastructure, you can reduce the time between deciding on the acquisition of new servers and then implementing them because new servers can be deployed in a matter of minutes. This powerful build and clone capability can enable you to launch new products and services without the exhaustive planning, purchasing, installing and configuring new hardware and software that can be associated with conventional discrete hardware servers. Development groups who need test environments built and rebuilt rapidly to enable them to efficiently deliver their projects in good time, handling change management in the process can also benefit from this unique advantage.

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

- Their virtualization capabilities are more mature and robust than any other hardware and hypervisor combination.
- z/VM provides a rich, functional and sophisticated level of systems management which can greatly benefit running large numbers of Linux servers.
- z/VM’s virtual switch (VSWITCH) makes 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 at the base installation level. There is only a relatively small number of configuration files. Once it is set up, z/VM will run for months with little maintenance and administration required.

Much function has been added to z/VM since version 5.2.
**z/VM 5.3**

z/VM 5.3 became generally available in June of 2007. Scalability was extended to allow 256 GB of real memory, a total of 8 TB of virtual storage, and 32 real processors. z/VM V5.3 also added support for the Collaborative Memory Management Assist (CMMA) on the z9 EC and the z9 BC processors. Virtual Machine Resource Manager (VMRM) detects when memory is constrained and notifies the Linux guests, which can then adjust their memory consumption to help relieve the memory constraint. In the previous major release, z/VM 5.2, many memory contention issues were removed 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 2 GB 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.

**z/VM 5.4**

z/VM 5.4, available in August of 2008, provides major improvements when operating on System z servers with large memory configurations. It improves scalability and can help support increased workloads on IBM System z servers. This release exploits new capabilities of the System z10 including:

- Greater flexibility, with support for the new z/VM-mode logical partitions, allowing all System z processor-types (CPs, IFLs, zIIPs, zAAPs, and ICFs) to be defined in the same z/VM LPAR for use by various guest operating systems
- Capability to install Linux on System z from the HMC that eliminates network setup or a connection between an LPAR and the HMC
- Enhanced physical connectivity by exploiting all OSA-Express3 ports, helping service the network and reducing the number of required resources

z/VM 5.4 dynamic memory upgrade support allows real memory to be added to a running z/VM system, avoiding the need to shut down z/VM and its guests, deactivate the LPAR, change its memory allocation, reactivate the LPAR, re-IPL z/VM, and restart its guests. Memory can be added non-disruptively to individual guests that support the dynamic memory reconfiguration architecture.

**z/VM 6.1**

z/VM 6.1, available in October of 2009 is intended to be the base for all future z/VM enhancements. This release implements a new Architecture Level Set (ALS) available only on the IBM System z10 Enterprise Class server and System z10 Business Class server and future generations of System z® servers. Requiring z10 technology or later allows z/VM to take advantage of newer hardware technology for future exploitation.

Enhancements in z/VM V6.1 provide:

- Enhanced performance of virtual networking environments running heavy guest-to-guest streaming workloads
- Faster access to data when utilizing FICON Express8
- Closer integration with IBM Systems Director to eliminate the need to download agents and help simplify the installation of those agents
- Significantly better and more highly secure guest transactions when using Crypto Express3 as compared to Crypto Express2
- Guest support for IBM System Storage DS8000 Extended Address Volumes (EAVs) to help simplify storage management and relieve address constraints
Read more about System z virtualization capabilities on the Web at:

http://www.vm.ibm.com

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 (sometimes called 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 z/VM native one, the Conversational Monitoring System (CMS) and Linux. CMS can be thought of as a z/VM shell. Virtual machines running Linux as guests of a z/VM host 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. This book will try to follow the quote from Albert Einstein at the start of this chapter: Everything should be made as simple as possible, but not simpler.

A lot of books and papers are talking about virtualization today, but not telling you how to do it. The remainder of this book gives you the HOWTO that backup these marketing words.

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 choices and assumptions made in this book:

- Use of a Cloning product versus “roll your own” cloning: Cloning products, such as Aduva’s Onstage, Mainstar’s Provisioning Expert, IBM Tivoli Provisioning Manager and IBM Systems Director 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 clone script and z/VM configuration described 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 CA’s 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, and 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 resources defined.

- Shared read-only 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. See:
http://www.vm.ibm.com/linux/dcss/

- Conventional 3390 ECKD DASD versus FBA disks accessed using SCSI over FCP: The System z server has traditionally only supported 3390 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: Two methods of cloning are described: manually and using a Linux bash script. The manual method is described so you can better learn the concepts. The Linux script is described and provided so you can save time.

- Use of VNC server for GUI based tools: Some programs, such as installation tasks, require a functioning GUI. Because a GUI is only occasionally needed, it is reasonable to have a lightweight environment in place. Complete desktops such as Gnome or KDE are not just overkill for System z, they are a bad idea in virtualized environment because they are costly in terms of resources. For this reason a VNC server is described as the recommended GUI. To read more, see section 11.4, “The X Window System” on page 189.

### 1.4 Infrastructure design

To install and configure z/VM, install, configure and 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 4 shows a block diagram of a System z10 with many LPARs. z/VM 6.1 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.

<table>
<thead>
<tr>
<th>LPAR</th>
<th>Resources:</th>
</tr>
</thead>
<tbody>
<tr>
<td>L P A R</td>
<td>CPU: 2 IFLs, shared</td>
</tr>
<tr>
<td></td>
<td>Memory: 3GB/1GB or more</td>
</tr>
<tr>
<td>L P A R</td>
<td>Disk: 25 3390-3 DASD</td>
</tr>
<tr>
<td>L P A R</td>
<td>Network: 16 OSA-E addresses</td>
</tr>
<tr>
<td>L P A R</td>
<td>TCP/IP 6 TCP/IP addresses</td>
</tr>
</tbody>
</table>

**OSA Express**

**Desktop machine**

**OSA Express**

**PC Linux NFS server**

**Desktop machine**

![Figure 1-1: System infrastructure and z/VM user IDs](image)
The user IDs that are described in this book have the following functions:

- **LNXMAINT**: A user ID on which to store files that will be used by both CMS and Linux
- **S11CLONE**: The cloner image that does the cloning. It also serves as the Linux install server
- **LINUX01-04**: The Linux virtual servers described in the *Cloning open source virtual servers* chapter. Each virtual server is configured with a two 3390-3 minidisks to allow for slightly more than 4 GB of disk space.
- **S11GOLD**: The SLES 11 golden image. This is the Linux system that is cloned.

### 1.5 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 67, 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 and you work hard.
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 describes conventions adopted for labeling 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 System z project can be divided into:

- Hardware
- Software
- Networking

2.1.1 Hardware resources

The following hardware is needed:

- A System z logical partition (LPAR); on a System z10 (z/VM 6.1 will not run on earlier hardware)
  - Processors or CPUs: One IFL (or CP) minimum, two or more are strongly 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 for relatively small systems such as in these example sizes. See the following Web site for a discussion of how to apportion memory:
– DASD: 24 3390-3s or 8 3390-9s at a minimum
– Open Systems Adapter (OSA) network cards: One card minimum with 8 device numbers (technically 6, 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. Setting up a Linux PC or UNIX server is described.
   ▶ A workstation or desktop that has network access to the mainframe

2.1.2 Software resources

The following software resources are needed:
▶ z/VM 6.1 install media with documentation (installation from DVD is described).
▶ SLES 11 install media or ISO image(s)
▶ 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
   – A Linux SSH client such as PuTTY (recommended) or TeraTerm
   – 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 must be routed to the OSA card(s) accessible to the LPAR.
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 System z DASD is addressed with a device number consisting of four hexadecimal digits. Each System z 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 DASDs 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 using 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 T is hard-coded into REXX EXECs that adopt this convention. If you want a different LPAR identifier character, it can easily be changed (set the firstChar variable).

2.2.2 Backup file naming convention

It is recommended 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, allows for command retrieval in both directions with the up arrow and down arrow keys.

There is a convention in z/VM to use the F12 function key (previously 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 using 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 2.3 GB, while 3390-9s are three times the size or about 6.8 GB. Either size will work, though 3390-3s have been recommended over 3390-9s by some performance analysts. This book describes using a mix of 3390-3s and 3390-9s.

Disk storage per Linux image

Disk storage has the following characteristics

- This version of the book now recommends two 3390-3 DASD to create minidisks at virtual addresses 100 and 101. Previous versions only recommended a single minidisk at virtual address 100.
- The root file system is on /dev/dasda1 with a recommended size of 384MB. It is not a logical volume so that if there are any problems with LVM, the system will still be able to boot.
Other file systems are on logical volumes that are part of single volume group with the following characteristics:

Table 2-1  Recommended logical volume file systems and sizes

<table>
<thead>
<tr>
<th>Mount point</th>
<th>Logical volume name</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>/usr/</td>
<td>usr-lv</td>
<td>2 GB</td>
</tr>
<tr>
<td>/var/</td>
<td>var-lv</td>
<td>512 MB</td>
</tr>
<tr>
<td>/opt/</td>
<td>opt-lv</td>
<td>384 MB</td>
</tr>
<tr>
<td>/tmp/</td>
<td>tmp-lv</td>
<td>384 MB</td>
</tr>
</tbody>
</table>

This layout uses about 3.5 GB out of 4.5 GB of disk space. You could choose to use other disk sizes than 3338 cylinders (3390-3 minus cylinder 0). For example, if you chose to use 3390-9s, you could give 100 and 101 each half of the volume, giving each Linux about 6.8 GB of disk space.

Important: However you choose to layout the minidisks, it is important that the golden image and all target Linux user IDs have two minidisks of the same size at virtual addresses 100 and 101. These assumptions are coded into the clone.sh script.

2.4 Memory planning

Planning memory may be the most difficult issue with z/VM and Linux on System z, yet the most important to ensure adequate performance. The simplest solution may appear to involve having enough central memory (storage) in the LPAR so that z/VM never pages and Linux never swaps. However, such resource is often not be realistically available. A good rule of thumb is to allocate memory on a just enough basis for each Linux server. A good starting point is to set a virtual machine size by changing the memory allocation value at just over the value at which the guest starts to swap at the Linux system level when under normal loading. If some level of sustained swapping is inevitable due to the nature of the workloads, then ensure virtual disks are used for the swap media.

An understanding of memory planning is recommended, here are some resources that cover this important topic:

- The IBM z/VM Performance Resource pages in general, on the Web at: http://www.vm.ibm.com/perf/

One rule that can be recommended is to only have as few virtual machines logged on (or disconnected) as possible to handle the workload being presented. Every virtual machine that is not required should be logged off where appropriate, as this will mean more memory for the other virtual servers which remain running.
2.5 Password planning

Good passwords are critical to good security. However, requiring many different passwords generally leads to people writing them down, which clearly 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 be the same person.

The method of backing up z/VM data onto the Linux cloner 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 cloner, 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 16.

2.6.1 z/VM resources used in this book

Table 2-2 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>LVM2</td>
<td>16 GB central storage/2 GB expanded, 10 shared IFLs</td>
</tr>
<tr>
<td>CPC name</td>
<td>H15C</td>
<td>Name of CPC on which the LPAR is located</td>
</tr>
<tr>
<td>z/VM system name</td>
<td>POKTST61</td>
<td>Name to be assigned to z/VM system</td>
</tr>
<tr>
<td>TCP/IP host name</td>
<td>gpok249</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>endicott.ibm.com</td>
<td>Helpful to set in DNS beforehand</td>
</tr>
<tr>
<td>TCP/IP gateway</td>
<td>9.60.18.129</td>
<td>The router to and from the local subnet</td>
</tr>
<tr>
<td>DNS server 1</td>
<td>9.0.2.11</td>
<td>Assigned by the network administrator</td>
</tr>
<tr>
<td>DNS server 2/3 (optional)</td>
<td>9.0.3.1</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>B420</td>
<td>Start of OSA triplet for the z/VM TCP/IP stack</td>
</tr>
<tr>
<td>TCP/IP address</td>
<td>9.60.18.249</td>
<td>The TCP/IP address of the z/VM system</td>
</tr>
<tr>
<td>Subnet mask</td>
<td>255.255.255.128</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>MTU size</td>
<td>1500</td>
<td>Check with network administrator</td>
</tr>
<tr>
<td>Primary OSA device number for VSWITCH</td>
<td>B440</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>B424</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 2-3 lists the z/VM DASD resource values used in the examples in this book.

<table>
<thead>
<tr>
<th>Real device</th>
<th>Label</th>
<th>Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>6280</td>
<td>610RES</td>
<td>CP owned</td>
<td>z/VM system residence volume</td>
</tr>
<tr>
<td>6281</td>
<td>TV6281</td>
<td>CP owned</td>
<td>z/VM spool volume 1 (label set at z/VM install)</td>
</tr>
<tr>
<td>6282</td>
<td>TV6282</td>
<td>CP owned</td>
<td>z/VM paging volume 1 (label set at z/VM install)</td>
</tr>
<tr>
<td>6283</td>
<td>TV6283</td>
<td>CP owned</td>
<td>z/VM first work volume (label set at z/VM install)</td>
</tr>
<tr>
<td>6284</td>
<td>TV6284</td>
<td>CP owned</td>
<td>z/VM 2nd work volume (label set at z/VM install)</td>
</tr>
<tr>
<td>6285</td>
<td>TV6285</td>
<td>CP owned</td>
<td>Paging volume 2</td>
</tr>
<tr>
<td>6286</td>
<td>TV6286</td>
<td>CP owned</td>
<td>Paging volume 3</td>
</tr>
<tr>
<td>6287</td>
<td>TV6287</td>
<td>CP owned</td>
<td>Paging volume 4</td>
</tr>
<tr>
<td>6289</td>
<td>TM6289</td>
<td>System minidisk</td>
<td>3390-3 for LNXMAINT and S11CLONE</td>
</tr>
<tr>
<td>6290</td>
<td>TM6290</td>
<td>System minidisk</td>
<td>3390-3 for S11CLONE</td>
</tr>
<tr>
<td>6293</td>
<td>TM6293</td>
<td>System minidisk</td>
<td>3390-3 for S11CLONE</td>
</tr>
<tr>
<td>6294</td>
<td>TM6294</td>
<td>System minidisk</td>
<td>3390-3 for S11CLONE</td>
</tr>
<tr>
<td>63CE</td>
<td>TM63CE</td>
<td>System minidisk</td>
<td>3390-9 for S11CLONE and S11GOLD</td>
</tr>
<tr>
<td>63CF</td>
<td>TM63CF</td>
<td>System minidisk</td>
<td>3390-9 for LINUX01 and LINUX02</td>
</tr>
<tr>
<td>63D0</td>
<td>TM63D0</td>
<td>System minidisk</td>
<td>3390-9 for LINUX02 and LINUX03</td>
</tr>
<tr>
<td>63D1</td>
<td>TM63D1</td>
<td>System minidisk,</td>
<td>3390-9 for LINUX04</td>
</tr>
</tbody>
</table>

2.6.3 Linux resources used in this book

Table 2-4 lists the Linux PC NFS server resources used for the first System z Linux install:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP/IP address</td>
<td>9.60.18.133</td>
<td></td>
</tr>
<tr>
<td>User/password</td>
<td>root/lnx4vm</td>
<td></td>
</tr>
<tr>
<td>NFS-exported install directory</td>
<td>/nfs/sles11/</td>
<td>Directory with DVD 1</td>
</tr>
</tbody>
</table>
Table 2-5 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>TCP/IP gateway</td>
<td>9.60.18.129</td>
<td>Obtain from network administrator</td>
</tr>
<tr>
<td>Subnet mask</td>
<td>255.255.255.128</td>
<td>Obtain from network administrator</td>
</tr>
<tr>
<td>DNS server</td>
<td>9.0.2.11, 9.0.3.1</td>
<td>Obtain from network administrator</td>
</tr>
<tr>
<td>VNC installation password</td>
<td>12345678</td>
<td>Must be 8 characters</td>
</tr>
</tbody>
</table>

2.6.4 Linux user IDs used in this book

Table 2-6 lists the z/VM user IDs for Linux used in the examples in this book.

<table>
<thead>
<tr>
<th>Name</th>
<th>IP address</th>
<th>DNS name</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>S11GOLD</td>
<td>9.60.18.223</td>
<td>gpok223.endicott.ibm.com</td>
<td>SLES 11 golden image</td>
</tr>
<tr>
<td>S11CLONE</td>
<td>9.60.18.224</td>
<td>gpok224.endicott.ibm.com</td>
<td>The cloner</td>
</tr>
<tr>
<td>LINUX01</td>
<td>9.60.18.225</td>
<td>gpok225.endicott.ibm.com</td>
<td>A Web virtual server</td>
</tr>
<tr>
<td>LINUX02</td>
<td>9.60.18.246</td>
<td>gpok246.endicott.ibm.com</td>
<td>An LDAP virtual server</td>
</tr>
<tr>
<td>LINUX03</td>
<td>9.60.18.247</td>
<td>gpok247.endicott.ibm.com</td>
<td>A file and print virtual server</td>
</tr>
<tr>
<td>LINUX04</td>
<td>9.60.18.248</td>
<td>gpok248.endicott.ibm.com</td>
<td>An application development virtual 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 2-7 to document the z/VM resources that you will use.

<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>Often “eth0”</td>
<td></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>Often “QDIO”</td>
<td></td>
</tr>
<tr>
<td>Network Type</td>
<td>Often “Ethernet”</td>
<td></td>
</tr>
<tr>
<td>Port name (optional)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Router Type</td>
<td>Often “None”</td>
<td></td>
</tr>
<tr>
<td>Primary OSA device number</td>
<td></td>
<td>Should be on a different CHPID/OSA card than primary</td>
</tr>
<tr>
<td>for VSWITCH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Secondary OSA device number</td>
<td></td>
<td>Should be on a different CHPID/OSA card than primary</td>
</tr>
<tr>
<td>number for VSWITCH</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2.7.2 z/VM DASD worksheet

Use the worksheet in Table 2-8 to document the z/VM DASD that you will use.

Table 2-8  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>
<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 2-10 to document the resources associated with the NFS server that will be used to be the install source of the first System z Linux.

**Table 2-9 Linux NFS server resources blank worksheet**

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP/IP address</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User/password</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NFS-exported install directory</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Use the worksheet in Table 2-11 to document your System z Linux resources.

**Table 2-10 Linux resources blank worksheet**

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<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 2-11 to document the Linux user IDs that you will create.

**Table 2-11 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>
Chapter 3. 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 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 3-4 should open. If you spend a few minutes to configure PuTTY it may pay off in time savings. The examples shown below are using PuTTY Release 0.60.

1. In the PuTTy Configuration window, in the left Category panel, click Session.
2. Under the *Connection Type* heading on the top right, click the **SSH** radio button as shown in Figure 3-1. This specifies to use the SSH protocol.

![PuTTY Configuration window](image)

*Figure 3-1  PuTTY Configuration window*

3. Click on **Logging** in the left panel as shown in Figure 3-2.
   - Click the radio button **Printable output** in the *Session logging* radio group. 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 a timestamp will be in the file name.

![Setting logging](image)

*Figure 3-2  Setting logging*
4. In the left panel, click **SSH** near the bottom as shown in Figure 3-3.

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

![Figure 3-3 Setting SSH Protocol 2](image)

6. In the left Category panel, click **Terminal** as shown in Figure 3-4.

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).

![Figure 3-4 Customizing PuTTY SSH settings (Part 1 of 4)](image)
8. Click **Window** in the left pane as shown in Figure 3-5.

9. You may choose a larger screen size and more lines of scroll back. In this example, 50 rows, 100 columns are and 1000 lines of scroll back are set.

10. Click **Session** in the left pane as shown in Figure 3-6.

11. Click **Default Settings** in the Saved Sessions pane, then click the **Save** button. This makes all future sessions that you define inherit the preferences you just set.
Saving sessions
To save sessions perform the following steps. In this example a session for S11CLONE, or the cloner, is saved

![PuTTY Configuration](image)

Figure 3-7 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* text area, choose a name that you will remember. In this example, the name S11CLONE (cloner) is used.
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 will be invoked.

3.2 Setting up a VNC client

A VNC client allows access to a graphical windowing 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 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](http://www.realvnc.com)
The download page is:

http://www.realvnc.com/download.html

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 4.1.2 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.

![RealVNC Select Components panel](image)

Figure 3-8  RealVNC Select Components panel

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.

If however, you have 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 check-box 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, 43 or more lines if they 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 two hours or more near the end of section 4.1.2, “Copying a vanilla z/VM system to DASD” on page 31. 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.

In this chapter you should perform the following steps:

- “Installing z/VM from DVD” on page 27
- “Configuring TCP/IP” on page 38
- “Configuring the XEDIT profile” on page 40
- “Customizing the SYSTEM CONFIG file” on page 41
- “Configuring TCP/IP to start at IPL time” on page 43
- “Adding paging volumes” on page 47
- “Creating a user ID for common files” on page 52

In addition, there are optional sections:

- “Addressing z/VM security issues” on page 59
- “Backing up your z/VM system to tape” on page 61
- “Relabeling system volumes” on page 61
- “Restoring your z/VM system from tape” on page 66

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 16. For System z9 hardware and older, you will need access to the Hardware...
Management Console (HMC) with a user ID that has authority to go into *single object operations mode*. For System z10 hardware and newer, the requirement to be in single object operations mode to access the *Load from CD-ROM, DVD or Server* function has been removed.

z/VM 6.1 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 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 6 Release 1.0*, GC24-6097. 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 do not address these options.

### 4.1.1 Booting z/VM from DVD

This section explains how to install z/VM 6.1 from an HMC with a DVD-ROM onto 3390-3 equivalent DASD. Some words are included for installing onto the larger 3390-9 DASD. For alternative configurations such as installing from tape or onto SCSI disks, refer to the z/VM documentation.

1. Logon to the Hardware Management Console. You should see the *HMC Workplace* window.
2. Select the LPAR on which you want to install z/VM - often by clicking on *CPC images* icon. **Note:** *BE SURE* you have the correct LPAR selected. If you are not completely sure, check with someone who is.
3. If necessary, click the *racetrack* buttons (two buttons that are circular arrows on the bottom right corner) to traverse to the *Recovery or CPC Recovery* menu.
4. On the *Recovery or 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 <your CPC>* 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.
5. Place the z/VM Product Package Version 6 Release 1.0 DVD in the HMC DVD drive.

**Important:** On z10 HMCs and later, it is no longer required to be in Single Object Operations mode in order to install z/VM.

6. On a z9 HMC and older, get into Single Object Operations mode (for z10 HMCs and later this step is not necessary). 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 not being valid may appear. 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.
7. The LPAR that z/VM will be installed into should still be selected. On the right you should still see the (CPC) Recovery menu. Double-click the Load from CD-ROM or Server icon as shown in Figure 4-2:

![CPC Recovery menu with "Load from CD-ROM or Server" icon present.]

8. On the Load CD-ROM or Server window as shown in Figure 4-3 on page 31, 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 6.1 installation code.

10. Click OK.
Chapter 4. Installing and configuring z/VM

11. Load the RAMDISK:

   a. From the *Load from CD-ROM or Server* panel, the software **610vm.ins** should be selected as shown in Figure 4-4. Click **OK**.

   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 two minutes or less.

   You should now have an in-memory z/VM 6.1 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 (if you are in it). To do so, log off the primary SE window by closing that window.

2. Move to the Integrated 3270 Console window (you can use the Alt-Tab sequence). The RAMdisk should IPL and you should see z/VM boot as shown in Figure 4-5.
   
   **Note:** The “Esc” key clears the screen on the HMC.

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
   ```

   ![Figure 4-5  z/VM first boot on Integrated console](image)
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 17. In this example, the devices are 6280-6284.

   => att 6280-6284 *
   6280-6284 ATTACHED TO MAINT

**Important:** The devices 6280-6284 are in bold italics 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 *

This convention is used throughout the book.

**Running INSTDVD**

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

1. Execute the INSTDVD EXEC:
2. If you are using 3390-3s, you see a panel asking for the five volumes as shown in Figure 4-7 (if you are using 3390-9s, you will only see three lines).

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

<table>
<thead>
<tr>
<th>TYPE</th>
<th>DASD</th>
<th>ADDRESS</th>
<th>FORMAT DASD</th>
</tr>
</thead>
<tbody>
<tr>
<td>R6S</td>
<td>610R6S</td>
<td>6280</td>
<td></td>
</tr>
<tr>
<td>IPool</td>
<td>TV6281</td>
<td>6281</td>
<td></td>
</tr>
<tr>
<td>Page</td>
<td>TV6282</td>
<td>6282</td>
<td></td>
</tr>
<tr>
<td>User</td>
<td>TV6283</td>
<td>6283</td>
<td></td>
</tr>
<tr>
<td>User</td>
<td>TV628A</td>
<td>6284</td>
<td></td>
</tr>
</tbody>
</table>

a. Enter the addresses of the five volumes (or three for 3390-9s) that z/VM will be installed on.

b. Press F5 to start the installation.

3. Verify that the five DASD addresses to be installed onto are correct. When you see the question **DO YOU WANT TO CONTINUE?**, type **Y**. You should see the message **NOW FORMATTING DASD 6280**.

4. You are asked to place the system RSU in the drive. Insert the **z/VM Stacked Recommended Service Upgrade 6101** DVD into the HMC DVD-ROM drive.

5. At the **Integrated 3270 Console**, type **60**. You should see a messages of the form **DVDLOAD: LOADING FILE CKD5000x IMAGE */.** This step should take two to four minutes.

6. 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. Your **3270 Integrated Console** session should still be running.

1. From the **HMC Workplace window**, **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. Set the load address to the new system residence (610RES) volume which is **6280** in this example.
   b. Set the load parameter to **SYSG**. This specifies to use the Integrated 3270 console.
   c. Click **OK** to IPL.

![Figure 4-8  Load window](image)

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. 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 1-3 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 using 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 **HCPLNM102E** or **HCPLNM101E** 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 then press **Enter** at the VM READ prompt in the lower right corner. You should see the Ready; prompt.

```text
==> ipl cms
==> Press Enter at the VM READ prompt
```

3. Run the **instvm dvd** command:

```text
==> 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. First **IPL CMS** then press **Enter** at the VM READ prompt:

```text
==> ipl cms
==> Press Enter at the VM READ prompt
Ready;
```

5. For z/VM 6.1, the service name is **6101RSU1**. Verify this file exists on the MAINT 500 disk:

```text
==> acc 500 c
DMSACC724I 500 replaces C (2CC)
==> listfile * * c
6101RSU1 SERVLINK C1
```

6. Run the **SERVICE ALL** command to apply the service:

```text
==> service all 6101rsu1
```

This step should take about 3-6 minutes. The last message should be:

```text
VMFSRV2760I SERVICE processing completed successfully.
```

7. Now **IPL CMS** and run the **put2prod** command. This puts the service into production:

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

This step should take about 2-4 minutes. The last message should be:

```text
VMFP2P2760I PUT2PROD processing completed successfully.
```

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

```text
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.

8. Enter the following command to shutdown and re-IPL your system:

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

9. You will lose the current session on the **Integrated 3270 Console**, but the system should come back in about 2-4 minutes.

10. 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 using 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 using 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

3. The z/VM TCP/IP Configuration Wizard opens as shown in the preceding example. The first field, User ID, should always be TCPIP. Obtain the remaining values from the 2.7.1, “z/VM resources worksheet” on page 16 and press F8.
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 (layer 3) with modern OSA devices. When completed, press **F8**.

**Note:** to utilize QDIO (layer 2), certain prerequisites must be met. Consult with the system administrator.
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 (9.12.5.22)
Successfully PINGed Gateway (9.12.4.1)
Successfully PINGed DNS (9.12.6.7)
DTCIPW2519I Configuration complete; connectivity has been verified
DTCIPW2520I File PROFILE TCPIP created on TCPIP 198
DTCIPW2520I File TCP/IP DATA created on TCPIP 592
DTCIPW2520I File SYSTEM DTCPARMS created on TCPIP 198
HCPINP8392I IPWIZARD EXEC ENDED SUCCESSFULLY
DMSVML2061I TCPIP 592 released
```

7. At this point your z/VM TCP/IP stack should be up. You should now be able to ping it from another system.

   If the **IPWIZARD** fails you must continue debugging it until it succeeds. Double check all values. Verify that the TCP/IP network and OSA information you were given are properly associated.

---

**HMC Integrated 3270 Console or 3270 emulator?** At this point z/VM should be accessible over the network. You can continue working at the HMC, or you can access your new system using a 3270 emulator. See 3.2, “3270 emulators” on page 17 for some brief words on that subject.

If you want to switch to 3270 emulator, first **LOGOFF** of MAINT or **DISConnect** on the Integrated 3270 Console.

If you logoff the session is ended - it is analogous to shutting and powering down a PC. If you disconnect, your session remains where it is and is resumed when you log back on. It is analogous to turning a PC’s monitor off. In general, you should **LOGOFF** of system administration user IDs such as MAINT. However, you should always **DISCONNECT** from z/VM service machines such as TCPIP and user IDs running Linux. Logging off of them will terminate the service or crash Linux.

---

**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 personal or shared system 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.

If you have never used XEDIT before, there is a cheat sheet in Appendix A.4.1, “XEDIT cheat sheet” on page 217. The z/VM 6.1 PDF library is on the Web at:

[http://www-03.ibm.com/systems/z/os/zos/bkserv/zvmpdf/#zvm61](http://www-03.ibm.com/systems/z/os/zos/bkserv/zvmpdf/#zvm61)
Search for the XEDIT User’s Guide and Command Reference. Also there is an old manual available online:

http://ukcc.uky.edu/ukccinfo/391/xeditref.html

One default 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:

```plaintext
=>> copy profile xedit a profile xeditorig 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 using the CPRELEASE command. The CP disks are queried using the QUERY CPDISK command. Note the MAINT CF1 disk is accessed as CP disk A before it is released but not after.

```plaintext
=>> q cpdisk
Label Userid Vdev Mode Stat Vol-ID Rdev Type StartLoc EndLoc  
MNTCF1 MAINT OCF1 A R/O 610RES 6280 CKD  39 158  
MNTCF2 MAINT OCF2 B R/O 610RES 6280 CKD  159 278  
MNTCF3 MAINT OCF3 C R/O 610RES 6280 CKD  279 398  
=>> cprel a
CPRELEASE request for disk A scheduled.  
HCPZAC6730I CPRELEASE request for disk A completed.  
=>> q cpdisk
Label Userid Vdev Mode Stat Vol-ID Rdev Type StartLoc EndLoc  
MNTCF2 MAINT OCF2 B R/O 610RES 6280 CKD  159 278  
MNTCF3 MAINT OCF3 C R/O 610RES 6280 CKD  279 398  
```

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 as your F disk.

```plaintext
=>> link * cf1 cf1 mr
=>> acc cf1 f
```
3. Make a backup copy of the vanilla SYSTEM CONFIG file using the COPYFILE command with the OLDDATE parameter so the timestamp of the file is not modified. Note that because the target file name (system) and mode (f) are the same, the equal sign (=) can be used as a wildcard.

```bash
=> copy system config f = conforig = (oldd
```

4. Edit the original file:

```bash
=> x system config f
```

5. The system name is set to ZVMV6R10 by default in the System_Identifier_Default statement. You can search for it using the / subcommand:

```bash
=====> /System_Identifier_D
```

Modify this to the new name of your system. In this example P0KST61 is used.

```
System_Identifier_Default P0KST61
```

6. 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 ,
   Syslim infinite , /* Allow VDISKS for Linux swaps */
   Userlim infinite
```

7. 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. This will set the first three bytes of the MAC address created for each virtual NIC. If you have a multiple z/VM systems, increment this value to avoid having identical MAC addresses created. The last three bytes of the MAC address are automatically incremented by z/VM as they are assigned, so they will be unique on each z/VM system. Modify the two starting addresses of the OSA triplets (B440 and B424 in this example) to those you specified in 2.7.1, “z/VM resources worksheet” on page 16.

```bash
=====> bot
=====> a3
/* define vswitch named vsw1 and set MAC address prefixes to 02-00-01 */
define vswitch vsw1 rdev B440 B424
vmlan macprefix 020001
```

8. Save your changes with the XEDIT FILE subcommand:

```bash
=====> file
```
9. Test your changes with the CPSYNTAX command which is on the MAINT 193 disk:

```plaintext
=> 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.

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

```plaintext
=> 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.
```

11. Verify that the CP disk A has been accessed using the QUERY CPDISK command:

```plaintext
=> 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 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>610RES</td>
<td>6280 CKD</td>
<td>39</td>
<td>158</td>
</tr>
<tr>
<td>MNTCF2</td>
<td>MAINT</td>
<td>0CF2</td>
<td>B</td>
<td>R/O</td>
<td>610RES</td>
<td>6280 CKD</td>
<td>159</td>
<td>278</td>
</tr>
<tr>
<td>MNTCF3</td>
<td>MAINT</td>
<td>0CF3</td>
<td>C</td>
<td>R/O</td>
<td>610RES</td>
<td>6280 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 TCPIP 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.

```plaintext
=> 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.

```plaintext
z/VM Version 6 Release 1.0, Service Level 0901 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES: NO RDR, NO PRT, NO PUN
LOGON AT 09:29:16 EST FRIDAY 11/20/09
DMSIND2015W Unable to access the Y-disk. Filemode Y (19E) not accessed
z/VM V6.1.0 2009-11-19 13:47
```

```plaintext
=> 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 the WMLINK 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:
```
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. You can safely delete the Address Command line.

   b. Add a line to start the TCPIP user ID using the XAUTOLOG command and keep two statements that start the VSWITCH cloners.

   c. Add a line to logoff of AUTLOG1 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:**
   ```
   /**************************/
   /*  Autolog1 Profile Exec */
   /**************************/
   'cp xautolog tcpip'  /* start up TCPIP */
   'CP XAUTOLOG VMSERVS'
   'CP XAUTOLOG VMSERVU'
   'CP XAUTOLOG VMSERVR'
   'CP XAUTOLOG DTCVSW1'
   'CP XAUTOLOG DTCVSW2'
   'cp logoff'  /* logoff when done */
   ```

7. Save your changes with the FILE subcommand

   ```
   ====> file
   ```

8. Logoff of AUTLOG1:

   ```
   => 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.
1. **Logoff of AUTOLOG1 and logon to TCPMAINT.** The PROFILE TCPIP file is on the TCPMAINT 198 disk which is accessed as the D disk.

2. Make a backup copy the original PROFILE TCPIP, then rename it to `<SYSTEM_ID> TCPIP` (where `<SYSTEM_ID>` is POKTST61 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 poktst61 = =

3. You have now backed up and renamed your TCP/IP profile. You can verify using the LISTFILE command:

   => listfile * * d
   POKTST61 TCPIP D1
   SYSTEM DTCPARMS D1
   PROFILE TCPIORIG D1

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.

   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**:

      => 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:

      => 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

   Turn on the FTP server by editing the renamed configuration file:

   1. Add an AUTOLOG statement near the top of the file with FTPSERVE as the only entry.

   2. 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 poktst61 tcpip d

      **Before:**

      ; -----------------------------------------------------------------------------------------------
      ; OPERATOR TCPMAINT MAINT MROUTE DHCPD REXECN SNMPD SNMPQE LDAPSRV
      ; ENDOPER
      ; -----------------------------------------------------------------------------------------------
      PORT
      ; 20 TCP FTPSERVE  NOAUTOLOG ; FTP Server
      ; 21 TCP FTPSERVE ; FTP Server
      ; 23 TCP INTCLIEN ; TELNET Server
      ; 25 TCP SMTP ; SMTP Server
      ...

      **After:**
3. Save your changes with the FILE subcommand:

```
====> file
```

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

You may 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.

1. 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. When your system is back up, perform the following commands:

2. Start a 3270 session and Logon as MAINT. This shows that there is TCP/IP access to z/VM.

3. Query the new VSWITCH:

```
====> q vswitch
VSWITCH SYSTEM VSW1 Type: VSWITCH Connected: 0 Maxconn: INFINITE
  PERSISTENT Restricted Nonrouter Accounting: OFF
  VLAN Unaware
  MAC address: 02-00-07-00-00-01
  State: Ready
  IPITimeout: 5 QueueStorage: 8
  Isolation Status: OFF
  RDEV: B440.P00 VDEV: B440 Controller: DTCVSW2
  RDEV: B424.P00 VDEV: B424 Controller: DTCVSW1 BACKUP
```

You should see that the VSWITCH VSW1 exists, that the OSA devices you specified are being used and that there are two built-in VSWITCH controllers, DTCVSW1 and DTCVSW2.

4. Use the QUERY VDISK and QUERY RETRIEVE commands to see the changes made to the Features statement in the SYSTEM CONFIG file:
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 6.1 is installed 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 at least three paging volumes so you will have a total of four (or one more 3390-9). 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 using 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.2.1, “The CPFORMAT EXEC” on page 219. 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 9. If you want different volume labels, you can use the CPFMTXA command and manually specify each volume label, or you can modify the REXX EXEC.

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 using FTP.

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, it is required in order to proceed.

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:

```bash
# cd /nfs/virt-cookbook-S11/vm
# 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. Issue the ASCII subcommand to be sure the ASCII characters are converted to EBCDIC, and use the PUT subcommand to copy the file.

```
# ftp 9.60.18.249
Name (9.12.5.22:root): maint
331-Password: maint
230-MAINT logged in; working directory = MAINT 191
...
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 `M<t><xxxx>` where:
- `<t>` is type - P (page), M (perm), S (spool) or T (Temp disk)
- `<xxxx>` is the 4 digit address

Syntax is:

```
  .-PAGE-.

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

The following example shows how to attach three 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 17:

```
For 3390-9 volumes: If you are installing onto 3390-9s only one more paging volume may be adequate to start. This will give you two full volumes, or the same equivalent of six 3390-3s.
```

1. The DASD that will be used for paging volumes are at real addresses 6285, 6286, and 6287. First query the DASD devices to see their status:

```
=> q 6285 6286 6287
DASD 6285 TM6285 , DASD 6286 TM6286 , DASD 6287 TM6287
```

2. Attach the devices to MAINT (the last parameter of `*` means the current user ID) using the ATTACH command:

```
=> att 6285-6287 *
6285-6287 ATTACHED TO MAINT
```

3. Use the CPFORMAT command with the AS PAGE parameter:

```
=> cpformat 6285-6287 as page
```
4.6.2 Formatting DASD for minidisks

In addition to CP disks such as page space, System disks will be needed to create minidisks for the virtual machines. In this section the DASD which will be used for the minidisks of LNXMAINT, S11CLONE and S11GOLD will be formatted.

1. Query the DASD that will be used for minidisks. In this example they are 6289, 6290, 6293, 6294 (3390-3s) and 63CE (a 3390-9).

   ```
   ==> q 6289 6290 6293 6294 63ce
   DASD 6289 TM6289 , DASD 6290 TM6290 , DASD 6293 TM6293 , DASD 6294 TM6294
   DASD 63CE TM63CE
   ```

2. Attach the eight volumes that will be used for the cloner, the common CMS disk and the golden image. In this example it is the DASD at addresses:

   ```
   ==> att 6289 6290 6293 6294 63ce *
   6289 6290 6293 6294 63CE ATTACHED TO MAINT
   ```

3. Invoke the CPFORMAT command against these volumes using the parameter **as perm**:

   ```
   ==> cpformat 6289 6290 6293 6294 63ce as perm
   ```

   Format the following DASD:

   ```
   TargetID  Tdev  OwnerID  Odev  Dtype  Vol-ID  Rdev     StartLoc  Size
   MAINT   6289  MAINT   6289  3390  TM6289  6289      0       3339
   MAINT   6290  MAINT   6290  3390  TM6290  6290      0       3339
   MAINT   6293  MAINT   6293  3390  TM6293  6293      0       3339
   MAINT   6294  MAINT   6294  3390  TM6294  6294      0       3339
   MAINT   63CE  MAINT   63CE  3390  TM63CE  63CE      0       10017
   ```

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

   ...
You should now have newly formatted volumes that can be used as minidisks.

### 4.6.3 Updating the SYSTEM CONFIG file

Now that the PAGE and PERM volumes are ready for use, they must be added to the SYSTEM CONFIG file so that z/VM can use them. Follow these steps to update the SYSTEM CONFIG file:

1. Logon to MAINT.

2. 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  610RES 61A2 CKD          39        158
   MNTCF2 MAINT    0CF2 B   R/O  610RES 61A2 CKD         159        278
   MNTCF3 MAINT    0CF3 C   R/O  610RES 61A2 CKD         279        398
   => 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.

3. Make a copy of the working SYSTEM CONFIG file using the “WRKS” (it works!) suffix convention:

   ```
   => copy system config f = confwrks =
   ```

4. 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
   ...  
   /*****************************************************************************/
   /*                     CP_Owned Volume Statements                        */
   /*****************************************************************************/
   CP_Owned Slot   1  610RES
   CP_Owned Slot   2  TV6281
   CP_Owned Slot   3  TV6282
   CP_Owned Slot   4  TV6283
   CP_Owned Slot   5  TV6284
   CP_Owned Slot   6  TP6285
   CP_Owned Slot   7  TP6286
   CP_Owned Slot   8  TP6287
   CP_Owned Slot   9  RESERVED
   CP_Owned Slot  10  RESERVED
   CP_Owned Slot  11  RESERVED
   ...
   ```

5. 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 using the **User Volume Include** statement. If you are using the labelling convention enforced by the **CPFORMAT EXEC** and no other LPAR will be using the same volumes with the same prefix, then add the following single line to include all PERM space as volume labels all begin with “TM” (the labeling convention used by the **CPFORMAT EXEC**).
6. Save your changes with the FILE subcommand. Verify the integrity of the changes with the CPSYNTAX command:

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

7. When you have confirm there are no syntax errors, put the MAINT CF1 disk back online. The following example shows how you did this previously:

```bash
===> 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.
===> q cpdisk
Label  Userid   Vdev Mode Stat Vol-ID Rdev Type   StartLoc     EndLoc
MNTCF1 MAINT    0CF1  A   R/O  610RES 0200 CKD          39         83
MNTCF2 MAINT    0CF2  B   R/O  610RES 0200 CKD          84        128
MNTCF3 MAINT    0CF3  C   R/O  610RES 0200 CKD         129        188
```

### 4.6.4 Testing the changes

It is recommended that you again shutdown and relIPL to test the changes. Before you shut down, note that you have only one page volume (TV6282 in this example) using the QUERY ALLOC PAGE command. Your output should look similar to the following:

```bash
===> q alloc page
EXTENT  EXTENT  TOTAL  PAGES  HIGH   %
VOLID  RDEV  START  END  PAGES IN USE  PAGE USED
------- ----- ------- ----- ---------- ---------- ---------- -----
TV6282  6282   1  3338  600840      0      0   0%
```
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 using a 3270 emulator, you will lose your session, but if all goes well, your system will be available again in a couple of minutes.

```sh
close ipl reipl iplparms cons=sysc
```

After the system comes back, `logon as MAINT` and look at the page space again. You should now see that you have six paging volumes:

```sh
c q alloc page
```

<table>
<thead>
<tr>
<th>VOLID</th>
<th>RDEV</th>
<th>START</th>
<th>END</th>
<th>PAGES IN USE</th>
<th>PAGE USED</th>
</tr>
</thead>
<tbody>
<tr>
<td>TV6282</td>
<td>6282</td>
<td>1</td>
<td>3338</td>
<td>600840</td>
<td>1 5 1%</td>
</tr>
<tr>
<td>TP6285</td>
<td>6285</td>
<td>0</td>
<td>3338</td>
<td>601020</td>
<td>0 0 0%</td>
</tr>
<tr>
<td>TP6286</td>
<td>6286</td>
<td>0</td>
<td>3338</td>
<td>601020</td>
<td>0 0 0%</td>
</tr>
<tr>
<td>TP6287</td>
<td>6287</td>
<td>0</td>
<td>3338</td>
<td>601020</td>
<td>0 0 0%</td>
</tr>
</tbody>
</table>

SUMMARY 2348K 1 1%
USABLE 2348K 1 1%

The output shows there are four paging volumes constituting 2348 KB pages, or about 9 GB of page space (a page is 4KB).

### 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:

```sh
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 17). 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:

```sh
c x user direct c
c == bottom  
c a 6

USER LNXMAINT LNXMAINT 64M 128M BEG
   INCLUDE TCPCMSU
   LINK TCPMAINT 592 592 RR
   MDISK 0191 3390 0001 0020 TM6289 MR READ WRITE MULTIPLE
   MDISK 0192 3390 0021 0300 TM6289 MR ALL WRITE MULTIPLE
```

Note the following points for the numbers in black:
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 TM6289

5. Define 192 minidisk of size 300 cylinders (approximately 225MB) from volume TM6289 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, overlaps can occur because 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:

```bash
==> x user diskmap
====> prefix off
```

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
```

Three GAPS should be listed on the right side:

- 501 cylinders on the $$$$$$ volume
• 1 cylinder on the $$LNX$$ volume
• 1 cylinder on volume used for LNXMAINT 191 and 192 disks (TM6289 in this example)

You don't have to worry about the first two gaps as they are expected given the layout of the default USER DIRECT file. 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.

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 at virtual address A04 for the first cylinder of the DASD you added (the label is TM6289 in this example):

```
===> x user direct
====> /user $alloc
USER $ALLOC$  NOLOG
    MDISK A01 3390 000 001 610RES R
    MDISK A02 3390 000 001 TV6283 R
    MDISK A03 3390 000 001 TV6284 R
    MDISK A04 3390 000 001 TM6289 R
```

7. Save your changes and run DISKMAP again. Edit the USER DISKMAP file. This time you should see just two gaps for volumes with labels $$LNX$$ and $$LNXMAINT$$. If you search for $ALLOC$ user ID, you should see the disk map of the volume you added for LNXMAINT:

```
===> diskmap user
===> x user diskmap
====> prefix off
====> /$ALLOC
TM6289   $ALLOC$   A04 3390 00000 00000 00001
       LNXMAINT 0191 3390 00001 00020 00020
       LNXMAINT 0192 3390 00021 00320 00300
...
```

8. When you are done you can quit without saving changes by pressing F3.

    ===> F3

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

```
===> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 5 RELEASE 4.0
EOJ DIRECTORY UPDATED AND ON LINE
HCPDIR494I User directory occupies 43 disk pages
```

If the DIRECTXA command fails, 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.**

```
LOGON LNXMAINT
z/VM Version 6 Release 1.0, Service Level 0901 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:  NO RDR,  NO PRT,  NO PUN
LOGON AT 13:14:38 EST FRIDAY 11/20/09
z/VM V6.1.0  2009-11-19 13:47
```
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:

```shell
==> 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:

```shell
==> 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. You can confirm this using the QUERY DISK command:

```shell
==> q disk
LABEL  VDEV M  STAT   CYL TYPE BLKSZ   FILES  BLKS USED-(%) BLKS LEFT  BLK TOTAL
LNX191 191 A   R/W    20 3390 4096        0          7-00       3593       3600
LXM192 192 D   R/W   300 3390 4096        0         11-00      53989      54000
MNT190 190 S   R/O   100 3390 4096      694      15028-83       2972      18000
MNT19E 19E Y/S R/O   250 3390 4096     1021      28254-63      16746      45000
```

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:

```shell
==> 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:

```shell
==> copy profile xedit z = = a
==> rel z (det)
```

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 using XEDIT and add the following lines (be sure to type the A file mode so you don't pick up a PROFILE EXEC on another disk). REXX EXECs must always begin with a C language-style comment.

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

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

2. You could test your changes by logging off and logging back on. However, typing the command `PROFILE` will do the same.

```c
profile
```

3. 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. Again verify with the `QUERY DISK` command:

```c
q disk
```

4. Verify that your F11 and F12 keys are set to the RETRIEVE command using the `QUERY PFKEYS` command:

```c
q pf
```

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. The directory name will be one of the following two depending on the distribution you are working with:

```bash
# cd /nfs/virt-cookbook-S11/vm
```

FTP to z/VM. By default FTP copies files to your 191 disk, so first change directory to the LNXMAINT 192 disk. The files are all in ASCII and the default behavior is to convert to ASCII to EBCDIC. Use the `mput *` subcommand to copy all the files from the `vm/` subdirectory to LNXMAINT:

```bash
# ftp 9.60.18.249
Connected to 9.12.5.22.
```
Logon to LNXMAINT. You should see the following files on your D disk:

```
===> file * * d
LNXMAINT FILELIST A0  V 169  Trunc=169 Size=5 Line=1 Col=1 Alt=0
Cmd Filename Filetype Fm Format Lrecl Records Blocks Date     Time
SLES11   EXEC     D1 V         68         10          1 11/20/09 13:45:52
SWAPGEN  EXEC     D1 V         72        467          6 11/20/09 13:45:52
CHPW610  XEDIT    D1 V         72        189          3 11/20/09 13:45:51
CPFORMAT EXEC     D1 V         79        252          3 11/20/09 13:45:51
SAMPLE   PARM-S11 D1 V         80         11          1 11/20/09 13:45:51
```

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 using the SET SIGNAL command
- Overcommit memory using the SET SRM STORBUF command
- Grant access to the VSWITCH for each Linux user
- Start user IDs that should be started using the XAUTOLOG command
- Limit minidisk cache in central storage and turn it off in expanded storage

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 6 Release 1.0, Service Level 0901 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES: NO RDR, NO PRT, NO PUN
LOGON AT 09:29:16 EST FRIDAY 11/20/09
DMSIND2015W Unable to access the Y-disk. Filemode Y (19E) not accessed
z/VM V6.1.0    2009-11-19 13:47
```

```
===> acc (noprof
```

2. Make a copy of the working PROFILE EXEC:

```
===> copy profile exec a = execwrks =
```
3. Edit the file and add the emboldened text. A **LOGOFF** command is added at the end of the EXEC so the virtual machine will be logged off when it is complete. This will save a small amount of memory on the system, but does add the requirement that you type acc (noprof) at the VM READ prompt when you log on interactively.

```plaintext
=> x profile exec
/*****************************/
/* Autolog1 Profile Exec */
/*****************************/
'cp xautolog tcpip'        /* start up TCPIP */
'CP XAUTOLOG VMSEsrvs'
'CP XAUTOLOG VMservu'
'CP XAUTOLOG VMservr'
'CP XAUTOLOG DTCVSW1'
'CP XAUTOLOG DTCVSW2'
'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 300' /* Allow guests 5 min to shut down */
'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 will not be optimal for all z/VM systems. 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 reIPL z/VM again. Be sure you are in a position to do so!

Perform the following steps:

1. Shutdown and reIPL your system.

```plaintext
=> shutdown reipl iplparms cons=sysc
SYSTEM SHUTDOWN STARTED
```

2. When your system comes back logon as MAINT.

3. Query the SRM values to see that the new STORBUF settings is in effect and the SIGNAL SHUTDOWN value is set to 300 seconds:

```plaintext
=> 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: 300 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 CA 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 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 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 CHPW610 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 an XEDIT macro named CHPW610 XEDIT has been included with this book.

The source code is in Appendix B.2.2, “The CHPW610 XEDIT macro” on page 220.

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 CHPW610 XEDIT macro:

```
=> 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 chpw610) followed by the new password. Rather than invoking the default profile of PROFILE XEDIT, this command will invoke the XEDIT macro named CHPW610 XEDIT and pass it the new password. For example, to change all passwords to lnx4vm, enter the following command:

```
=> x user direct c (profile chpw610) lnx4vm
```

Changing all passwords to: LNX4VM

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 4.0
EQI DIRECTORY UPDATED AND ON LINE
HCPDIR494I User directory occupies 43 disk pages
```

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 vm53nx, invoke the following commands:

```
=> x user direct c
====> c/LNX4VM/VM4LNX/** *
DMSXCG517I 798 occurrence(s) changed on 345 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 610RES, 610SPL, 610PAG, 610W01 and 610W02 (or just the first three if you are using 3390-9s). If you changed the labels of the last four at install time, then use those labels. 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.

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 system volumes

In previous books, the z/VM installation was described using “standard labels” on the CP-owned volumes (e.g. 610RES, 610SPL, 610PAG, 610W01 and 610W02). In this book, changing the last four labels to include the real device address in the last four characters of each label is recommended (the label of the “res pack”, e.g. 610RES cannot be modified at install time). This alleviates the possibility that another vanilla z/VM system with the same labels is installed onto volumes accessible by your z/VM system. If that happens, it is likely that one of the systems will not IPL correctly.

To understand this possibility, refer to Figure 4-13 on page 61. 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 540RES 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!

![Figure 4-13 The problem with two z/VM systems with identical volume labels](image)

In previous books a REXX EXEC and an XEDIT macro were provided to help in the process of relabeling system volumes. However, if you followed the previous steps, you will have only one standard label, 610RES. The EXEC and macro are no longer provided because they relied on standard labels. However, high level steps are still included. If you modified all labels...
except for the first one at install time, it is usually not necessary to perform the steps in this section.

If you do need to relabel the system volumes, perform the following steps:

- “Modifying labels in the SYSTEM CONFIG file” on page 62
- “Modifying labels in the USER DIRECT file” on page 63
- “Changing the labels on the five volumes” on page 64
- “Shutting down your system and restarting it” on page 64

**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 LABEL540 EXEC and **XEDIT** macro:
   ```
   => 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:
   ```
   => q cpowned
   1  540RES  D850  Own  Online and attached
   2  540SPL  D851  Own  Online and attached
   3  540PAG  D852  Own  Online and attached
   4  540W01  D853  Own  Online and attached
   5  540W02  D854  Own  Online and attached
   6  MPD855  D855  Own  Online and attached
   ...
   ```
4. 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:
   ```
   => cprel a
   CPRELEASE request for disk A scheduled.
   HCPZAC6730I CPRELEASE request for disk A completed.
   => link * cf1 cf1 mr
   => acc cf1 f
   => copy system config f = confwrks = (oldd rep
   ```
5. Edit the **SYSTEM CONFIG** file and modify the five labels (if you installed onto 3390-9s, there are only three labels, no W01 and W02 volumes are required):
   ```
   => x system config f
   ===> c/540RES/MVD850
   DMSXCG517I 3 occurrence(s) changed on 3 line(s)
   ===> top
   ===> c/540SPL/MVD851
   DMSXCG517I 1 occurrence(s) changed on 1 line(s)
   ===> top
   ===> c/540PAG/MVD852
   DMSXCG517I 1 occurrence(s) changed on 1 line(s)
   ===> top
6. 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:

```bash
====> top
====> /cp_owned
/**                     CP_Owned Volume Statements                     */
******************************************************************************
CP_Owned   Slot   1  MVD850
CP_Owned   Slot   2  MVD851
CP_Owned   Slot   3  MVD852
CP_Owned   Slot   4  MVD853
CP_Owned   Slot   5  MVD854
CP_Owned   Slot   6  MPD855
...
====> file
```

7. Verify there are no syntax errors:

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

8. Release and detach the F disk, CPACCESS the A disk and verify with the QUERY CPDISK command:

```bash
==> rel f (det
DASD OCF1 DETACHED
==> cpacc * cfl a
CPACCESS request for mode A scheduled.
Ready; T=0.01/E.01 09:19:57
HCPZAC6732I CPACCESS request for MAINT's OCF1 in mode A completed.
==> q cpdisk
Label  Userid   Vdev Mode Stat Vol-ID Rdev Type   StartLoc     EndLoc
MNTCF1 MAINT    0CF1  A   R/O  540RES D850 CKD          39        158
MNTCF2 MAINT    0CF2  B   R/O  540RES D850 CKD         159        278
MNTCF3 MAINT    0CF3  C   R/O  540RES D850 CKD         279        398
```

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 labels in the USER DIRECT file. For 3390-9s you will need only the first three CHANGE subcommands:

```bash
==> copy user direct c = direwrks = (oldd rep
==> x user direct c (profile label540) <d850 d851 d852 d853 d854>
====> c/540RES/MVD850
DMSXCG65171 94 occurrence(s) changed on 94 line(s)
====> top
====> c/540SPL/MVD851
DMSXCG65171 78 occurrence(s) changed on 78 line(s)
====> top
====> c/540PAG/MVD852
```
Traverse the file to view the changes before saving the changes with the FILE subcommand:

```bash
ten 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 LABEL540 EXEC has been written to make this step easier.

1. Use the CPFMTXA command to relabel the 5 system volumes (you will only need the first three if you installed onto 3390-9s):

```bash
impl rpm  d850 mvd850 label
...impl rpm  d851 mvd851 label
...impl rpm  d852 mvd852 label
...impl rpm  d853 mvd853 label
...impl rpm  d854 mvd854 label
```

2. Now that the five volumes have been relabeled (sometimes called clipping the volumes, derived from a contraction of the z/OS term change label program), you can run the DIRECTXA command to update the directory:

```bash
impl rpm  d850 mvd850 label
```

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
impl rpm  shutdown
```

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.

```
impl rpm  SHUTDOWN STARTED
HCPSHU960I System shutdown may be delayed for up to 210 seconds
```
Perform the following steps to bring the system back up:

1. Open an HMC session
2. Select your LPAR
3. Use the circular arrow racetrack buttons to get to the CPC Recovery (or just Recovery) menu.
5. Double-click the **LOAD** menu item. The Load Address (0850 in this example) and Load Parameter (SYSG) fields should be correct from the previous IPL.
6. Select the **Clear** radio button. The Load Address and Load Parameter fields should be correct from the previous IPL. Click **OK**
7. Click **Yes** on the Load Task Confirmation window.
8. 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**
9. Press the **F10** key to continue the IPL of your z/VM system. This should take 1-3 minutes.
10. At the Start prompt you have to specify a **FORCE** start, again because the spool volume label has changed:

    ```
    => force drain
    ```
11. Do not change the time of day clock:

    ```
    => no
    ```
12. When the IPL completes, **DISCONNECT** from the **OPERATOR** user ID:

    ```
    => disc
    ```
13. Close the HMC windows.
14. Start a 3270 emulator session as the TCPIP service machine should be up. **Logon as MAINT**.

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>MVD850</td>
<td>D850</td>
<td>Own</td>
<td>Online and attached</td>
</tr>
<tr>
<td>2</td>
<td>MVD851</td>
<td>D851</td>
<td>Own</td>
<td>Online and attached</td>
</tr>
<tr>
<td>3</td>
<td>MVD852</td>
<td>D852</td>
<td>Own</td>
<td>Online and attached</td>
</tr>
<tr>
<td>4</td>
<td>MVD853</td>
<td>D853</td>
<td>Own</td>
<td>Online and attached</td>
</tr>
<tr>
<td>5</td>
<td>MVD854</td>
<td>D854</td>
<td>Own</td>
<td>Online and attached</td>
</tr>
<tr>
<td>6</td>
<td>MPD855</td>
<td>D855</td>
<td>Own</td>
<td>Online and attached</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Important:** 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**. To rebuild saved segments, try the following commands (**only do this** if your saved segments are trashed!):

```bash
=> vmfsetup zvm cms
=> sampnss cms
=> ipl 190 clear parm nosprof instseg no
=> acc (noprof
=> acc 5e6 b
=> acc 51d d
=> 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. If you have backed up your system in section 4.10, “Backing up your z/VM system to tape” on page 61, 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 the two main types of service:

- A Programming Temporary Fix (PTF) which is analogous to a bug fix.
- A Recommended Service Upgrade (RSU) which is analogous to a Service Pack.

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 an up-to-date backup of your system before starting this section.

The application of corrective service to z/VM is covered in the *z/VM V5R4 Service Guide* and *VMSES/E Introduction and Reference*. Both of these documents can be downloaded starting at the following URL:

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

From that Web page, you can choose the type of media you want: z/VM Information Center, z/VM Bookmanager or PDF 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 completed, the PUT2PROD EXEC places the z/VM components, features, or products that are installed on the z/VM System deliverable, and were serviced, into production.
A good Web site to start at is
http://www.vm.ibm.com/service/

The body of the page should look similar to the following figure:

![Figure 5-1 z/VM Service main Web page](image)

You may want to consider viewing some of the links from this page.

## 5.1 Applying a PTF

This section is based on z/VM 6.1.

You may determine that you need to apply a specific fix or PTF to your system. For example, an Authorized Program Analysis Report (APAR), VM64670, was opened when Linux guests were hanging intermittently. The summary of the APAR is as follows:

**PROBLEM SUMMARY:** LINUX USER HUNG BECAUSE SVPBK LOCK HELD  
**USERS AFFECTED:** All users of z/VM running Linux guests.  
**PROBLEM DESCRIPTION:** Linux guests may become hung due to a problem in managing a lock word. This problem is timing-related and may occur intermittently.  
**PROBLEM CONCLUSION:** Lock word processing in HCPWED is updated to properly handle all possible states of the lock.

The APAR was assigned the following Programming Temporary Fix (PTF) numbers for each of the following z/VM releases:
So for z/VM 6.1, you want to apply PTF UM32811. Following is an example of how to do so.

### 5.1.1 Getting service using ShopzSeries

Service for z/VM is still available on the media of tape. However, getting service over the Internet is more convenient and becoming more common. Typically this is done with *IBM ShopzSeries*.

Click on the link **IBM ShopzSeries** under the *IBM Support Portals* heading on the main Service page as shown on Figure 5-1 on page 68. This should take you to the following URL:


From there you can search for an APAR if you have the APAR number. In Figure 5-2 on page 70, the first three steps to do this are shown:
Figure 5-2 Searching for a PTF on ShopzSeries

1. On the menu bar at the top, click on **Support and Downloads**, then choose **Search** in the drop-down menu. This is shown at the top of the figure.

2. In the **Support type** drop-down menu, choose **System z** and in the **Search** text area, type the APAR number, **VM64670**, in this example. This is shown in the middle of the figure.

3. If the APAR is found, you should see a link as a result. Click on that Link, **VM64670: LINUX USER HUNG**, in this example. This is shown at the bottom of the figure.

Clicking on the link should bring you to the APAR. In this example, you should find the information on APAR VM64670 that was summarized previously. At the top of the page, look for the section **A fix is available**. In this example, there is a fix available.

Farther down on the page, note the **Fixed component name** which is important. In this example it is **VM CP** shown near the bottom of Figure 5-3 on page 71.
At the bottom of the page the Applicable component levels section shows that PTF UM32811 is available for z/VM 6.1. Before getting that PTF, you may want to be sure that it has not already been applied.

### 5.1.2 Determining if a PTF has been applied

Check to make sure the PTF has not previously been applied. In this example it is known that the PTF is **UM32811** and the component is **VM CP**.

1. Because the description of the PTF cites a component name of “VM CP”, the component CP is used in the following command.

2. Use the **SERVICE** command to query whether the PTF has been applied:

   ```
   ==> service cp status um32811
   VMFSRV2760I SERVICE processing started
   VMFSRV1227I UM32811 is not received or applied to CP (6VMCPR10%CP)
   VMFSRV2760I SERVICE processing completed successfully
   ```

   This shows that PTF UM32811 has **not** been applied. The sections that follow describe how to obtain and apply it.

### 5.1.3 Downloading the service to z/VM

From the previous APAR web page search, the link for UM32811 is clicked on which results in a Web page that should be similar to the following:
Get zSeries related fixes

Order the fix using one of the services below. Not all services are available in every country. If you are not sure which service to use, we suggest you use ShopzSeries.

<table>
<thead>
<tr>
<th>ShopzSeries</th>
<th>IBMLink</th>
</tr>
</thead>
<tbody>
<tr>
<td>ShopzSeries - Electronic or physical delivery</td>
<td>IBM's productivity tool for planning and ordering zSeries software has been enhanced to allow a direct link to the fix ordering service.</td>
</tr>
<tr>
<td>Available worldwide. Service agreement required. At this time we cannot provide a direct link to the fix ordering service.</td>
<td></td>
</tr>
</tbody>
</table>

*Figure 5-4  Getting fixes from ShopzSeries on IBMLink*

1. In this example the link **ShopzSeries - Electronic or physical delivery** is clicked on. Sign into ShopzSeries with your IBM ID and follow the five self-explanatory steps to order your PTF. When you are finished, click on **Submit** to place your order.

2. You should receive an e-mail within a few minutes. It will have your order number a link to start the download of service files. Following is an example of the important information in the e-mail.

   From: Oms Client01/Boulder/IBM
   Subject: IBM Order <Bxxxxxxx> is ready for download.
   ...
   To access your order directly, go to:

3. Point your browser to the link in the e-mail. You should see a Web page similar to the following:

*Figure 5-5  Web page created for downloading a PTF*
4. Choose a method of downloading the **VMSES PTF Envelope** for your order to your desktop machine. You may also choose to download the **VMSES Documentation Envelope**.

5. There should be both a SES envelope (the PTF or PTFs themselves) and a documentation envelope. Copy both to z/VM in binary with fixed 1024 byte records to the MAINT 500 disk. Usually, FTP is used. The PTF 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 of FTPing from a DOS session:

```
C:\downloads> ftp 9.60.18.249
User (9.60.18.249:(none)): maint
Password:
...
ftp> cd maint.500
...
ftp> bin
...
ftp> quote site fix 1024
...
ftp> mput s8873674.*
mput S8873674.SHIPDOCS? y
...
ftp: 6144 bytes sent in 0.05Seconds 130.72Kbytes/sec.
mput S8873674.SHIPTFSS? y
...
ftp: 4096 bytes sent in 0.01Seconds 273.07Kbytes/sec.
ftp> quit
```

6. Logon to z/VM as MAINT.

7. Access the MAINT 500 disk as C:

```
=> acc 500 c
DMSACC724I 500 replaces C (2CC)
```

8. The envelope files arrive in a compressed format to speed downloads. In order to use them they must first be renamed to have a file type of SERVLINK and uncompressed with the DETERSE command. Therefore it is recommended to leave the file name of the SES envelope unchanged, but change the prefix letter of the documentation envelope to D. First rename them, then use the DETERSE command with the (REPLACE parameter to uncompress them in place and save disk space:

```
=> rename s8873674 shipftss c = servlink =
=> rename s8873674 shipdocs c d8873674 servlink =
=> dexter s8873674 servlink c = = (replace
=> dexter d8873674 servlink c = = (replace
```

Be sure all commands complete successfully.

### 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 minidisk with a lot of free space - that is what the MAINT 500 minidisk is for.

1. Access the MAINT 500 disk as file mode C:

```
=> acc 500 c
DMSACC724I 500 replaces C (2CC)
```
2. Use the SERVICE ALL command specifying the envelope files you downloaded. Many, many screens of output will scroll by and will automatically be cleared. Important messages will be saved to the 500 disk. This process may take many minutes. Following is an example:

```bash
==> service all s8873674
```

VMFSUT2760I VMFSUFTB processing started
VMFSUT2760I VMFSUFTB processing completed successfully
VMFSRV2760I SERVICE processing completed successfully

If you see no number in parenthesis after the Ready; prompt, then the return code is 0. Any non-zero return code will be in parenthesis. A return code of 0 is ideal. In general a return code of 4 is acceptable - it means that only warnings were issued. A return code of 8 or greater generally means that errors were encountered.

3. The output files are of the form $VMF* $MSGLOG. You may wish to inspect these files.

```bash
==> file $vmf* $msglog
```

<table>
<thead>
<tr>
<th>VMFSRV</th>
<th>MSGLOG</th>
<th>A1 V</th>
<th>80</th>
<th>728</th>
<th>14 12/15/09 13:43:34</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMFBLD</td>
<td>MSGLOG</td>
<td>A1 V</td>
<td>80</td>
<td>787</td>
<td>11 12/15/09 13:41:47</td>
</tr>
<tr>
<td>VMFAPP</td>
<td>MSGLOG</td>
<td>A1 V</td>
<td>80</td>
<td>252</td>
<td>4   12/15/09 13:41:37</td>
</tr>
<tr>
<td>VMFREC</td>
<td>MSGLOG</td>
<td>A1 V</td>
<td>80</td>
<td>56</td>
<td>1   12/15/09 13:41:36</td>
</tr>
<tr>
<td>VMFMRD</td>
<td>MSGLOG</td>
<td>A1 V</td>
<td>80</td>
<td>231</td>
<td>4   12/15/09 13:41:35</td>
</tr>
<tr>
<td>VMFP2P</td>
<td>MSGLOG</td>
<td>A1 V</td>
<td>80</td>
<td>805</td>
<td>15  11/19/09 13:52:09</td>
</tr>
<tr>
<td>VMFINS</td>
<td>MSGLOG</td>
<td>A1 V</td>
<td>80</td>
<td>163</td>
<td>3   11/19/09 13:47:25</td>
</tr>
</tbody>
</table>

4. 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
```

You are viewing ¬ST: messages from the LAST run.
Number of messages shown = 1 '<=>' Number of messages not shown = 3

Ideal there will be no output. If there are errors they must be addressed. If there are warnings, they may be acceptable but should be investigated.

5.1.5 Putting the service into production

Use the PUT2PROD command to put the service into production.

```bash
==> put2prod
```

VMFP2P2760I PUT2PROD processing completed successfully

Again, watch for a return code of 0. 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. If you are in a position to re-IPL your system it may be safest to reIPL using the SHUTDOWN REIPL command in order to completely test the changes:

```bash
==> shutdown reipl ilparms cons=sysc
```

SYSTEM SHUTDOWN STARTED
Your z/VM system should come back in a few minutes. When the system comes back up, start a 3270 session to MAINT and again query the status of the PTF:

```shell
==> service cp status um32811
VMFSRV2760I SERVICE processing started
VMFSRV1226I CP (6VMCPR10%CP) PTF UM32811 status:
VMFSRV1226I    RECEIVED  12/15/09 13:41:36
VMFSRV1226I    APPLIED   12/15/09 13:41:37
VMFSRV1226I    BUILT     12/15/09 13:42:14
VMFSRV1226I    PUT2PROD  12/15/09 13:47:59
VMFSRV2760I SERVICE processing completed successfully
```

This shows that the PTF has been successfully applied.

### 5.1.6 Checking for APARMEMO files

After you have applied PTFs, you should check for files with a file type of APARMEMO on the MAINT 500 disk. These files may have additional instructions on work to do after the PTFs have been applied.

```shell
==> listfile * aparmemo c
6VMCMS10 APARMEMO C1
```

Look at the contents of the file:

```shell
==> type 6vmcms10 aparmemo c

APAR MEMOS 01/26/10.12:50:20
=================================
THE FOLLOWING MEMOS WERE INCLUDED WITH THE PTFS SHIPPED:

NONE.
```

In this example the file was created, but no additional memos are present.

### 5.2 Applying a Recommended Service Upgrade or RSU

This section is based on z/VM 5.4 because at the time of writing there was no RSU available for z/VM 6.1 (other than the RSU shipped on the z/VM system deliverable which is applied at z/VM install time).

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. You may prefer to use the official z/VM documentation.

Following is an example of upgrading to a z/VM 5.4 RSU with the medium being files downloaded from the Internet.

The section that follows is a summary of applying service and also describes how to obtain service over the Internet using IBM ShopzSeries.

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

```shell
==> q cplevel
z/VM Version 5 Release 4.0, service level 0801 (64-bit)
```
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 “0801” means the first service level for the year 2008 has been applied. If you installed from DVD, that was on the second RSU DVD disc. As this section was written in late 2009, it seems clear that a new RSU will be available.

The overall steps in applying an RSU are as follow:

- “Getting service from the Internet” on page 76
- “Downloading the service files” on page 76
- “Creating a new MAINT minidisk” on page 77 (not usually required)
- “Receiving, applying, and building the service” on page 78
- “Putting the service into production” on page 80

5.2.1 Getting service from the Internet

An RSU is obtained by its PTF number. The PTF for the most current RSU is of the form UM97xyz where xyz is the z/VM version-release-modification level. So for z/VM 6.1 the RSU would be UM97610.

With ShopzSeries, knowing the PTF number is not necessary. If you know you want the latest RSU, you can get it directly, based on the version of z/VM you are running. Perform the following steps:

1. Point a Web browser to the z/VM Service page:
   http://www.vm.ibm.com/service/
2. Click on IBM ShopzSeries.
3. Click on Sign In. If you have an user ID and password, use that. If you do not, you can fill out the form to create an ID and password. You must have your customer number.
4. Click on create new software orders at the top menu.
5. Under the Package Category section, click on the z/VM - Service radio button and also choose RSU recommended service in the drop-down menu. Click Next.
6. There will be five screens of forms that are hopefully self-explanatory. On screen 3 of 5, choose the radio button that is applicable to your version of z/VM. In this example it was z/VM Version 5.4.0 Stacked 5405RSU (PTF UM97540).
7. On screen 4 of 5 choose Internet as the delivery mechanism.
8. On screen 5 of 5, complete the form and click Submit.
9. In a few minutes, you should get two e-mails - one for the core RSU and one for the PSP bucket (additional fixes that may have come out after the RSU).

5.2.2 Downloading the service files

In this example, the service files are staged on a desktop machine then copied to z/VM with FTP.

1. Download the files to your desktop or another staging system. This example has two files: the SHIPTFSS file is for the PSP bucket and the SHIPRSU1 file is for the RSU.
2. FTP the file to the MAINT 500 disk. Following is an example of FTPing from a DOS session:
   C:\Downloads>ftp 9.60.18.141
User (9.60.18.141:(none)): maint
Password:
ftp> cd maint.500
...
ftp> bin
...
ftp> quote site fix 1024
...
ftp> put s8873949.shiptfss
...
ftp> put s8873950.shiprsul
...
ftp> quit

3. Logon to MAINT. Access the MAINT 500 disk as file mode C. Query the disks:

```plaintext
==> acc 500 c
DMSACC724I 500 replaces C (2CC)
==> q disk
LABEL VDEV M STAT CYL TYPE BLKSZ FILES BLKS USED(~) BLKS LEFT BLK TOTAL
MNT191 191 A R/W 175 3390 4096 41 214-01 31286 31500
MNT5E5 5E5 B R/W 9 3390 4096 131 1290-80 330 1620
MNT500 500 C R/W 600 3390 4096 3 58248-54 49752 108000
MNT51D 51D D R/W 26 3390 4096 305 1574-34 3106 4680
MNT190 190 S R/O 100 3390 4096 691 14921-83 3079 18000
MNT19E 19E Y/S R/O 250 3390 4096 1021 28225-63 16775 45000
```

4. Deterse the files. The second file is large so it may take a few minutes to deterse:

```plaintext
==> deterse s8873949 shiptfss c = servlink =
==> deterse s8873950 shiprsul c = servlink =
DMSERD107S Disk C(500) is full
No traceback - not enough CTL storage
...
```

Usually this step should succeed, however, this RSU is unusually large, and thus too large to be decompressed onto the 500 disk. An extra step of creating a new disk is necessary.

### 5.2.3 Creating a new MAINT minidisk

This step is usually not necessary, however it is with stacked RSU 5405 for z/VM 5.4 because it is unusually large. If you have adequate space to DETERSE the files on the MAINT 500 disk, you can skip this section.

1. Create a new MAINT 501 disk for temporary storage of the uncompressed RSU by using 400 cylinders of space taken from the end of the W02 disk (volser is DV6156 in this example). Verify the disk layout is good, then bring the changes online with the DIRECTXA command:

```plaintext
==> x user direct c
...
USER MAINT LNX4VM 128M 1000M ABCDEFG
  AUTOLOG AUTOLOG1 OP1 MAINT
  ACCOUNT 1 SYSPROG
...
* add a new MAINT 501 disk for additional space for service files
  MDISK 501 3390 2371 400 DV6156  MR LNX4VM LNX4VM LNX4VM
...
==> diskmap user
...
==> x user diskmap
...
// check the report file for gaps or overlaps
2. Log off MAINT and log back on to load the new directory entry. An attempt is made to access the MAINT 500 and 501 disks as file mode C and F, respectively. However, the new 501 disk has never been formatted. Format it as F:

```
=> log
... // log back on
=> acc 500 c
DMSACC724I 500 replaces C (2CC)
=> acc 501 f
DMSACP112S F(501) device error
=> format 501 f
DMSFOR603R FORMAT will erase all files on disk F(501). Do you wish to continue? Enter 1 (YES) or 0 (NO).
1
DMSFOR605R Enter disk label:
mnt501
DMSFOR733I Formatting disk F
```

Now that a new MAINT 501 disk is available it can be used to stage the RSU file:

1. Move the large RSU file from the MAINT 500 (C) to the 501 (F) disk and query the disks:

```
=> copy s8873950 shipsul c = = f
=> erase s8873950 shipsul c
=> q disk
```

2. Deterse the RSU from the 501 disk (F) back to the 500 disk (C) and again query the disks:

```
=> deterse s8873950 shipsul f = servlink c
=> q disk
```

This shows that the MAINT 500 disk is now 91% full. The tersed file on the 501 disk is no longer necessary, but it is left there for reference.

### 5.2.4 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:

```
vmfmrdsk zvm cp apply (setup
vmfsetup zvm cp
```
Then the same steps were needed for many other components. The process is much easier now with the \texttt{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. Apply the service with the \texttt{SERVICE} command. The RSU should be applied first (\texttt{S8873950 SERVILINK} in this example). Then any PTFs that came after the RSU can be applied:

\begin{verbatim}
===> service all S8873950
... VMFSRV2760I SERVICE processing completed successfully for GCS BUILD
VMFSUT2760I VMFSUFTB processing started
VMFSUT2760I VMFSUFTB processing completed successfully
VMFSRV2760I SERVICE processing completed with warnings
Ready(00004); T=129.22/138.98 10:14:11
\end{verbatim}

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. View details with the \texttt{VMFVIEW} command:

\begin{verbatim}
===> vmfview service
You are viewing -ST: messages from the LAST run.
Number of messages shown = 12 <=> Number of messages not shown = 985
************************************************************************
****             SERVICE               USERID: MAINT                ****
************************************************************************
****            Date: 12/17/09            Time: 10:06:17            ****
************************************************************************
CK:VMFSUI2104I PTF UM32616 contains user information. Review the :UMEMO
CK: section in file UM32616 $PTFPART
CK:VMFSUI2104I PTF UM32616 contains user information. Review the :UMEMO
CK: section in file UM32616 $PTFPART
CK:VMFSUI2104I PTF UA46229 contains user information. Review the :UMEMO
CK: section in file UA46229 $PTFPART
CK:VMFSUI2104I PTF UA46229 contains user information. Review the :UMEMO
CK: section in file UA46229 $PTFPART
CK:VMFSUI2104I PTF UA46229 contains user information. Review the :UMEMO
CK: section in file UA46229 $PTFPART
CK:VMFSUI2104I PTF UA46229 contains user information. Review the :UMEMO
CK: section in file UA46229 $PTFPART
CK:VMFSUI2104I PTF UA46229 contains user information. Review the :UMEMO
CK: section in file UA46229 $PTFPART
WN:VMFBDC2250W The following OSA objects have been built on BUILD0 100
WN:            (L) and should be copied to your workstation:
WN:VMFBDC2250W IOAJAVA  BIN
CK:VMFSUI2104I PTF UM32654 contains user information. Review the :UMEMO
CK: section in file UM32654 $PTFPART
CK:VMFSUI2104I PTF UM32654 contains user information. Review the :UMEMO
CK: section in file UM32654 $PTFPART
WN:VMFBDC2250W The following VMHCD objects have been built on BUILD0 300
WN:            (J) and should be copied to your workstation:
WN:VMFBDC2250W EEq INST EXEBIN
\end{verbatim}

For these warnings, if you are running OSA or HCD then as the \texttt{VMFBDC2250W} message states you will need to copy the stated objects to your workstation at some point.
2. ReIPL CMS and re-access the MAINT 500 disk as C. Apply the PSP bucket (S8873949 in this example):

```plaintext
=> ipl cms
z/VM V5.4.0 2008-10-22 15:36

Ready; T=0.01/0.01 10:46:46
=> acc 500 c
DMSACC724I 500 replaces C (2CC)
=> service all S8873949
...
VMFSUT2760I VMFSUFTB processing completed successfully
VMFSRV2760I SERVICE processing completed successfully
```

You should see that the service was installed successfully.

### 5.2.5 Putting the service into production

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

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:

```plaintext
=> put2prod
...
VMFP2P2760I PUT2PROD processing completed successfully for SAVECMS
VMFP2P2760I PUT2PROD processing completed with warnings
Ready(00004); T=64.39/68.87 11:06:11
```

The return code was 4 in this example. Review the warning messages with the `VMFVIEW` `PUT2PROD` command:

```plaintext
==> vmfview put2prod

You are viewing ¬ST: messages from the LAST run.
No messages meet the search criteria.

************************************************************************
****             PUT2PROD              USERID: MAINT                ****
************************************************************************
****            Date: 12/17/09            Time: 10:59:51            ****
************************************************************************
WN:DTCPRD3043W   File LDAP-DS SCONFIG I has been updated; Its content
WN:            should be reviewed for changes that may affect your use of
WN:            this file
WN:DTCPRD3043W   File SMTPCMDX SAMPEXC I has been updated; Its content
WN:            should be reviewed for changes that may affect your use of
WN:            this file
WN:DTCPRD3043W   File SMTPCMDX SAMPASM I has been updated; Its content
WN:            should be reviewed for changes that may affect your use of
WN:            this file
WN:DTCPRD3021W TCP2PROD processing completed with RC = 4
```

With these warnings you should do as message DTCPRD3043W says and compare the files to see if you need to pick up any of the new changes in your running copy of the sample file.

Even though the service has been “put into production”, the `QUERY CLEVEL` command should still return the current service level; in this example 0801. This is because the new CP load module (nucleus) has not been invoked:

```plaintext
=> q cplevel
z/VM Version 5 Release 4.0, service level 0801 (64-bit)
```
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 0903:

```
=> shutdown reipl iplparms cons=sysc
HCPSHU960I System shutdown may be delayed for up to 330 seconds
Ready; T=0.01/0.01 11:12:32
... // after system comes back up a new 3270 session is started...
=> q cplevel
z/VM Version 5 Release 4.0, service level 0903 (64-bit)
```

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](http://www.vm.ibm.com/virtualnetwork/) and [Virtual Switch TCP/IP Maintenance Levels](http://www.vm.ibm.com/virtualnetwork/)

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

```
=> cp query vmlan
VMLAN maintenance level:
  Latest Service: VM64604
VMLAN MAC address assignment:
  MACADDR Prefix: 020003
  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 VM64604.

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

```
=> q vswitch
VSWITCH SYSTEM VSW1     Type: VSWITCH Connected: 0    Maxconn: INFINITE
  PERSISTENT  RESTRICTED    NONROUTER                 Accounting: OFF
    VLAN Unaware
    MAC address: 02-00-03-00-00-01
    State: Ready
    IPTimeout: 5         QueueStorage: 8
    Isolation Status: OFF
    RDEV: 1004.P00 VDEV: 1004 Controller: DTCVSW1
    RDEV: 1100.P00 VDEV: 1100 Controller: DTCVSW2  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:

```
=> netstat tcp dtcvsw1 level
```
VM TCP/IP Netstat Level 540       TCP/IP Server Name: DTCVSW1

IBM 2084; z/VM Version 5 Release 4.0, service level 0903 (64-bit), VM TCP/IP Level 540; RSU 0903 running TCPIP MODULE E2 dated 12/17/09 at 10:53
TCP/IP Module Load Address:  OOC21000

This shows information about the TCPIP MODULE. Use the TCPSLVL command and the complete file specification (TCPIP MODULE E in this example) to get more information. Of particular interest is the latest APAR applied to TCT00SD:

```bash
===> tcpslvl tcpip module e
DTCLVL3306I SLVL data obtained; file TCPIP SLVLDATA A created
===> x TCPIP SLVLDATA
SLVL TCPIP PK67610
...  
SLVL TCT00SD PK98608
... 
```

**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.
6.1 Downloading files associated with this book

This book has several files associated with it that will be needed to set your system up quickly. You can download the tar file on the Web at:
The tar file \texttt{virt-cookbook-S11.tgz} is only about 20KB. Download the file and untar it. The following example shows this being done from the directory \\texttt{/nfs/}:

\begin{verbatim}
# mkdir -p /nfs
# cd /nfs
... download or copy the file virt-cookbook-S11.tgz ...
# tar xzf virt-cookbook-S11.tgz
\end{verbatim}

This should create a new directory named \texttt{virt-cookbook-S11/}. List the files:

\begin{verbatim}
# ls -F virt-cookbook-S11/
  README.txt  clone.sh*  vm/
\end{verbatim}

There is a README file, a clone script, and a directory for z/VM files. You now have downloaded and uncharted the files associated with this book.

Make a directory \texttt{/nfs/sles11/}. This will hold the SLES 11 install ISO images:

\begin{verbatim}
# mkdir sles11
\end{verbatim}

\section*{6.2 Setting up a SLES 11 install server}

You may have a licensed version of SLES 11 on physical DVDs or you may choose to try an evaluation copy. There is an evaluation copy on the Web starting at the following URL:

\url{http://www.novell.com/products/server/eval.html}

Follow the link named \textit{SUSE Linux Enterprise Server 11 for IBM System z} and create an account to download the ISO images.

\subsection*{6.2.1 SLES 11 DVD ISO image file}

Following are the DVD ISO images for SLES 11 that were available at the time this book was written:

\begin{table}[h]
\centering
\caption{SLES 11 DVDs}
\begin{tabular}{|l|l|l|}
\hline
DVD number & File name & File size in bytes \\
\hline
1 & SLES-11-DVD-s390x-GM-DVD1.iso & 2,833,274,880 \\
2 & SLES-11-DVD-s390x-GM-DVD2.iso & 3,929,954,304 \\
\hline
\end{tabular}
\end{table}

The second DVD is not described in this book because it contains only source RPMs and is not needed.

If you are starting with a soft copy of the DVD 1 ISO image, \textbf{copy it} to the \texttt{/nfs/sles11/} directory now.

\subsection*{6.2.2 Starting from physical DVD}

\textbf{NOTE:} Perform this section \textit{only if} you are starting from physical DVD discs.

If you are starting with physical DVDs, you must first convert them to ISO images. This can be accomplished using the Linux \texttt{dd} command which basically does a byte-for-byte copy of the DVD contents.
Put the first DVD 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 or under /media directory) is the device file for the DVD drive.

Now copy the contents of the DVD to an ISO image using the `dd if` (input file) and `of` (output file) parameters. The following example copies the SLES 11 DVD to the appropriately named file. Sometimes, /dev/cdrom is automatically mounted over /mnt/cdrom when you put the DVD in the drive. If so, you will need to unmount it using the `umount` command after copying the contents of the DVD:

```
# cd /nfs/sles11
# dd if=/dev/cdrom of=SLES-11-DVD-s390x-GM-DVD1.iso
```

The DVD should start spinning and this will take a couple of minutes to copy. Repeat the `dd` command for each additional disc you have, using appropriate file names.

```
# umount /mnt/cdrom
# mount ... // next DVD
```

### 6.2.3 Verifying the ISO images

You should first verify the integrity of the ISO images. This is done using a file of checksum values and ISO file names. The checksums were calculated from the contents of the DVD. After downloading or `dd`ing the ISO images, the checksums are calculated again and compared against the original values using the `md5sum` command and the checksum files.

Following is an example of using the `md5sum` command against an MD5SUM file. When you obtain the DVDs, be they physical discs or ISO images, you should also obtain an MD5SUM file.

**SLES 10 MD5SUM values**

```
# cat MD5SUM
3e717006f08f9317c451fd0377041188  SLES-11-DVD-s390x-GM-DVD1.iso
b0d5a3b266ae102177788d76a127dff  SLES-11-DVD-s390x-GM-DVD2.iso
```

Use the `md5sum -c` command to verify the integrity of the ISO images. All should report OK. Following is an example:

```
# md5sum -c MD5SUM
SLES-11-DVD-s390x-GM-DVD1.iso: OK
SLES-11-DVD-s390x-GM-DVD2.iso: OK
```

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

### 6.2.4 Configuring the SLES 11 install server

With SLES 9 and earlier, creating an install tree became complicated with Service Packs. SLES 10 brought the ability to automount the DVDs. There are different possible methods of making the SLES 11 distribution available:

- By copying the contents of all DVDs into one directory
- By loop mounting each ISO image to different directory
- By using the ISO images directly

The last option is the easiest. The installation process will access the ISO images directly and mount them automatically.
The initial RAMdisk (initrd) and the kernel (vmrdr.ikr), will have to be copied to z/VM for the first Linux install. These files are on the first DVD in the /boot/ directory.

To allow access to these files, the first DVD must be loopback mounted using the \texttt{-o loop} flag to the \texttt{mount} command. First create a directory, \texttt{dvd1/}, as a mount point:

\begin{verbatim}
# cd /nfs/sles11
# mkdir dvd1
# mount -o loop SLES-11-DVD-s390x-GM-DVD1.iso dvd1
\end{verbatim}

You should now have a directory, \texttt{/nfs/}, with subdirectories of \texttt{sles11/} and \texttt{virt-cookbook-S11/}.

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 the NFS RPMs are 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 \texttt{nfs-utils}. If this RPM is not installed, then install it now.

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

\begin{verbatim}
# cd /etc
# cp exports exports.orig
# vi exports // add two lines at the bottom
/nfs/virt-cookbook-S11 *(ro,sync)
/nfs/sles11/dvd1 *(ro,sync)
\end{verbatim}

The \texttt{*(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 \texttt{man exports} for more details.

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

\begin{verbatim}
# chkconfig --list nfsserver
nfsserver  0:off 1:off 2:off 3:on  4:off  5:on  6:off
\end{verbatim}

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 \texttt{chkconfig} command and turn it on for the current run level with the \texttt{rcnfsserver start} command (if the NFS server is already running, you can use the \texttt{restart} parameter):

\begin{verbatim}
# chkconfig nfsserver on
# rcnfsserver start
Starting kernel based NFS server done
\end{verbatim}
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.

```bash
# ls /mnt
# mount localhost:/nfs/sles11/dvd1 /mnt
# ls -F /mnt
ARCHIVES.gz   README          gpg-pubkey-0dfb3188-41ed929b.asc  ls-lR.gz
COPYING       boot/           gpg-pubkey-307e3d54-481f30aa.asc  media.1/
COPYING.de    content         gpg-pubkey-3d25d3d9-36e12d04.asc  pubring.gpg
COPYRIGHT     content.asc     gpg-pubkey-7e2e3b05-4816488f.asc  suse/
COPYRIGHT.de  content.key     gpg-pubkey-9c800aca-481f343a.asc  suse.ins
ChangeLog     control.xml     gpg-pubkey-a1912208-446a0899.asc
INDEX.gz      directory.yast  gpg-pubkey-b37b98a9-486b702f.asc
NEWS          docu/           license.tar.gz
```

This shows that the SLES 11 install root directory is accessible. Now unmount it and test the virt-cookbook-S11/ directory:

```bash
# umount /mnt
# mount localhost:/nfs/virt-cookbook-S11 /mnt
# ls -F /mnt
README.txt  clone.sh*  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 in on Linux on System z.

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 ]
```

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:

```bash
# 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 ]
```
Chapter 7. Installing SLES11 on the cloner

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

Chapters 4, 5 and 6 must be completed before proceeding with this chapter.

In this chapter you will install SLES 11 onto the user ID S11CLONE, which is referred to as the cloner - This section describes how to configure the cloner which will provide the following functions:

- A SLES11 install server: a tree of the RPMs and other files needed for installation
- Cloning: the ability to copy a Linux system from a logged off source user ID to a target user ID with identical minidisks.
- NFS server: to export the install tree and possibly other data and directories.

To achieve this, perform the following steps:

1. “Creating the user ID S11CLONE” on page 89
2. “Updating AUTOLOG1’s PROFILE EXEC” on page 91
3. “Preparing SLES11 bootstrap files” on page 92
4. “Installing the cloner” on page 95
5. “Configuring the cloner” on page 107

7.1 Creating the user ID S11CLONE

In this section you will define the user ID S11CLONE. Think of this as defining a virtual PC with memory and small disks and a Network Interface Card (NIC), but no operating system (yet).

To accomplish this, perform the following steps:

1. Logon to MAINT
2. Edit the USER DIRECT file:

   ==> 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 using the INCLUDE statement. You used the existing profile TCPCMSU when you defined the LNXMAINT user.

3. 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 37-49:

```
037 ""PROFILE IBMDFLT
038 SPOOL 000C 2540 READER *
039 SPOOL 000D 2540 PUNCH A
040 SPOOL 000D 2540 PUNCH A
041 SPOOL 000E 1403 A
042 CONSOLE 009 3215 T
043 LINK MAINT 0190 0190 RR
044 LINK MAINT 019D 019D RR
045 LINK MAINT 019E 019E RR
046 LINK MAINT 0190 0190 RR
047 LINK MAINT 019E 019E RR
048 LINK MAINT 0190 0190 RR
049 ""
```

4. Edit the duplicated profile by deleting the two LINK MAINT 040x lines, and inserting the lines that are in bold text below:

```
PROFILE LNXDFLT
IPL CMS
MACHINE ESA 4
CPU 00 BASE
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 0190 0190 RR
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 (Note: some workloads will benefit from a second virtual CPU by adding another line with CPU 01)
4. Defines a virtual NIC connected to the VSWITCH starting at virtual address 600
5. Provides read access to the LNXMAINT 192 disk as the user's 191 disk
6. Provides read access to the TCPMAINT 592 disk, so that the user has access to TCPIP services such as an FTP client.

5. Go to the bottom of the file and add the definition for a new user ID named S11CLONE. This user ID is given class B privilege, 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 shown in this example with the labels of your DASD:

```
USER S11CLONE LNX4VM 512M 1G BDEG
INCLUDE LNXDFLT
```
Chapter 7. Installing SLES11 on the cloner

This Linux user ID will have the following minidisks:

- **100**: A single 3390-3 for the root file system.
- **101-104**: 4 3390-3s that will be used to create a logical volume mounted over /nfs/. This will store the SLES11 installation tree and the files associated with this book. Note that the 3390 at device address 6289 starts at cylinder 321 because the first 320 cylinders were used for the LNXMAINT 191 and 192 disks.

6. Go back to the top of the file and search for string \$ALLOC\$. Add cylinder 0 of each of the four new volumes to this dummy user ID so they don't show up as gaps.

7. Run **DISKMAP** to check for overlaps and gaps. You should only see the two gaps that are in the default **USER DIRECT** file. These are expected. Press F3 to quit.

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

---

7.2 Updating AUTOLOG1’s PROFILE EXEC

The new Linux user ID, S11CLONE, will need 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 so the cloner 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 S11CLONE to the section that grant access to the VSWITCH. Also add S11CLONE to the section that XAUTOLOG the Linux user IDs:

```bash
===> link autolog1 191 1191 mr
===> acc 1191 f
===> x profile exec f
/*********************/
/* Autolog1 Profile Exec */
/*********************/
cp xautolog tcpip' /* start up TCPIP */
cp xautolog vmservs'
cp xautolog vmservu'
cp xautolog vmservr'
cp xautolog dtcvsw1'
cp xautolog dtcvsw2'
cp set pf12 ret' /* set the retrieve key */
cp set mdc stor Om 128m' /* Limit minidisk cache in CSTOR */
cp set mdc xstore Om 0m' /* Disable minidisk cache in XSTOR */
cp set srm storbuf 300% 250% 200%' /* Overcommit memory */
cp set signal shutdown 300' /* Allow guests 5 min to shut down */

/* Grant access to VSWITCH for each Linux user ID */
cp set vswitch vsw1 grant S11CLONE' 

/* XAUTOLOG each Linux user that should be started */
cp xautolog S11CLONE'
cp logoff' /* logoff when done */
====>
```

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:

```bash
===> set vswitch vsw1 grant S11CLONE
Command complete
===> q vswitch acc
VSWITCH SYSTEM VSW1 Type: VSWITCH Connected: 0 Maxconn: INFINITE
    PERSISTENT RESTRICTED NONROUTER Accounting: OFF
    VLAN Unaware
    MAC address: 02-00-07-00-00-01
    State: Ready
    IPTtimeout: 5 QueueStorage: 8
    Isolation Status: OFF
    Authorized userids:
        SYSTEM S11CLONE
        RDEV: B440.P00 VDEV: B440 Controller: DTCVSW2
        RDEV: B424.P00 VDEV: B424 Controller: DTCVSW1 BACKUP
```

The new user ID, S11CLONE, has now been granted access to the VSWITCH VSW1 for this session and at IPL time, and it automatically logged on at IPL time.

### 7.3 Preparing SLES11 bootstrap files

To IPL a SLES11 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.
1. **Start an SSH session as root** on the NFS server.

2. Change directory to the mounted DVD. You should see a directory `boot/` where the kernel and RAMdisk are located:

   ```
   # cd /nfs/sles11/dvd1
   # ls -F
   ARChives.gz  README          gpg-pubkey-0dfb3188-41ed929b.asc  ls-1R.gz
   COPYING       boot/          gpg-pubkey-307e3d54-481f30aa.asc  media.1/
   COPYING.de    content        gpg-pubkey-3d25d3d9-36e12d04.asc  pubring.gpg
   COPYRIGHT     content.asc    gpg-pubkey-7e2e3b05-4816488f.asc  suse/
   COPYRIGHT.de  content.key    gpg-pubkey-9c800aca-481f343a.asc  suse.ins
   ChangeLog     control.xml    gpg-pubkey-a1912208-446a0899.asc
   INDEX.gz      directory.yast gpg-pubkey-b37b98a9-486b702f.asc
   NEWS          docu/          license.tar.gz
   ``

3. Change directory to `boot/s390x/` and invoke the `ls` command. You should see the initial RAMdisk and kernel named `initrd` and `vmrdr.ikr`.

   ```
   # cd boot/s390x
   # ls -l initrd vmrdr.ikr
   -r--r--r-- 1 root root 12897335 Mar  8  2009 initrd
   -r--r--r-- 1 root root  6761472 Mar  8  2009 vmrdr.ikr
   ``

4. FTP to z/VM and login as `LNXMAINT`. Copy the SLES11 kernel (the file `vmrdr.ikr` copied as `SLES11 KERNEL`) and the initial RAMdisk (the file `initrd` copied as `SLES11 INITRD`). These files must have a record format of fixed 80 byte records and be transferred in binary. 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:

   ```
   # ftp 9.60.18.249
   Connected to 9.60.18.249.
   220-FTPSERVE IBM VM Level 610 at GPOK249.ENDICOTT.IBM.COM, 08:59:44 EST THURSDAY 2009-11-26
   220 Connection will close if idle for more than 5 minutes.
   Name (9.60.18.249:root): lnxmaint
   331 Send password please.
   Password: lnx4vm
   230 LNXMAINT logged in; working directory = LNXMAINT 191
   Remote system type is z/VM.
   ftp> cd lnxmaint.192
   250 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 SLES11.KERNEL
   local: vmrdr.ikr remote: SLES11.KERNEL
   ...
   ftp> put initrd SLES11.INITRD
   local: initrd remote: SLES11.INITRD
   ...
   ftp> quit
   ``

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

6. Besides the kernel and RAMdisk that you just copied, the file `SLES11 EXEC` should exist on the LNXMAINT 192 disk. Use the `FILELIST` command to verify that the kernel and RAMdisk were copied in fixed-80 byte record format. You should see the following files:

   ```
   ==> file lsl sles11 * d
   LNXMAINT FILELIST A0 V 169 Trunc=169 Size=18 Line=1 Col=1 Alt=0
   Cmd Filename Filetype Fm Format Lrecl Records Blocks Date Time
   SLES11 INITRD D1 F 80 161217 3149 11/26/09 9:16:09
   ``
7. Quit by pressing F3.

8. Verify that the file `SLES11 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:

```plaintext
=> type sles11 exec d
/* EXEC to punch SLES11 install system to reader and IPL from it */
'CP SPOOL PUN *'
'CP CLOSE RDR'
'PUR RDR ALL'
'PUN SLES11 KERNEL * (NOH'
'PUN userid() PARM-S11 * (NOH'
'PUN SLES11 INITRD * (NOH'
'CH RDR ALL KEEP'
'IPL OCC CLEAR'
```

9. A sample parameter file named `SAMPLE PARM-S11` is provided to save typing. These are the parameters, especially networking values, that will be used to install SLES. Observe the contents using the `TYPE` command:

```plaintext
=> type sample parm-s11 d
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
HostIP=n.n.n.n Hostname=x.x.x.x.xxx
Gateway=n.n.n.n Netmask=n.n.n.n
Broadcast=n.n.n.n Layer2=0
ReadChannel=0.0.0.0600 WriteChannel=0.0.0.0601 DataChannel=0.0.0.0602
Nameserver=n.n.n.n portname=whatever portno=0
Install=nfs://n.n.n.n/nfs/sles11/dvd.iso
UseVNC=1 VNCPassword=xxxxxxx
InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
```

10. View the SLES 11 sample parameter file using the `TYPE` command:

```plaintext
=> type sample parm-s11 d
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
HostIP=9.60.18.224 Hostname=gpok224.endicott.ibm.com
Gateway=9.60.18.129 Netmask=255.255.255.128
Broadcast=9.60.18.255 Layer2=0
ReadChannel=0.0.0.0600 WriteChannel=0.0.0.0601 DataChannel=0.0.0.0602
Nameserver=9.0.2.11 portname=whatever portno=0
Install=nfs://9.60.18.133/nfs/sles11/SLES-11-DVD-s390x-GM-DVD1.iso
UseVNC=1 VNCPassword=12345678
InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
```

11. Copy the sample parameter file to a new file with the file name `S11CLONE`. Edit the file and update the networking variables with the values correct for your site. Refer to the worksheet in section 2.7.3, "Linux resources worksheet" on page 18. The fields you should change are in **bold-italics**. The examples used in this book are as follows:

```plaintext
=> copy sample parm-s11 d s11clone = =
=> x s11clone parm-s11 d
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
HostIP=9.60.18.224 Hostname=gpok224.endicott.ibm.com
Gateway=9.60.18.129 Netmask=255.255.255.128
Broadcast=9.60.18.255 Layer2=0
ReadChannel=0.0.0.0600 WriteChannel=0.0.0.0601 DataChannel=0.0.0.0602
Nameserver=9.0.2.11 portname=whatever portno=0
Install=nfs://9.60.18.133/nfs/sles11/SLES-11-DVD-s390x-GM-DVD1.iso
UseVNC=1 VNCPassword=12345678
InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
```

12. Save your changes with the `FILE` subcommand.

**Note:** The Hostname parameter was used by the install process in SLES9 and not used in SLES10. SLES11 again uses this parameter during installation. This parameter is also used in the `clone.sh` script which also reads the parameter file.

**Note:** SLES11 documentation states VNC password should be six to eight character. However, actual installation process requires it to be at least eight characters long. If it is shorter, the installer will prompt for the password.
Now you are ready to start the cloner installation.

7.4 Installing the cloner

In this section you will install the cloner Linux image onto the new user ID S11CLONE.

1. **Log onto S11CLONE**, the cloner user ID. You should see a virtual NIC being created at virtual address 600, and the common PROFILE EXEC running which creates two VDISKS for swap spaces at virtual addresses 300 and 301:

   00: **NIC 0600 is created**; devices 0600-0602 defined
   z/VM Version 6 Release 1.0, Service Level 0901 (64-bit),
   built on IBM Virtualization Technology
   There is no logmsg data
   FILES: 0003 RDR, NO PRT, NO PUN
   LOGON AT 14:55:00 EST THURSDAY 11/26/09
   z/VM V6.1.0 2009-11-19 13:47

   DMSACP723I A (191) R/O
   DMSACP723I C (592) R/O
   DIAG swap disk defined at virtual address 300 (64989 4K pages of swap space)
   DIAG swap disk defined at virtual address 301 (129981 4K pages of swap space)
   Do you want to IPL Linux from minidisk 100? y/n

2. Answer no (n) to the question asking you to IPL Linux from 100:

   Do you want to IPL Linux from minidisk 100? y/n
   n

3. You may want to look the PROFILE EXEC using the TYPE command:

   ```
   === type profile exec
   /* PROFILE EXEC for Linux virtual servers */
   'CP SET RUN ON'
   'CP SET PF11 RETRIEVE FORWARD'
   'CP SET PF12 RETRIEVE'
   'ACC 592 C'
   'SWAPGEN 300 524288' /* create a 256M VDISK disk swap space */
   'SWAPGEN 301 1048576' /* create a 512M VDISK disk swap space */
   'PIPE CP QUERY' userid() '| var user'
   parse value user with id . dsc .
   if (dsc = 'DSC') then /* user is disconnected */
   'CP IPL 100'
   else /* user is interactive -> prompt */
   do
   say 'Do you want to IPL Linux from minidisk 100? y/n'
   parse upper pull answer .
   if (answer = 'Y') then 'CP IPL 100'
   end /* else */
   ```

4. Run the SLES11 EXEC to purge the reader, punch the bootstrap files and IPL from the reader. You should see the Linux RAMdisk getting loaded into memory:

   ```
   ===> sles11
   00: NO FILES PURGED
   RDR FILE 0001 SENT FROM S11CLONE PUN WAS 0001 RECS 085K CPY 001 A NOHOLD NOKEEP
   RDR FILE 0002 SENT FROM S11CLONE PUN WAS 0002 RECS 0009 CPY 001 A NOHOLD NOKEEP
   RDR FILE 0003 SENT FROM S11CLONE PUN WAS 0003 RECS 161K CPY 001 A NOHOLD NOKEEP
   00: 0000003 FILES CHANGED
   00: 0000003 FILES CHANGED
   Initializing cgroup subsys cpuset
   Initializing cgroup subsys cpu
   ```
Linux version 2.6.27.19-5-default (geeko@buildhost) (gcc version 4.3.2 [gcc-4_3-branch revision 141291] (SUSE Linux) ) #1 SMP 2009-02-28 04:40:21 +0100
setup.1a06a7: Linux is running as a z/VM guest operating system in 64-bit mode...

Important: If you see the following output on your 3270 screen, there is a problem.

Activating manual setup program.

>>> Linuxrc v3.3.34 (Kernel 2.6.27.19-5-default) <<<

Main Menu

1) Start Installation
2) Settings
3) Expert
4) Exit or Reboot
>

Often, this is because the install program cannot successfully mount the NFS file system, or it can mount it, but cannot find the SLES 11 install tree. Also, this can result when there is not enough memory (though the example uses 512 MB, which should be plenty).

If you get these messages, you may want to immediately restart the install process with the **IPL 00C** command which IPLs from the reader. Watch the console messages carefully.

5. A VNC server process will be started. You should see the messages:

starting VNC server...
A log file will be written to: /var/log/YaST2/vncserver.log ...

***
*** You can connect to **9.60.18.224, display :1** now with vncviewer
*** Or use a Java capable browser on **http://9.60.18.224:5801/**
***

(When YaST2 is finished, close your VNC viewer and return to this window.)

*** Starting YaST2 ***

6. 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.

![VNC Authentication](image)

*Figure 7-1 VNC viewer through a Java-enabled browser*
In addition to a browser, you can also use a standalone VNC viewer if desired (RealVNC in this example). The server to connect to would be 9.60.18.224:1 in this example, as shown on the left side of Figure 7-2. Enter the password specified in the parameter file (12345678 in this example).

![VNC Viewer: Connection Details](image)

![VNC Authentication: 9.60.18.224:1](image)

**Figure 7-2  Using the VNC viewer**

7. You should see a window entitled *Welcome*. Select your Language and Keyboard. After reading Licence Agreement accept the License Terms and click **Next**.

8. At the *Disk Activation* window click the **Configure DASD Disks** button.

9. At the *DASD Disk Management* window:
   a. Click **Select or Deselect** for all seven read-write minidisks (100-104, 300 and 301).
   b. Click **Activate** on the *Perform Action* pop-up menu as shown on Figure 7-3 on page 98. The DASD will be activated quickly.
   c. Click **Select or Deselect** to deselect minidisks 300 and 301 so that five minidisks, 100-104, are selected.
   d. Click **Format** on the *Perform Action* pop-up menu.
   e. Click **OK** to the query to format 5 parallel disks and then **Yes** to confirm. The DASD will be formatted in parallel. This will take a number of minutes.
   f. Click **Next** when the formatting is complete.
   g. In the *Disk Activation* window click **Next**.
10. In the *Installation Mode* window, accept the default of **New installation** and Click **Next**.

11. In the *Clock and Time Zone* window, select your time settings and click **Next**.

12. In the *Installation Settings* window click **Partitioning** and the *Expert Partitioner* window will appear.

13. In the *System View* column click on **Hard disks** to expand a list of all available minidisks as shown in Figure 7-4.

![DASD Disk Management](image-url)

*Figure 7-3  Activating DASD on the cloner*
14. Select dasda in System View column and click on Add button to add a partition as shown in Figure 7-5 on page 100.

Figure 7-4   Expert partitioner - Hard Disks
15. The window Add Partition on /dev/dasda should appear.

16. Set partition size to Maximum Size and click Next.

17. On next screen, accept the defaults (Format partition, Ext3 file system, Mount partition and / (the root file system) as the Mount Point. See Figure 7-6 on page 101. Click Finish.
18. Back in the Expert Partitioner panel you should see dasda1 partition under the dasda disk.
19. Select dasdb in the left column and click Add button to add a partition to it.
20. Select Maximum Size for the partition and click Next.
21. The disk /dev/dasdb1 will be used in a logical volume. Select Do not format partition and Do not mount partition as shown in Figure 7-7 on page 102. Click Finish.
22. You should now see a /dev/dasdb1 in the Expert Partitioner window. **Repeat** the previous three steps for /dev/dasdc, /dev/dasdd and /dev/dasde. This will give you four partitions /dev/dasdb1-dasde1 for use with LVM.

23. Select **Volume Management** in **System View** column. There should be no volume groups defined as shown in “Volume Management” on page 103. Click **Add Volume Group** button.
24. In the resulting *Add Volume Group* window set the volume group name to `nfs-vg`. Accept the default *Physical Extent Size* of 4M. Click *Add All* button to add `dasdb-dasde` to a newly defined volume group. Click *Finish*.

25. Back in the *Expert Partitioner* window you should see `nfs-vg` under *Volume Management* in *System View*. Click on it and the *Logical Volumes* view will appear. There should be no logical volumes defined. Click on *Add* button to add a logical volume.

26. Enter `nfs-lv` as a logical volume name and click *Next* button.

27. Select *Maximum Size* (8.94 GB) and click *Next*.

28. Choose *Format Partition* and select a *File system* type of `ext2`. Specify a *Mount Point* of `/nfs`. Click *Finish*.

29. Back in the *Expert Partitioner* click on `nfs-lv` in the *System View* column. It will display a summary of the logical volume you just defined as shown in Figure 7-9 on page 104.

30. Click *Accept* to close the partitioner.
31. Back on the **Installation Settings** window, click **Software**. You may get a warning window about low disk space - this is OK, it will be fixed. Deselect all items except **Base Server**, **32bit Runtime Environment**, **Help and Support Documentation** and **Minimal System** as shown in Figure 7-10 on page 105. To deselect a box, click on it once. There is always a short delay between selecting/deselecting a group and refreshing disk usage window on the bottom right. Click **Ok**.
32. In **Expert** tab click on **Default Runlevel** and choose **3: Full multiuser with network** as shown in Figure 7-11 on page 105. Click **OK**. If you receive a VNC warning, click **Yes**.

33. In the **Installation Settings** window, click **Install**.

34. On the **Confirm Installation** window, click **Install**. This will begin the process of laying down RPMs onto your system. Copying the RPMs should take about 5-20 minutes depending on network and disk speeds. When copying of the RPMs is done, your VNC viewer session will close.

**Completing the cloner installation**

1. Go back to your 3270 session. You will see messages indicating the Linux image is being restarted. At the end of the reIPL, the VNC server is started again.
starting VNC server...
A log file will be written to: /var/log/YaST2/vncserver.log ...

***
*** You can connect to 9.60.18.224, display :1 now with vncviewer
*** Or use a Java capable browser on  http://9.60.18.224:5801/
***

(When YaST2 is finished, close your VNC viewer and return to this window.)

*** Starting YaST2 ***

2. Start the VNC viewer session again.
3. On the Password for root user window, set the root password and click Next.
4. On the Host and Domain Name window, the Hostname and Domain Name fields should be filled in by values from parameter file (S11CLONE PARM-S11). Uncheck the box Change Hostname via DHCP. Click Next.
5. On the Network Configuration window in the Firewall section click disable which will disable the firewall. Click Next.
6. On the Test Internet Connection window, if you do not have an Internet connection, select No, Skip This Test and click Next.
7. On the Installation Overview window, accept the defaults and click Next. A certificate will be created.
8. On the User Authentication Method window, accept the default of local (/etc/passwd) and click Next.
9. On the Add a new local user window, add a non-root user and click Next.
10. At the Release Notes window it is recommended that you read the information. Click Next.
11. At the Hardware Configuration window click Next.
12. At the Installation Complete window deselect the check box click Clone this system for Autoyast, then click Finish.
13. The VNC viewer session will end. Go back to the 3270 session and DISCONNECT using the #CP prefix:
   
   %>% #cp disc

You have now installed the cloner. You should be able to access the new system using SSH.

7.4.1 Verifying the installation

Start a new SSH session to the cloner as root. Verify some settings using the df -h and swapon -s commands. You should see the following file systems mounted and the two swap spaces:

```
# df -h
Filesystem  Size  Used  Avail  Use% Mounted on
/dev/dasd1   2.3G  1.1G  1.1G  51% /
udev          248M  140K  248M  1% /dev
/dev/mapper/nfs--vg-nfs--lv 8.8G  21M  8.4G  1% /nfs
```

```
# swapon -s
Filename    Type  Size  Used  Priority
/dev/dasd1  partition  259956  0    -1
/dev/dasd1  partition  519924  0    -2
```

...
Linux is now installed on the cloner. The next step is to configure it.

### 7.5 Configuring the cloner

Now that your cloner is installed, it must be configured. The following steps are involved:

- “Copying files to the cloner” on page 107
- “Resetting the install source location” on page 108
- “Configuring the NFS server” on page 109
- “Turning off unneeded services” on page 110
- “Applying service if necessary - online update” on page 111
- “Installing the cmsfs package” on page 112
- “Enabling the vmcp and cmm modules” on page 113
- “Turning system to halt on SIGNAL SHUTDOWN” on page 114
- “Rebooting the system” on page 115
- “Verifying the changes” on page 115

#### 7.5.1 Copying files to the cloner

The cloner can now be configured to replace the NFS (PC) server to make the SLES11 install tree and the files associated with this book available with NFS. Copy the following files from the NFS server to the cloner.

- The SLES 11 DVD ISO image(s) - the first ISO image is needed, the second is optional.
- The tar file associated with this book: `virt-cookbook-S11.tgz`

Perform the following steps:

1. Create the directory `/nfs/sles11/`:
   ```
   # cd /nfs
   # mkdir sles11
   ```
2. Change directory into the `sles11/` directory and copy the DVD ISO image(s) from the PC NFS server to this directory:
   ```
   # cd sles11
   # scp 9.60.18.133:/nfs/sles11/*.iso .
   The authenticity of host '9.60.18.133' (9.60.18.133) can't be established.
   Are you sure you want to continue connecting (yes/no)? yes
   Warning: Permanently added '9.60.18.133' (RSA) to the list of known hosts.
   Password:
   ```
   ```
   SLES-11-DVD-s390x-GM-DVD1.iso                100% 2702MB 14.5MB/s  04:32
   ```
3. Change directory to `/nfs/` and copy the files associated with this book from the PC NFS server to this directory:
   ```
   # cd /nfs
   # scp 9.60.18.133:/nfs/virt-cookbook-S11.tgz .
   ```
4. Untar the file `virt-cookbook-S11.tgz`:
   ```
   # tar xzf virt-cookbook-S11.tgz
   ```
5. Change into the newly-created directory `virt-cookbook-S11/` and view the files:
   ```
   # cd virt-cookbook-S11/
   # ls
   README.txt  clone.sh*  vm/
   ```
6. Copy the script `clone.sh` file to `/usr/local/sbin/`. It will be used later for cloning:

```bash
# cp clone.sh /usr/local/sbin
```

You have now copied the files you will need from the PC NFS server.

### 7.5.2 Resetting the install source location

When SLES 11 is installed, the location of the `install source` is remembered. In this case, it is the PC NFS server. Now that the DVD ISO image has been copied from the PC NFS server to the cloner, you can reset the install source location to point to `/nfs/sles11` in the local file system. To do this task, perform the following steps:

1. Invoke `yast`:

```bash
# yast
```

2. Accept the default of **Software** in the left column and select **Software Repositories** in the right column and press **Enter**.

3. At the top of the **Configured Software Repositories** panel, you should see a single entry for SLES 11. Add a new installation source by using the Tab key to move to the **Add** button at the bottom and press **Enter**.

   - **Configured Software Repositories**

   | ---Services | Priority | Enabled | Autorefresh | Name               |
   | ---Repositories | 99 (Default) | x | x | SUSE-Linux-Enterprise-Server-11 11-0 |

   ```
   SUSE-Linux-Enterprise-Server-11 11-0
   URL: nfs://9.60.18.133/nfs/sles11/dvd1
   Category: YaST
   Properties
   [x] Enabled              Priority
   [x] Automatically Refresh   v 99^ 
   ```

   ```
   [Add][Edit][Delete] [GPG Keys...][Refreshâ]
   ```

4. Use the Tab key to move to **Local ISO Image**... and press **Enter** to select. Move to the **Next** field and press **Enter**.

   - ( ) Scan Using SLP...
   - ( ) Specify URL...
   - ( ) FTP...
   - ( ) HTTP...
   - ( ) HTTPS...
   - ( ) SMB/CIFS
   - ( ) NFS...
   - ( ) CD...
   - ( ) DVD...
   - ( ) Hard Disk...
   - ( ) USB Mass Storage (USB Stick, Disk)...
   - ( ) Local Directory...
   - (x) Local ISO Image...
5. Set the Path to ISO Image to /nfs/sles11/SLES-11-DVD-s390x-GM-DVD1.iso. Tab to Next and press Enter:

Repository Name
________________________________________________________
Path to ISO Image
/nfs/sles11/SLES-11-DVD-s390x-GM-DVD1.iso------------------------[Browse...]

6. In the License Agreement panel, move to Yes and press Enter to select. Move to Next and press Enter:

(x) Yes, I Agree to the License Agreement
( ) No, I Do Not Agree

7. You should now see the Configured Software Repositories panel again, this time with two entries. Information about selected row is described below. Find out, which repository is the old one (on PC NFS) and delete it.

8. Confirm the delete by choosing Yes.

9. Finish with Software Repositories dialog by pressing F10 key.

10. Leave YaST.

7.5.3 Configuring the NFS server

To configure the NFS server to make the SLES 11 install directory available using NFS, perform the following steps.

1. The NFS server will be used to export the SLES11 ISO image. The service needed is named nfsserver. Install it with yast -i command:

```
# yast -i nfs-kernel-server
```

2. Add the directory /nfs to the /etc/exports file. This will allow /nfs/sles11/ to be mounted, and any other directory under /nfs/:

```
# cd /etc
# vi 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.
/nfs      *(ro,sync,no_subtree_check)
```

You may want to look at the /etc/exports man page with the command man exports to understand the hosts and options specified.

3. Turn on the service using the chkconfig command:

```
# chkconfig nfsserver on
```
When you restart the cloner later, the NFS server will be started.

4. Start the NFS server for this session:

```
# rcnfsserver start
```

You can retire PC NFS server at this point because all necessary files are on S11CLONE NFS server.
7.5.4 Turning off unneeded services

There are a number of services which are started in a SLES11 minimum system. They can be viewed using the following `chkconfig` command:

```bash
# chkconfig -l | grep 3:on
```

```
cron          0:off 1:off 2:on 3:on 4:off 5:on 6:off
dbus          0:off 1:off 2:on 3:on 4:off 5:on 6:off
earlysyslog   0:off 1:off 2:on 3:on 4:off 5:on 6:off
fbset         0:off 1:on 2:on 3:on 4:off 5:on 6:off
haldaemon     0:off 1:off 2:on 3:on 4:off 5:on 6:off
irq_balancer  0:off 1:on 2:on 3:on 4:off 5:on 6:off
network       0:off 1:off 2:on 3:on 4:off 5:on 6:off
network-remotefs 0:off 1:off 2:on 3:on 4:off 5:on 6:off
nfs           0:off 1:off 2:off 3:on 4:off 5:on 6:off
nfsserver     0:off 1:off 2:off 3:on 4:off 5:on 6:off
nscl          0:off 1:off 2:off 3:on 4:off 5:on 6:off
postfix       0:off 1:off 2:off 3:on 4:off 5:on 6:off
random        0:off 1:off 2:on 3:on 4:off 5:on 6:off
rpcbind       0:off 1:off 2:off 3:on 4:off 5:on 6:off
smartd        0:off 1:off 2:on 3:on 4:off 5:on 6:off
splash        0:off 1:on 2:on 3:on 4:off 5:on 6:off $:on
splash_early  0:off 1:off 2:on 3:on 4:off 5:on 6:off
sshd          0:off 1:off 2:off 3:on 4:off 5:on 6:off
syslog        0:off 1:off 2:off 3:on 4:off 5:on 6:off
xinetd        0:off 1:off 2:off 3:on 4:off 5:on 6:off
```

In order to keep the cloner as lean as possible in terms of CPU usage, some of these services can be turned off. Turn off the following services using the `chkconfig` command:

```bash
# chkconfig fbset off
# chkconfig network-remotefs off
# chkconfig postfix off
# chkconfig splash off
# chkconfig splash_early off
# chkconfig smartd off
# chkconfig xinetd off
```

You can review which services are now configured to start in run level 3 with the following command:

```bash
# chkconfig -l | grep 3:on
```

```
cron          0:off 1:off 2:on 3:on 4:off 5:on 6:off
dbus          0:off 1:off 2:on 3:on 4:off 5:on 6:off
earlysyslog   0:off 1:off 2:on 3:on 4:off 5:on 6:off
haldaemon     0:off 1:off 2:on 3:on 4:off 5:on 6:off
irq_balancer  0:off 1:on 2:on 3:on 4:off 5:on 6:off
network       0:off 1:off 2:on 3:on 4:off 5:on 6:off
nfs           0:off 1:off 2:off 3:on 4:off 5:on 6:off
nfsserver     0:off 1:off 2:off 3:on 4:off 5:on 6:off
nscl          0:off 1:off 2:off 3:on 4:off 5:on 6:off
postfix       0:off 1:off 2:off 3:on 4:off 5:on 6:off
random        0:off 1:off 2:on 3:on 4:off 5:on 6:off
rpcbind       0:off 1:off 2:off 3:on 4:off 5:on 6:off
sshd          0:off 1:off 2:off 3:on 4:off 5:on 6:off
syslog        0:off 1:off 2:off 3:on 4:off 5:on 6:off
```

SLES11 configures a VNC connection which starts by default. It is configured through the `xinetd` daemon. It is recommended that you disable it although you disabled the `xinetd` service in the previous paragraph. It is just for a case you decide to use another service provided by `xinetd` later.
Edit the file `/etc/xinetd.d/vnc`. In sections `service vnc` and `service vnchttpd` add the parameter `disable=yes`.

```bash
# cd /etc/xinetd.d
# vi vnc

service vnc {
    socket_type = stream
    protocol = tcp
    wait = no
    user = nobody
    server = /usr/bin/Xvnc
    server_args = -noreset -inetd -once -from localhost -query localhost -geometry 1024x768 -depth 16
    type = UNLISTED
    port = 5901
    disable = yes
}

... 

service vnchttpd {
    socket_type = stream
    protocol = tcp
    wait = no
    user = nobody
    server = /usr/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 (this can be verified with the `chkconfig --list` command). So you can turn off the `xinetd` service with the `service xinetd stop` command for this session and the `chkconfig` command at boot time:

```bash
# service xinetd stop
Shutting down xinetd: done
# chkconfig xinetd off
```

### 7.5.5 Applying service if necessary - online update

You may want to apply service using YaST Online Update. Internet access was not available during the writing of this book, so step-by-step details are not available.

If you have access to the Internet, or an online update source, invoke `yast => Software => Online update`.

While writing this book problems were discovered with `chccwdev` commands which made the system non-responsive at times. The kernel was upgraded manually. The steps are shown here for a reference, it is recommended that you update the default kernel of SLES 11.

```bash
# ls -l
total 11240
-rw-r--r-- 1 root root 4577354 Oct 26 14:24 kernel-default-2.6.27.29-0.1.1.s390x.rpm
-rw-r--r-- 1 root root 5962759 Oct 26 14:25 kernel-default-base-2.6.27.29-0.1.1.s390x.rpm
-rw-r--r-- 1 root root 767921 Oct 26 14:25 kernel-default-man-2.6.27.29-0.1.1.s390x.rpm
-rw-r--r-- 1 root root 139411 Oct 26 14:25 module-init-tools-3.4-70.6.1.s390x.rpm
```
The Virtualization Cookbook for SLES 11

7.5.6 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 SLES11 distribution. Install it using the yast -i command:

```sh
# yast -i cmsfs
```

You should see some screens flash by as the cmsfs RPM is installed:

```sh
# rpm -q cmsfs
cmsfs-1.1.8-6.2
```

To test that the cmsfs package is properly installed, see if you can read the S11CLONE PARM=S11 parameter file. First you need to bring the 191 disk online using the chccwdev -e command. Then view the DASD that the system knows about using the lsdasd command:

```sh
# chccwdev -e 191
Setting device 0.0.0191 online
Done
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
```

Check the current kernel level and reboot:

```sh
# uname -r
2.6.27.19-5-default
# reboot
Broadcast message from root (pts/0) (Tue Dec 8 09:57:51 2009):
The system is going down for reboot NOW!
```

When the system comes back up, you should see an upgraded kernel:

```sh
# uname -r
2.6.27.29-0.1-default
```
Test some of the cmsfs utilities. The cmsfst command lists files on a minidisk:

```
# cmsfst -d /dev/dasdh
FILENAME FILETYPE FM FORMAT LRECL       RECS     BLOCKS     DATE     TIME
DIRECTOR P0 F         64         11          1 11/26/2009 15:07:13
ALLOCMAP P0 F       4096          2          2 11/26/2009 15:07:13
CHPW610  XEDIT    B1 V         72        189          3 11/20/2009 13:45:51
CPFORMAT EXEC     B1 V         79        252          3 11/20/2009 13:45:51
PROFILE EXEC     B1 V         63         17          1 11/26/2009 14:54:34
SAMPLE  PARM-S11 B1 V         80         11          1 11/20/2009 13:45:51
SLES11  EXEC     B1 V         68         10          1 11/20/2009 13:45:52
SWAPGEN  EXEC     B1 V         72        467          6 11/20/2009 13:45:52
```

The cmsfscat command types the contents of a file:

```
# cmsfscat -d /dev/dasdh -a s11clone.parm-s11
ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
HostIP=9.60.18.224 Hostname=gpok224.endicott.ibm.com
Gateway=9.60.18.129 Netmask=255.255.252.0
Broadcast=9.60.18.255 Layer2=0
ReadChannel=0.0.0600  WriteChannel=0.0.0601  DataChannel=0.0.0602
Nameserver=9.0.2.11 Portname=whatever Portno=0
Install=nfs://9.60.18.133/nfs/sles11/dvd1/SLES-11-DVD-s390x-GM-DVD1.iso
UseVNC=1  VNCPassword=12345678
InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
```

7.5.7 Enabling the vmcp and cmm modules

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.

The cmm module allows Linux to do cooperative memory management (aka CMM1) with z/VM. This feature must be enabled on both the Linux and z/VM sides for it to function (hence the name cooperative:).

To configure the vmcp and cmm modules to be loaded at boot time, edit the file /etc/sysconfig/kernel and add the module names 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
# You will find a few default modules for hardware which
```
# cannot be detected automatically
#
MODULES_LOADED_ON_BOOT="vmcp cmm"
...

Save the file and you should be able to issue CP commands using the vmcp after your system is rebooted.

### 7.5.8 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:

```bash
# 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.

### 7.5.9 Modifying zipl.conf

Two changes are recommended to the default zipl.conf file:

1. Add the parameters `vmpoff=LOGOFF` and `vmhalt=LOGOFF`. These instruct the z/VM user ID to be logged off when Linux is shut down. This can be convenient for shutting the z/VM system down more efficiently and also for getting a refreshed 3270 emulator session.

2. The `timeout` value is modified to 3 seconds. When SLES 11 boots, the default time that is allowed to enter a menu command is 10 seconds. Because the default value is most commonly used, this 10 seconds delays the starting of Linux. It is backed up to three seconds (you might want to choose five if you’re not a fast typist :))

Make a backup copy of the file and add the string `vmpoff=LOGOFF` to the parameter line in the `[Linux]` section and set `timeout=3` in the `menu` section:

```bash
# cd /etc
# cp zipl.conf zipl.conf.orig
# vi zipl.conf
// add the dasd= string
### Don't change this comment - YaST2 identifier: Original name: linux###
[Linux]
  image = /boot/image-2.6.27.19-5-default
  target = /boot/zipl
  ramdisk = /boot/initrd-2.6.27.19-5-default,0x200000
  parameters = "root=/dev/disk/by-path/ccw-0.0.0100-part1 vmpoff=LOGOFF vmhalt=LOGOFF
TERM=dumb init=/linuxrc"

  ;menu
  default = 1
  prompt = 1
  target = /boot/zipl
  timeout = 3
```
1 = Linux
2 = ipl
3 = Failsafe

###Don't change this comment - YaST2 identifier: Original name: ipl###

[ipl]
image = /boot/image
target = /boot/zipl
ramdisk = /boot/initrd,0x2000000
parameters = "root=/dev/disk/by-path/ccw-0.0.0100-part1 TERM=dumb"

###Don't change this comment - YaST2 identifier: Original name: failsafe###

[Failsafe]
image = /boot/image-2.6.27.19-5-default
target = /boot/zipl
ramdisk = /boot/initrd-2.6.27.19-5-default,0x2000000
parameters = "root=/dev/disk/by-path/ccw-0.0.0100-part1 TERM=dumb x11failsafe"

Now, run the `zipl` command so the changes are written to the boot record:

```
# zipl
Using config file '/etc/zipl.conf'
Building bootmap in '/boot/zipl'
Building menu 'menu'
Adding #1: IPL section 'Linux' (default)
Adding #2: IPL section 'ipl'
Adding #3: IPL section 'Failsafe'
Preparing boot device: dasda (0100).
Done.
```

These changes will be utilized the next time Linux is rebooted.

**Note:** Previous versions of this book recommended to list the minidisks to activate at boot time in the `zipl.conf` file. On SLES11 this approach does not work anymore. If you want to add DASD to the system, use the `dasd_configure` command instead of two step process using `chccwdev` and modifying `zipl.conf`.

If you want to check which hardware devices are activated at IPL, look in the `/etc/udev/rules.d/` directory at the `51-*` files.

### 7.5.10 Rebooting the system ###

You should now reboot the system to test the changes:

```
# reboot
```

After your system comes back, start a new SSH session.

### 7.5.11 Verifying the changes ###

You are now done customizing the cloner Linux image. **SSH back into the cloner** and check a few settings. Test the `vmcp` command with a CP command such as `QUERY NAMES`:

```
# vmcp q n
FTPSERVE - DSC , DTCVSW2 - DSC , DTCVSW1 - DSC , VMSERVER - DSC
VMSERVERU - DSC , VMSERVERS - DSC , TCPIP - DSC , OPERSYMP - DSC
DISKACNT - DSC , EREP - DSC , OPERATOR - DSC , S11CLONE -L0005
```
Confirm that both of your swap spaces are operational:

```
# 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/dasdf1</td>
<td>partition</td>
<td>259956</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>/dev/dasdg1</td>
<td>partition</td>
<td>519924</td>
<td>0</td>
<td>-2</td>
</tr>
</tbody>
</table>

Verify the NFS server is running:

```
# rcnfsserver status
```

Checking for kernel based NFS server: idmapd running
mountd running
statd running
nfsd running

Test mounting the SLES 11 install directory locally:

```
# ls /mnt
# mount localhost:/nfs/sles11 /mnt
# ls /mnt
SLES-11-DVD-s390x-GM-DVD1.iso  SLES-11-DVD-s390x-GM-DVD2.iso
```

This shows that you can mount the SLES 11 directory. Now unmount it:

```
# umount /mnt
```

Congratulations! You have installed and configured a SLES 11 Linux system onto the cloner. The next step is to install and configure the golden image.
Installing SLES11 on the golden image

“I never think of the future. It comes soon enough.”
— Albert Einstein

This chapter describes how to install SLES 11 onto the user ID S11GOLD, which is referred to as the golden image. Chapters 4, 5, 6 and 7 must be completed before proceeding. The golden image is the copy of Linux that will be cloned. Normally the system is shut down and the user ID logged off because it is not recommended to clone a running Linux system.

In this example, the golden image is given two 3390-3s at minidisk addresses 100 and 101. This allows for about 4.5GB of disk space for each Linux system. If you want to increase that size, it is recommended that larger volumes such as 3390-9s be used, but that minidisks at addresses 100 and 101 still be used. This will allow the clone.sh script to continue to function.

To install and configure the golden image, perform the following steps:

- “Creating the S11GOLD user ID” on page 117
- “Creating the S11GOLD parameter file” on page 119
- “Updating AUTOLOG1’s PROFILE EXEC” on page 119
- “Installing the golden image” on page 120
- “Configuring the golden image” on page 135

8.1 Creating the S11GOLD user ID

The golden image has a default memory size of 512 MB, and it is given just class G privilege. It is given the following minidisks:

- 100 Half of the disk space for the golden image.
- 101 The other half of the disk space.

To accomplish this, perform the following steps:
1. **Logon to MAINT**

2. Edit the USER DIRECT file, add 6 new lines at the bottom of the file and create the following user directory entry. Set the 3390 disk labels to those appropriate for your system. In this example, TM63CE is used:

```plaintext
=> x user direct c
=====> bot
=====> a 6
*
USER S11GOLD LNX4VM 512M 1G G
  INCLUDE LNXDFLT
  OPTION LNKNOPAS APPLMON
  MDISK 100 3390 3339 3338 TM63CE MR LNX4VM LNX4VM LNX4VM
  MDISK 101 3390 6677 3338 TM63CE MR LNX4VM LNX4VM LNX4VM
```

3. In our case, we use the DASD which was already defined in $ALLOC$ section. If you used a new DASD, go back to the top of the file and search for string USER $ALLOC$. Add cylinder 0 of each of the two 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 610RES R
  MDISK A02 3390 000 001 TV6283 R
  MDISK A03 3390 000 001 TV6284 R
  MDISK A04 3390 000 001 TM6289 R
  MDISK A05 3390 000 001 TM6290 R
  MDISK A06 3390 000 001 TM6293 R
  MDISK A07 3390 000 001 TM6294 R
  MDISK A08 3390 000 001 TM63CE R
  MDISK A09 3390 000 001 ...... R
...
=====> file
```

4. Run DISKMAP to check for overlaps and gaps. You should see a 501 and a 1 cylinder gap.

```plaintext
==> diskmap user
==> x user diskmap
=====> all /gap/ /overlap/
=====> pre off
0     500     501     GAP
------------------------  6 line(s) not displayed ------------------------
0       0       1     GAP
------------------------  330 line(s) not displayed ------------------------
=====> quit
```

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

```plaintext
==> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 6 RELEASE 1.0
EOJ DIRECTORY UPDATED AND ON LINE
HCPDIR4941 User directory occupies 44 disk pages
```

You have now defined the user ID that will contain the Linux golden image.
8.2 Creating the S11GOLD parameter file

A SLES 11 parameter file will be needed for this new user ID. You will need to change the IP address (HostIP variable) and the host name (Hostname variable). In this example those are 9.60.18.223 and gpok223.

Also, the cloner is used to provide the SLES 11 install directory using NFS by setting the Install variable.

Perform the following steps:

- Logon to LNXMAINT.
- Copy the S11CLONE parameter file to one with a file name of S11GOLD on the LNXMAINT 192 (D) disk:

  ```
  ==> copy s11clone parm-s11 d s11gold = =
  ```

- Edit the new file and set the networking values correctly:

  ```
  ==> x s11gold parm-s11
  ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
  HostIP=9.60.18.223 Hostname=gpok223.endicott.ibm.com
  Gateway=9.60.18.129 Netmask=255.255.252.0
  Broadcast=9.60.18.255 Layer2=0
  ReadChannel=0.0.0600 WriteChannel=0.0.0601 DataChannel=0.0.0602
  Nameserver=9.0.2.11 Portname=whatever Portno=0
  Install=nfs://9.60.18.224/nfs/sles11/SLES-11-DVD-s390x-GM-DVD1.iso
  UseVNC=1 VNCPassword=12345678
  InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
  ```

  These changes set the golden images IP address and host name, and also point to the new install server on the cloner (9.60.18.224 in this example).

When S11GOLD is logged onto, this new parameter file will be accessible on it's A (191) disk.

8.3 Updating 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.

Perform the following steps

- Logon to AUTOLOG1.
- Issue the ACCESS (NOPROF command so the PROFILE EXEC is not run:

  ```
  ==> acc (noprof
  ```

- Add a line to grant the S11GOLD user ID access to the VSWITCH:

  ```
  ==> x profile exec
  ...
  /* Grant access to VSWITCH for each Linux user */
  'cp set vswitch vsw1 grant s11clone'
  'cp set vswitch vsw1 grant s11gold'
  /* XAUTOLOG each Linux user that should be started */
  'cp xautolog s11clone'

  '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:

```bash
==> set vswitch vswl grant s11gold
Command complete
```

### 8.4 Installing the golden image

You should now be ready to begin the install onto the golden image. Linux will be installed onto the 100-101 minidisks. It will use 300-301 virtual disks for swapping. Most Linux user IDs described in this book will have two read/write minidisks and two VDISKS. Both swap disks are VDISKS, which means they are in z/VM virtual memory and thus provide fast access. Disk 300 is 256MB and will act as a primary swap space. Only after it is full, disk 301 which is 512 MB will be used. In a production environment there may be also another real DASD used as a swap device with the lowest priority; just in case.

To install the golden image, perform the following steps:

1. **Logon to S11GOLD.** When you logon, you should see message indicating that a virtual NIC has been created at address 0600 and that a VDISKs 300 and 301 have been created.

   ```
   LOGON S11GOLD
   NIC 0600 is created; devices 0600-0602 defined
   z/VM Version 6 Release 1.0, Service Level 0901 (64-bit),
   built on IBM Virtualization Technology
   There is no logmsg data
   FILES:   NO RDR,   NO PRT,   NO PUN
   LOGON AT 18:36:35 EST SATURDAY 11/28/09
   z/VM V6.1.0 2009-11-19 13:47
   DMSACP7231 A (191) R/O
   DMSACP7231 C (592) R/O
   DIAG swap disk defined at virtual address 300 (64989 4K pages of swap space)
   DIAG swap disk defined at virtual address 301 (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 minidisk 100? y/n
   ==> n
   ```

3. **Before you install Linux, it is good to verify the resources.** Verify there are minidisks at virtual addresses 100 and 101 and virtual disks at addresses 300 and 301 using the `QUERY` command:

   ```
   ==> q 100-101
   DASD 0100 3390 TM63CE R/W 3338 CYL ON DASD 63CE SUBCHANNEL = 0000
   DASD 0101 3390 TM63CE R/W 3338 CYL ON DASD 63CE SUBCHANNEL = 0001
   ==> q 300-301
   DASD 0300 9336 (VDSK) R/W 524288 BLK ON DASD VDSK SUBCHANNEL = 000E
   DASD 0301 9336 (VDSK) R/W 1048576 BLK ON DASD VDSK SUBCHANNEL = 000F
   ```

4. **Verify that you have a virtual OSA at addresses 600-602 with the `QUERY VIRTUAL OSA` command:**

   ```
   ==> q v osa
   OSA 0600 ON NIC 0600 UNIT 000 SUBCHANNEL = 0002
       0600 DEVTYPE OSA CHPID 00 OSD
       0600 MAC 02-00-07-00-00-03 CURRENT
       0600 QDIO-ELIGIBLE QIOASSIST-ELIGIBLE
   OSA 0601 ON NIC 0600 UNIT 001 SUBCHANNEL = 0003
       0601 DEVTYPE OSA CHPID 00 OSD
   ```
5. Use the `QUERY VIRTUAL STORAGE` command to show that you have a 512MB machine:

```bash
>>> q v stor
00: STORAGE = 512M
```

This shows that you have the resources necessary to install SLES 11.

### 8.4.1 Begin the SLES11 installation

Follow these steps to begin the installation of S11GOLD.

1. Run the `SLES11 EXEC`. You should see many screens of questions and answers scrolling by. If you had used the default parameter file shipped with S11GOLD, you would have had to answer all the networking questions manually. With the proper parameters set in the file S11GOLD PARM-S11, the install process should proceed to where you access the install program using a VNC client:

```bash
>>> sles11
sles11
  NO FILES PURGED
RDR FILE 0001 SENT FROM S11GOLD PUN WAS 0001 RECS 085K CPY 001 A NOHOLD NOKEEP
RDR FILE 0002 SENT FROM S11GOLD PUN WAS 0002 RECS 0009 CPY 001 A NOHOLD NOKEEP
RDR FILE 0003 SENT FROM S11GOLD PUN WAS 0003 RECS 161K CPY 001 A NOHOLD NOKEEP
0000003 FILES CHANGED
0000003 FILES CHANGED
Initializing cgroup subsys cpuset
Initializing cgroup subsys cpu
Linux version 2.6.27.19-5-default (geeko@buildhost) (gcc version 4.3.2 [gcc-4_3-
branch revision 141291] (SUSE Linux) ) #1 SMP 2009-02-28 04:40:21 +0100
setup.1a06a7: Linux is running as a z/VM guest operating system in 64-bit mode
Zone PFN ranges:
  DMA       0x00000000 -> 0x00080000
  Normal    0x00080000 -> 0x00080000
Movable zone start PFN for each node
early_node_map[] active PFN ranges
  0: 0x00000000 -> 0x00020000
Built 1 zonelists in Zone order, mobility grouping on. Total pages: 129024
Kernel command line: ramdisk_size=65536 root=/dev/raml ro init=/linuxrc TERM=dumb
  HostIP=9.60.18.223 Hostname=gpok223.endicott.ibm.com
  Gateway=9.60.18.129 Netmask=255.255.252.0
  Broadcast=9.60.18.255 Layer2=0
  ReadChannel=0.0.0600 WriteChannel=0.0.0601 DataChannel=0.
0.0602
  Nameserver=9.0.2.11 Portname=whatever Portno=0
  Install=nfs://9.60.18.224/nfs/sles11/SLES-11-DVD-s390x-GM-D
VD1.iso
  UseVNC=1 VNCPassword=12345678
  InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0
... 
```

The install system or `starter system` should continue to boot. You should see the message:

```
starting VNC server...
A log file will be written to: /var/log/YaST2/vncserver.log ...
```

```
You can connect to 9.60.18.223, display :1 now with vncviewer
Or use a Java capable browser on http://9.60.18.223:5801/

(When YaST2 is finished, close your VNC viewer and return to this window.)

*** Starting YaST2 ***

2. You can open a Java-enabled browser to access YaST2 at the specified URL. The logon prompt in Figure 8-1 shows VNC access through a Java-enabled browser.

In addition to a browser, you can also use a standalone VNC viewer if desired. Connect to the VNC server (9.60.18.223:1 in this example) as shown in Figure 8-2. Enter the password specified in the parameter file (12345678 in this example).

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).

Now the installation process should begin.

### 8.4.2 Beginning YaST installation

The installation program that is running is `yast2`. Perform the following steps:

1. Choose your language and keyboard, (English US in this example). Read the License Agreement, choose I agree to the License Agreement and Click Next.
2. The *Disk Activation* window should appear. Choose **Configure DASD Disks**.

3. The *DASD Disk Management* window should appear; you will see all the DASD available to S11GOLD.
   a. Highlight each of the minidisks and VDISKs, 100, 101, 300 and 301 and click **Select or Deselect**.
   b. You should see a *Yes* appear next to them in the *Sel.* column on the left. Activate them by clicking **Perform Action -> Activate**, as shown in the left side of Figure 8-3 on page 123.
   c. Disks 100 and 101 must be formatted so that Linux can use them. Deselect disks 300 and 301 using the **Select or Deselect** button, so that 100 and 101 remain selected. Now click **Perform Action -> Format** as shown on the right side of the figure.

![DASD Disk Management](image)

*Figure 8-3  DASD that are available to S11GOLD*

4. You should see a window asking for 2 Parallel Formatted Disks. Accept the default of writing a volume label and click **OK**.

5. Click **Yes** to the question *Really format the following disks?*

6. A progress indicator window should appear displaying the progress of the parallel formats. This step can take 1-10 minutes depending on a number of factors.

7. When the formatting is complete, click **Next** in the *DASD Disk Management* window.

8. In the *Disk Activation* window click **Next** again.

9. In the *Installation Mode* window accept the default of **New installation** and Click **Next**.

10. The *Clock and Time Zone* window will appear. Choose your region and time zone and click **Next**.

11. This will bring you to the *Installation Settings* window. Click **Partitioning** in the **Overview** tab. The **Expert Partitioner** window will appear as shown in Figure 8-4 on page 124.
12. In this example, two partitions will be made on /dev/dasda and one partition will be made on /dev/dasdb.
   a. Click the plus sign (+) to the left of Hard Disk. It should show dasda–dasdd in the tree.
   b. Click on dasda on the left and then on the Add button. You should see a new window named Add Partition on /dev/dasda.

13. Accept the default of the Custom size radio button and choose a size of 384 MB as shown in Figure 8-5 on page 125. Click Next.
14. You should see a new window *Add Partition on /dev/dasda*. If not set by default, perform the following steps:
   a. Click the **Format** radio button.
   b. Choose **Ext3** as the file system.
   c. Accept the default **Mount Point of /**.

15. Click **Finish**. This will create the partition `/dev/dasda1`.

16. To create a second partition click **Add** while dasda is selected in the left tree. You should see a new window *Add partition on /dev/dasda*. Accept the default size of all the remaining space or click the **Maximum Size** radio button. Click **Next**.

17. In new *Add partition on /dev/dasda* window as shown in Figure 8-6 on page 126 perform the following steps:
   a. Click the **Do not format** radio button.
   b. Click **Do not mount partition** radio button.
   c. Click **Finish**. This should create an empty partition `/dev/dasda2`. 
18. Create an empty partition /dev/dasdb1 in the same fashion as you did to create /dev/dasda2 (/dev/dasdb => Add => Do not format, empty mount point). Accept the default size of all the space.

19. Now /dev/dasda2 and /dev/dasdb1 can be used to create a volume group. Click Volume Management as shown on Figure 8-7 on page 127.
20. Click on **Add Volume Group**.

21. At the *Create a Volume Group* window set the volume group name to **system-vg**. Select both `/dev/dasda2` and `/dev/dasdb1` in *Available Physical Volumes* and **Add** them to *Selected Physical Volumes* as shown on Figure 8-8 on page 128. Click **Finish**.
22. Back in *Expert Partitioner*, click on freshly created *system-vg* under *Volume Management* as shown on Figure 8-9 on page 129. It shows there are no logical volumes defined in system-vg. Click on **Add**.
23. Enter `usr-lv` as the name for new logical volume and click **Next**.

24. On the **Add Logical Volume** window, click the **Manual Size** radio button and enter a size of 2 GB. Click **Next**.

25. Format partition as **ext3** and choose a mount point `/usr` as shown in Figure 8-10 on page 130. Click **Finish**.
26. Create three more logical volumes, \texttt{var-lv}, \texttt{opt-lv}, and \texttt{tmp-lv} similarly using the following table:

\begin{table}[h]
\centering
\begin{tabular}{|l|l|l|}
\hline
Mount point & Logical volume name & Size \\
\hline
\texttt{/usr/} & \texttt{usr-lv} & 2 GB \\
\texttt{/var/} & \texttt{var-lv} & 512 MB \\
\texttt{/opt/} & \texttt{opt-lv} & 384 MB \\
\texttt{/tmp/} & \texttt{tmp-lv} & 384 MB \\
\hline
\end{tabular}
\caption{Logical volumes added to system-vg volume group}
\end{table}

You may choose other sizes. See 2.3, “Disk planning” on page 10 for more discussion.

27. Back in the \textit{Expert Partitioner} window, click on the top object in the \textit{System View} (\texttt{gpok223} in this example). You should see output similar to Figure 8-11. You don’t have to format /dev/dasdc1 (vdev 300) and /dev/dasdd1 (vdev 301) because they were properly formatted as a Linux swap space by the \texttt{SWAPGEN EXEC}, and thus should be recognized as a swap space. Click \texttt{Accept}.
28. In the **Installation Settings** window, select **Software**. You may see an error window.

29. The **Software Selection** window opens. Leave **Server Base System, 32Bit Runtime Environment, Help and Support Documentation** and **Minimal System** selected. Uncheck all other setting as showed in Figure 8-12 on page 132, click **OK**.
30. In **Expert** tab click on Default Runlevel and choose 3: Full multiuser with network as shown in Figure 8-13 on page 132. Click **OK**. You will receive VNC warning, click **Yes**.

31. In the **Languages** window, select your primary and optionally secondary languages and click **Accept**.

32. You are now ready to begin copying the RPMs onto your root file system. In the **Installation Settings** window, check the settings and click **Install**.

33. In the **Confirm Installation** window click **Install**.

SLES11 system will be installed onto DASD. This should take about 5-20 minutes depending on a number of factors.
8.4.3 Rebooting your new Linux system from disk

After the first part of installation completes, the Linux system will reboot automatically. This time it boots from disk, not from the reader. You should see messages scrolling in the z/VM 3270 session. You will have to clear the screen a number of times. The install program will bring up a VNC server again to complete the installation.

```
... starting VNC server...
A log file will be written to: /var/log/YaST2/vncserver.log ...
***
*** You can connect to 9.60.18.223, display :1 now with vncviewer
*** Or use a Java capable browser on   http://9.60.18.223:5801/
***
(When YaST2 is finished, close your VNC viewer and return to this window.)
...
```

8.4.4 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 (12345678 in this example).
2. 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!
3. In the Hostname and Domain Name window, both Hostname and Domain Name are entered by installer as they were specified in S11GOLD PARM-S11 file. In this example gpok223 is the host name and endicott.ibm.com is the domain name. Uncheck the Change Hostname via DHCP check-box. Click Next.
4. In the Network Configuration window you will see Firewall is enabled, click on the word disable to disable it as shown in Figure 8-14 on page 134. All other values should be correct so just click Next.
5. In the *Test Internet Connection* window, if you do not have Internet access, select **No, skip this test** and click **Next**.

6. In the *Installation Overview* accept the default and click **Next**. A default CA and certificate will be created.

7. In the *User Authentication Method* window accept the default of **Local (/etc/passwd)** and click **Next**.

8. The next window should be *Add a new local user*. It is recommended that you add at least one user so as to have a non-root ID on all cloned systems. In this example, the user mikemac was added. When you are done, click **Next**.

9. In the *Writing the system configuration* window the **SuSEconfig** tool writes all your settings to disk.

10. The next window will be *Release Notes*. After reviewing the release notes, click **Next**.

11. In the *Hardware Configuration* window choose the **Skip Configuration** radio button and click **Next**.

12. The last installation window is *Installation Completed*. **Uncheck** the check box **Clone This System for Autoyast** check box and click **Finish**.

The VNC session should end. Return to the 3270 session and you may have to clear the screen a few times. Then you should see a login prompt. You are done installing Linux! You can disconnect from the 3270 session using the **DISCONNECT** command:

```
=> #cp disc
```
From this point forward, it is recommended that you access your Linux systems using 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 19.

8.5 Configuring the golden image

Now you want to customize the golden 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.

- “Configuring the VNC server” on page 135
- “Preparing for Online Update” on page 138
- “Removing unnecessary RPMs” on page 138
- “Turning off unneeded services” on page 138
- “Enabling the cmrn module” on page 139
- “Applying service - online update” on page 140
- “Configuring /etc/inittab” on page 140
- “On Demand Timer patch” on page 140
- “Configuring SSH keys” on page 141
- “Modifying zipl.conf” on page 142
- “Cleaning temporary files” on page 143
- “Rebooting the system and verifying changes” on page 143

8.5.1 Configuring the VNC server

Start an SSH session as root to the golden image.

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 using the `vncserver` command.

SLES11 configures a VNC connection which starts by default. It is configured through the `xinetd` daemon. It is recommended that you disable it. We will configure VNC later on and it will run only when needed.

Edit the file `/etc/xinetd.d/vnc`. In sections `service vnc1` and `service vnchttpd1` add the parameter `disable=yes`.

```
# cd /etc/xinetd.d
# vi vnc
service vnc1
{
    socket_type = stream
    protocol = tcp
    wait = no
    user = nobody
    server = /usr/bin/Xvnc
    server_args = -noreset -inetd -once -from localhost -query localhost
                  -geometry 1024x768 -depth 16
    type = UNLISTED
    port = 5901
    disable = yes
}
```

... 

```
The Virtualization Cookbook for SLES 11

```plaintext
{
socket_type = stream
protocol = tcp
wait = no
user = nobody
server = /usr/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 (this can be verified with the `chkconfig --list` command). You can turn off the `xinetd` service with the `service xinetd stop` command for this session and the `chkconfig` command at boot time:

```plaintext
# service xinetd stop
Shutting down xinetd: done
# chkconfig xinetd off
```

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:

```plaintext
# 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
xauth: creating new authority file /root/.Xauthority

New 'X' desktop is gpok223:1

Creating default startup script /root/.vnc/xstartup
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/gpok223:1.log
```

Stop the VNC server using the `-kill :1` argument:

```plaintext
# vncserver -kill :1
Killing Xvnc process ID 6961
```

The `icewm` package allows for the IceWM, a window manager that is more usable than the Tiny Window Manager (`twm`) that VNC uses by default. It is therefore recommended that you change to `icewm`. The package should be already installed on your system. Verify the RPM was added:

```plaintext
# rpm -q icewm icewm-lite
icewm-1.2.36-1.35
icewm-lite-1.2.36-1.18
```

Change the window manger from `twm` to `icewm` in the file `/root/.vnc/xstartup`:

```plaintext
# cd /root/.vnc/
# vi xstartup
#!/bin/sh
xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
icewm &
```
You may want to remove the **passwd** file so the cloned system does not have the same password as you just entered. If so, use the command `rm passwd`. If you remove the password file, when a system is cloned, the password will be prompted for the first time that the VNC server is initialized.

**Verifying VNC server configuration**

Start the VNC server again:

```
# vncserver
```

New 'X' desktop is gpok223:1

Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/gpok223:1.log

Start a VNC client as shown on the left side of Figure 8-15. Enter the password and the resulting VNC session is shown on the right side of Figure 8-15.

Note that the VNC server will not be started across reboots. When you need a graphical environment, you will either have to start `vncserver` manually (recommended), or you will have to start it in the desired run level through a script such as `/etc/init.d/boot.local`.

Unless you need a graphical environment, you can again stop the VNC server using the `-kill :1` argument:

```
# vncserver -kill :1
Killing Xvnc process ID 6961
```
Do not forget to remove the password file, so that your password from the golden image is not distributed to every cloned image (unless it is your intent).

8.5.2 Preparing for Online Update

This step is recommended if you have Internet access.

**Note:** The system used for the writing of this book did not have Internet access, so the following steps have not been tested for SLES 11.

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.

A graphical environment is recommended for this step. Start a VNC viewer. In a terminal session, start YaST with command `yast2` and choose the **Software => Online Update Configuration** selection.

8.5.3 Removing unnecessary RPMs

For the golden image it is desirable to have as lean a Linux image as possible from which to build on. Therefore in previous releases of this book we provided here a list of RPMs which we considered unnecessary. However, this section is no longer recommended:

**Important:** This section is not recommended by a Novell employee for the following reason:

The problem that I (and others) ran into was that a number of these packages are part of the "Base pattern" and when you install the next service pack, those got re-installed automatically. Some of those have been removed from the Base pattern for SP2, but others have been added. And, with SP2, those packages that are in the Base pattern now have actual RPM dependencies that will prevent them from being removed after installation. Novell Technical Support considers systems that have Base pattern packages "missing" as problematic for support and updates. So, I can't recommend doing that at all.

So only consider removing RPMs if you really, really want to save space.

8.5.4 Turning off unneeded services

There are a number of services which are started in a SLES10 SP2 minimum system. They can be viewed using the following `chkconfig` command:

```
# chkconfig -l | grep 3:on
```

cron          0:off 1:off 2:on 3:on 4:off 5:on 6:off
dbus          0:off 1:off 2:on 3:on 4:off 5:on 6:off
earlysyslog   0:off 1:off 2:on 3:on 4:off 5:on 6:off
fbset         0:off 1:on 2:on 3:on 4:off 5:on 6:off
haldaemon     0:off 1:off 2:on 3:on 4:off 5:on 6:off
irq_balancer  0:off 1:on 2:on 3:on 4:off 5:on 6:off
network       0:off 1:off 2:on 3:on 4:off 5:on 6:off
network-remotefs 0:off 1:off 2:on 3:on 4:off 5:on 6:off
nfs           0:off 1:off 2:off 3:on 4:off 5:on 6:off
nscd          0:off 1:off 2:off 3:on 4:off 5:on 6:off
postfix       0:off 1:off 2:off 3:on 4:off 5:on 6:off
random        0:off 1:off 2:on 3:on 4:off 5:on 6:off
In order to keep the golden image as lean as possible in terms of CPU usage, some of these services can be turned off. Turn off the following services using the `chkconfig` command:

```
# chkconfig fbset off
# chkconfig network-remotefs off
# chkconfig postfix off
# chkconfig splash off
# chkconfig splash_early off
# chkconfig smartd 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
```

This shows the services that will run in the default run level of 3.

### 8.5.5 Enabling the cmm module

The `cmm` module allows Linux to do cooperative memory management (aka CMM1) with z/VM. This feature must be enabled on both the Linux and z/VM sides for it to function (hence the name `cooperative`).

To enable it, edit the file `/etc/sysconfig/kernel` and add `cmm` to the variable `MODULES_LOADED_ON_BOOT` (around line 30):

```
# cd /etc/sysconfig
# vi kernel
...
## 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="cmm"
...
```

Save the file and the `cmm` module should be loaded after your system is rebooted.
8.5.6 Applying service - online update

If you have a SuSE Maintenance Web account then you can use it to retrieve the latest patches for SLES 11. Because many of these patches contain security and bug fixes, it is recommended that you apply the patches for the golden image so that it is up to date. Subsequently all the servers you clone after the golden image will also be up to date.

Section 8.5.2, “Preparing for Online Update” on page 138 must be completed before proceeding.

The system used for the writing of this book did not have Internet access, so this part was not tested at all. You may choose which packages to install and which not. It is highly recommended to download security patches.

As described in “Applying service if necessary - online update” on page 111 it is strongly recommended to update default kernel of SLES 11.

When you finish here, don’t reboot yet, you will reboot shortly after some boot configuration changes are made.

8.5.7 Configuring /etc/inittab

If you did not change the default runlevel from 5 to 3 during the install process, this would be a good time to do so. By default, SLES 11 boots into run level 5, which is designed for a graphical environment. To keep the golden image as lean as possible, it is recommended that this be reset to run level 3. This is set in the /etc/inittab file with the variable initdefault.

Linux 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, it might be better that your system halts (shuts down).

Change these two settings in the /etc/inittab file as follows:

- Modify id:5:initdefault: to id:3:initdefault: if needed
- Modify shutdown -r to shutdown -h

These changes will be picked up when the system is rebooted.

8.5.8 On Demand Timer patch

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 timer unless the server has a heavy, constant workload. Otherwise, even an idle Linux image never goes idle from a z/VM point of view.
In the past, there was the On Demand Timer patch needed on System z which solved this problem (disabled the timer pop up). It was System z specific code. Later on, people on x86 also discovered that the 10ms timer can also have drawbacks, mainly because it reduced battery lifetime. So generic timer code was developed. This code is enabled by default and makes sure that idle Linux guest stays idle also from z/VM point of view.

With the new code, system control was also dropped. With system control it was possible to enable/disable the timer at runtime. The current code just works and it is enabled by default. The only interface to it is a kernel parameter `nohz=` that allows to switch off the NOHZ code at the boot time.

### 8.5.9 Configuring SSH keys

SSH sessions are typically authenticated using passwords typed in from the keyboard. With SSH **key-based authentication** sessions can be authenticated using 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 cloner (client) to the virtual servers. If the golden image has a copy of cloner's public key in the file `/root/.ssh/authorized_keys`, and the cloner 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.

To set up key-based authentication from the cloner to the golden image (and consequently from cloner to all cloned images), perform steps that follow. You will need SSH sessions on both the golden image (`gpok223`) and the cloner (`gpok224`), so prefixes are used in the following examples.

- **On the golden image**, create the directory `/root/.ssh` and set the permission bits to octal 700 with the `chmod` command:

  ```bash
  gpok223: # mkdir /root/.ssh
  gpok223: # chmod 700 /root/.ssh
  ```

- **Start an SSH session as root** on the cloner, the IP address is `9.60.18.224` in this example.

- **Copy the cloner’s public key from the `/etc/ssh/` directory to the `/root/.ssh/` directory that you just created on the golden image (`9.60.18.223` in this example):

  ```bash
  gpok224: # cd /etc/ssh
  gpok224: # scp ssh_host_dsa_key.pub 9.60.18.223:/root/.ssh/authorized_keys
  ```

  The authenticity of host '9.60.18.223 (9.60.18.223)' can't be established.
  Are you sure you want to continue connecting (yes/no)? yes
  Warning: Permanently added '9.60.18.223' (RSA) to the list of known hosts.
  
  Password: 

  ssh_host_dsa_key.pub 100% 602 0.6KB/s 00:00

- **On the cloner, create a symbolic link from `/root/.ssh/id_dsa` to the cloner's private key:

  ```bash
  gpok224: # cd /root/.ssh
  gpok224: # ls -s /etc/ssh/ssh_host_dsa_key id_dsa
  ```

- **Verify the symbolic link was created using the `ls -l` command:

  ```bash
  gpok224 # ls -l
  lrwxrwxrwx 1 root root 25 Nov 30 08:30 id_dsa -> /etc/ssh/ssh_host_dsa_key
  ```

  ```bash
  gpok224 # ls -l
  lrwxrwxrwx 1 root root 25 Nov 30 08:30 id_dsa -> /etc/ssh/ssh_host_dsa_key
  ```
Verify that you can `ssh` into the golden image without needing a password.

```
gpok224: # ssh 9.60.18.223
```

If you get an SSH session without having to supply a password, as with the above example, it shows that key-based authentication is working.

### 8.5.10 Modifying zipl.conf

In previous versions of this book, this section described how to add disks which should be brought online during IPL. The `dasd=` parameter in the `/etc/zipl.conf` was used for that. In SLES 11 this parameter is no longer supported. Instead `dasd_configure` script is used for adding disks to the system.

Two changes are recommended to the default `zipl.conf` file:

1. The parameters `vmpoff=LOGOFF` and `vmhalt=LOGOFF` have the effect of logging the z/VM user ID off when Linux is shut down. This can be convenient for shutting the z/VM system down more efficiently and also for getting a refreshed 3270 emulator session.

2. The `timeout` value is modified to 3 seconds. When SLES 11 boots, the default time that is allowed to enter a menu command is 10 seconds. Because the default value is most commonly used, this 10 seconds delays the starting of Linux. It is backed up to three seconds (you might want to choose five if you’re not a fast typist :))

Make a backup copy of the file and add the string `vmpoff=LOGOFF` to the parameter line in the `[Linux]` section and set `timeout=3` in the `menu` section:

```
# cd /etc
# cp zipl.conf zipl.conf.orig
# vi zipl.conf // add the dasd= string
# Modified by YaST2. Last modification on Sun Nov 29 00:56:23 UTC 2009
[defaultboot]
defaultmenu = menu

[Linux]
image = /boot/image-2.6.27.19-5-default
target = /boot/zipl
ramdisk = /boot/initrd-2.6.27.19-5-default,0x2000000
parameters = "root=/dev/disk/by-path/ccw-0.0.0100-part1 vmpoff=LOGOFF vmhalt=LOGOFF TERM=dumb init=/linuxrc"

:menu
  default = 1
  prompt = 1
  target = /boot/zipl
  timeout = 3
  1 = Linux
  2 = ipl
  3 = Failsafe

### Don't change this comment - YaST2 identifier: Original name: linux###
```

```
[[ipl]]
image = /boot/image
target = /boot/zipl
ramdisk = /boot/initrd,0x2000000
```

---

The Virtualization Cookbook for SLES 11
parameters = "root=/dev/disk/by-path/ccw-0.0.0100-part1 TERM=dumb"

### Don't change this comment - YaST2 identifier: Original name: failsafe ###
[Failsafe]
  image = /boot/image-2.6.27.19-5-default
  target = /boot/zipl
  ramdisk = /boot/initrd-2.6.27.19-5-default,0x2000000
  parameters = "root=/dev/disk/by-path/ccw-0.0.0100-part1 TERM=dumb x11failsafe"

Now, run the `zipl` command so the changes are written to the boot record:

```bash
# zipl
Using config file '/etc/zipl.conf'
Building bootmap in '/boot/zipl'
Building menu 'menu'
Adding #1: IPL section 'Linux' (default)
Adding #2: IPL section 'ipl'
Adding #3: IPL section 'Failsafe'
Preparing boot device: dasda (0100).
Done.
```

These changes will be utilized the next time Linux is rebooted.

### 8.5.11 Cleaning temporary files ###

Now is the best time to clean temporary files from golden image. If they stay there they will be copied to each clone and occupy space there. There are a number of directories which can be cleaned up and the history of the commands executed can be cleared.

- Create a script `cloneprep.sh` that can be run to prepare for cloning:

  ```bash
  # cd /usr/local/sbin
  # vi cloneprep.sh
  #!/bin/bash
  rm -fr /tmp/.ICE-unix
  rm -fr /tmp/.X11-unix
  rm -rf /tmp/*
  rm -f /var/log/YaST2/*
  history -c
  
  You may choose different steps to prepare for cloning.
  ```

- Make the script executable with the `chmod +x` command:

  ```bash
  # chmod +x cloneprep.sh
  ```

- Call the script interactively:

  ```bash
  # cloneprep.sh
  ```

The system should now be cleaned for cloning.

### 8.5.12 Rebooting the system and verifying changes ###

You are now done customizing the golden Linux image. Now `reboot` to test your changes:

```bash
# reboot
Broadcast message from root (pts/0) (Mon Nov 30 08:51:49 2009):

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 golden image** and check a few settings.

2. Use the `df` command to display your file systems:

   ```
   # df -h
   /dev/dasda1           372M  100M  254M  29% /
   udev                  248M  120K  248M  1% /dev
   /dev/mapper/system--vg-opt--lv  372M  17M  337M  5% /opt
   /dev/mapper/system--vg-tmp--lv  372M  17M  337M  5% /tmp
   /dev/mapper/system--vg-usr--lv  2.0G  979M  936M  52% /usr
   /dev/mapper/system--vg-var--lv  504M  50M  429M 11% /var
   ```

3. Confirm that both of your swap spaces are operational:

   ```
   # swapon -s
   Filename                                Type            Size    Used    Priority
   /dev/dasdc1                             partition       259956  0       -1
   /dev/dasdcl                             partition       519924  0       -2
   ```

   Device with higher priority will be used first.

4. Shutdown your golden image from the SSH session (to clone Linux it should be shut down).

   ```
   # shutdown -h now
   ```

   Congratulations! You have now successfully installed the golden image. This image will normally be shut down.
Chapter 9. Cloning SLES 11

It has become appallingly obvious that our technology has exceeded our humanity.
--Albert Einstein

This chapter focuses on cloning the golden image. Examples are given for cloning manually and through the use of a script. The following steps are described:

- “Formatting DASD for minidisks” on page 145
- “Defining a new user ID for a virtual server” on page 146
- “Cloning a virtual server manually” on page 149
- “Cloning a virtual server automatically” on page 154
- “Creating three more virtual servers” on page 156

9.1 Formatting DASD for minidisks

In section 4.6.2, “Formatting DASD for minidisks” on page 49, DASD was formatted to become minidisks for the cloner and the golden image. The CPFMTXA command can be used to format one DASD at a time, but the CPFORMAT EXEC is a wrapper around CPFMTXA that allows the formatting of multiple DASD.

To have access to enough DASD to define four more user IDs with two 3390-3 volumes each, three more 3390-9 volumes must be formatted. In the examples used in this book, the three volumes that will be used are at addresses 63CF-63D1. Consult your worksheets on 2.7.2, “z/VM DASD worksheet” on page 17.

1. Logon to a 3270 session as MAINT.
2. Query the devices that will be used for LINUX01-LINUX04. In this example they are 63CF-63D1.

```shell
=> q 63cf-63d1
DASD 63CF DM63CF , DASD 63D0 DM63D0 , DASD 63D1 DM63D1
```
3. Attach the seven volumes that will be used for the cloner, the common CMS disk and the golden image. In this example it is the DASD at addresses:

```shell
=> att 63cf-63d1 *
63CF-63D1 ATTACHED TO MAINT
```
4. Invoke the CPFORMAT command against these volumes using the parameter as perm:

```bash
===> cpformat 63cf-63d1 as perm
```

Format the following DASD:

<table>
<thead>
<tr>
<th>TargetID</th>
<th>Tdev</th>
<th>OwnerID</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>63CF</td>
<td>MAINT</td>
<td>63CF</td>
<td>3390</td>
<td>DM63CF</td>
<td>63CF</td>
<td>0</td>
</tr>
<tr>
<td>MAINT</td>
<td>63D0</td>
<td>MAINT</td>
<td>63D0</td>
<td>3390</td>
<td>DM63D0</td>
<td>63D0</td>
<td>0</td>
</tr>
<tr>
<td>MAINT</td>
<td>63D1</td>
<td>MAINT</td>
<td>63D1</td>
<td>3390</td>
<td>DM63D1</td>
<td>63D1</td>
<td>0</td>
</tr>
</tbody>
</table>

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

DASD successfully formatted: TM63CF TM63D0 TM63D1
63CF-63D1 DETACHED
63CF-63D1 ATTACHED TO MAINT

DASD status after:

<table>
<thead>
<tr>
<th>TargetID</th>
<th>Tdev</th>
<th>OwnerID</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>63CF</td>
<td>MAINT</td>
<td>63CF</td>
<td>3390</td>
<td>TM63CF</td>
<td>63CF</td>
<td>0</td>
</tr>
<tr>
<td>MAINT</td>
<td>63D0</td>
<td>MAINT</td>
<td>63D0</td>
<td>3390</td>
<td>TM63D0</td>
<td>63D0</td>
<td>0</td>
</tr>
<tr>
<td>MAINT</td>
<td>63D1</td>
<td>MAINT</td>
<td>63D1</td>
<td>3390</td>
<td>TM63D1</td>
<td>63D1</td>
<td>0</td>
</tr>
</tbody>
</table>

Detach the three volumes from MAINT using the DETACH command:

```bash
===> det 63cf-63d1
63CF-63D1 DETACHED
```

Attach the newly formatted DASD to SYSTEM so they can be used for minidisks:

```bash
===> att 63cf-63d1 system
DASD 63CF ATTACHED TO SYSTEM TM63CF
DASD 63D0 ATTACHED TO SYSTEM TM63D0
DASD 63D1 ATTACHED TO SYSTEM TM63D1
```

The three DASD volumes will now be available to be used for minidisks in the USER DIRECT file. They will also be available after the next IPL because their new labels match the pattern specified by the User_Volume_Include TM* statement in the SYSTEM CONFIG file.

### 9.2 Defining a new user ID for a virtual server

In this section you will define a new user ID, LINUX01.

1. Logon to MAINT and edit the USER DIRECT file to add a new user ID, LINUX01. Define it with the same sized 100 and 101 minidisks so that the golden image can be cloned to it.

```bash
===> x user direct c
```

2. Go to the bottom of the file and add the following six lines. In this example the user ID will be LINUX01 with a password of LNX4VM. Two 3390-3 DASD are used (about 4.5 GB total). In this example, the devices are at real address 63CF:

```bash
* 
USER LINUX01 LNX4VM 256M IG G  
INCLUDE LNXDFLT 
OPTION APPLMON 
MDISK 100 3390 0001 3338 TM63CF MR LNX4VM LNX4VM LNX4VM 
MDISK 101 3390 3339 3338 TM63CF MR LNX4VM LNX4VM LNX4VM
```
3. Add the new volume(s) 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:

```plaintext
====> top
====> /alloc
USER $ALLOC$ NOLOG

...   
MDISK A09 3390 000 001 TM63CF 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.

```plaintext
=> 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:

```plaintext
=> directxa user
```

The new Linux user ID has now been defined.

### 9.2.1 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:

```plaintext
=> link autolog1 191 1191 mr
=> acc 1191 f
=> x profile exec f  // add two lines
/****************************/
/*  Autolog1 Profile Exec */
/****************************/
'cp xautolog tcipip'       /* start up TCPIP */
'CP XAUTOLOG VMSEVS'
'CP XAUTOLOG VMSEVU'
'CP XAUTOLOG VMSEVR'
'CP XAUTOLOG DTCVSW1'
'CP XAUTOLOG DTCVSW2'
'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 300' /* Allow guests 5 min to shut down */

/* Grant access to VSWITCH for each Linux user ID */
'cp set vswitch vsw1 grant s11clone'
'cp set vswitch vsw1 grant s11gold'
'cp set vswitch vsw1 grant linux01'
```
/* XAUTOLOG each Linux user that should be started */
'cp xautolog s11clone'
'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
Command complete

The user ID LINUX01 should now have permission to attach its NIC to the VSWITCH.

9.2.2 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 SLES11 manually.

1. Logon to LNXMAINT.
2. Copy an existing parameter file to a new one with the file name equal to the new Linux
   user ID.
   
   => copy s11gold parm-s11 d linux01 =
   => x linux01 parm-s11 d

3. Edit the new parameter file as you did for S11GOLD (see “Creating the S11GOLD
   parameter file” on page 119). If the new Linux is going to be on the same network as the
   cloner 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 9.60.18.225 and the DNS name to
gpok225.endicott.ibm.com:

   ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb
   HostIP=9.60.18.225 Hostname=gpok225.endicott.ibm.com
   Gateway=9.60.18.129 Netmask=255.255.252.0
   Broadcast=9.60.18.255 Layer2=0
   ReadChannel=0.0.0600 WriteChannel=0.0.0601 DataChannel=0.0.0602
   Nameserver=9.0.2.11 Portname=whatever Portno=0
   Install=nfs://9.60.18.224/nfs/sles11/SLES-11-DVD-s390x-GM-DVD1.iso
   UseVNC=1 VNCPassword=12345678
   InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0

4. Logoff of LNXMAINT and **logon to LINUX01**.
5. Watch for error messages as LINUX01 is logged on. Answer **N** to the question “Do you
   want to IPL Linux from minidisk 100”:

   LOGON LINUX01
   NIC 0600 is created; devices 0600-0602 defined
   z/VM Version 6 Release 1.0, Service Level 0901 (64-bit),
   built on IBM Virtualization Technology
   There is no logmsg data
   FILES: NO RDR, NO PRT, NO PUN
   LOGON AT 08:39:10 EST TUESDAY 12/01/09
   z/VM V6.1.0 2009-11-19 13:47

   DMSACP723I A (191) R/O
   DMSACP723I C (592) R/O
   DIAG swap disk defined at virtual address 300 (64989 4K pages of swap space)
   DIAG swap disk defined at virtual address 301 (129981 4K pages of swap space)
Do you want to IPL Linux from minidisk 100? y/n
n
...

6. Verify that the new Linux user ID has a NIC at addresses 600-602:

   => q osa
   OSA 0600 ON NIC 0600 UNIT 000 SUBCHANNEL = 0002
       0600 DEVTYPE OSA CHPID 00 OSD
       0600 MAC 02-00-07-00-00-03 CURRENT
       0600 QDIO-ELIGIBLE QIOASSIST-ELIGIBLE
   OSA 0601 ON NIC 0600 UNIT 001 SUBCHANNEL = 0003
       0601 DEVTYPE OSA CHPID 00 OSD
       0601 QDIO-ELIGIBLE QIOASSIST-ELIGIBLE
   OSA 0602 ON NIC 0600 UNIT 002 SUBCHANNEL = 0004
       0602 DEVTYPE OSA CHPID 00 OSD
       0602 QDIO-ELIGIBLE QIOASSIST-ELIGIBLE

7. Verify that the minidisks at 100 and 101 the VDISKs at 300 and 301 are read-write:

   => q da
   DASD 0100 3390 TM63CF R/W   3338 CYL ON DASD 63CF SUBCHANNEL = 0000
   DASD 0101 3390 TM63CF R/W   3338 CYL ON DASD 63CF SUBCHANNEL = 0001
   DASD 0190 3390 610RES R/O   107 CYL ON DASD 6280 SUBCHANNEL = 0009
   DASD 0191 3390 TM6289 R/O   300 CYL ON DASD 6289 SUBCHANNEL = 000C
   DASD 0190 3390 TV6283 R/O   146 CYL ON DASD 6283 SUBCHANNEL = 000A
   DASD 019E 3390 TV6283 R/O   250 CYL ON DASD 6283 SUBCHANNEL = 000B
   DASD 0300 9336 (VDSK) R/W   524288 BLK ON DASD VDSK SUBCHANNEL = 000E
   DASD 0301 9336 (VDSK) R/W   1048576 BLK ON DASD VDSK SUBCHANNEL = 000F
   DASD 0592 3390 TV6284 R/O   70 CYL ON DASD 6284 SUBCHANNEL = 000D

8. Logoff of LINUX01. To clone to a user ID, it should be logged off so the target disks can be linked read-write.

You should now be ready to clone to this new user ID.

9.3 Cloning a virtual server manually

Before using the shell script **clone.sh** to clone a server, you may want to 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 (golden) Linux image is on the user ID S11GOLD on minidisks 100 and 101.
- The target user ID, LINUX01 in this example, has identically sized minidisks.
- The **vmcp** command is available to the cloner, S11CLONE, 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 source disks read-only
2. Link the target disks read-write
3. Copy the source to the target disk using **FLASHCOPY** or the Linux **dasdfmt** and **dd** commands.
4. Mount the newly copied root file system.
5. Modify the networking information on the target system.
6. Detach the target disks.
7. IPL the target system.
8. Modify the SSH keys on the target system.

The following sections describe these steps in detail.

**Linking the source disks read-only**

**Start an SSH session as root** to the cloner, S11CLONE.

The source minidisks at virtual addresses 100 and 101 are on the golden image, S11GOLD. They are linked read-only as virtual addresses 1100 and 1101 using the RR parameter to the CP LINK command:

```
# vmcp link s11gold 100 1100 rr
# vmcp link s11gold 101 1101 rr
```

**Linking the target disk read-write**

The target minidisks, also at addresses 100 and 101 are on the cloner, LINUX01. They are linked multi-read (read-write if no other user ID has write access) using the MR parameter as virtual device 2100 and 2101 using the CP LINK command:

```
# vmcp link linux01 100 2100 mr
# vmcp link linux01 101 2101 mr
```

**Copying the source to the target disk using FLASHCOPY**

The two disks are copied using the CP FLASHCOPY command:

```
# vmcp flashcopy 1100 0 end to 2100 0 end
Command complete: FLASHCOPY 1100 0 3337 TO 2100 0 3337
# vmcp flashcopy 1101 0 end to 2101 0 end
Command complete: FLASHCOPY 1101 0 3337 TO 2101 0 3337
```
Mounting the newly copied root file system

First enable target 100 (local 2100) disk for Linux usage with `chccwdev -e` command.

```bash
# chccwdev -e 2100
Setting device 0.0.2100 online
Done
```

Note: If you do not have FLASHCOPY support, you can use the Linux `dasdfmt` and `dd` commands. You must first enable the source and target disks using the `chccwdev -e` command, then determine the device name using the `lsdasd` command:

```bash
# chccwdev -e 1100
Setting device 0.0.1100 online
Done
# chccwdev -e 1101
Setting device 0.0.1101 online
Done
# chccwdev -e 2100
Setting device 0.0.2100 online
Done
# chccwdev -e 2101
Setting device 0.0.2101 online
Done
```

In this example the source devices are `/dev/dasdh` and `/dev/dasdi`, and the target devices are `/dev/dasdj` and `/dev/daskd`. Format the target disks using the `dasdfmt` command then copy it using the `dd` command using a block size of 4K (4096) bytes:

```bash
# dasdfmt -b 4096 -y -f /dev/dasdj
... # dasdfmt -b 4096 -y -f /dev/daskd
... # dd if=/dev/dasdh of=/dev/dasdj bs=4096
  600840+0 records in
  600840+0 records out
# dd if=/dev/dasdi of=/dev/daskd bs=4096
  600840+0 records in
  600840+0 records out
```

The golden image should now be copied to the target disks. Disable and detach the disks:

```bash
# chccwdev -d 1100
# vmpc det 1100
# chccwdev -d 1101
# vmpc det 1101
# chccwdev -d 2100
# vmpc det 2100
# chccwdev -d 2101
# vmpc det 2101
```
Determine which device name is the target 100, or local 2100 disk:

```
# lsdasd | grep 2100
0.0.2100 active dasdh 94;28 ECKD 4096 2347MB 600840
```

The first target disk in this example is the device node is /dev/dasdh. The first partition is the root file system, it will be /dev/dasdh1. Mount it over a the mount point /mnt and cd into it:

```
# mount /dev/dasdh1 /mnt
# cd /mnt
```

Observe that this appears to be a root file system:

```
# ls
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 9.60.18.223 to 9.60.18.225) and the host name (from gpok223 to gpok225).

The host name is changed in the file /mnt/etc/HOSTNAME:

```
# cd etc
# vi HOSTNAME
gpok225.endicott.ibm.com
```

The IP address is changed in the file /mnt/etc/sysconfig/network/ifcfg-eth0:

```
# cd sysconfig/network
# vi ifcfg-eth0
BOOTPROTO='static'
IPADDR='9.60.18.225/22'
BROADCAST='9.60.19.255'
STARTMODE='onboot'
NAME='OSA Express Network card (0.0.0600)'
```

Detach the target disks

Change directory to /, use the sync command to flush the disks and the umount command to unmount the modified root file system:

```
# cd /
# sync
# umount /mnt
```

Remove the target disk from cloner:

```
# chccwdev -d 2100
Setting device 0.0.2100 offline
Done
```

Detach the source and target minidisks using the CP DETACH command:

```
# vmcp det 1100
DASD 1100 DETACHED
# vmcp det 1101
DASD 1101 DETACHED
# vmcp det 2100
DASD 2100 DETACHED
# vmcp det 2101
DASD 2101 DETACHED
```
The newly copied and modified system disks have now been detached.

**IPL the target system**

You should now be ready to IPL the manually cloned system.

**Logon to LINUX01 and IPL from minidisk 100:**

```
LOGON LINUX01
NIC 0600 is created; devices 0600-0602 defined
z/VM Version 6 Release 1.0, Service Level 0901 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:  NO RDR,  NO PRT,  NO PUN
LOGON AT 11:22:06 EST TUESDAY 12/01/09
z/VM V6.1.0  2009-11-19 13:47

DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 300 (64989 4K pages of swap space)
DIAG swap disk defined at virtual address 301 (129981 4K pages of swap space)
Do you want to IPL Linux from minidisk 100? y/n
y
zIPL v1.8.0 interactive boot menu

0. default (Linux)
1. Linux
2. ipl
3. Failsafe

Note: VM users please use 'cp vi vmsg <number> <kernel-parameters>'

Please choose (default will boot in 3 seconds):
Booting default (Linux)...
Initializing cgroup subsys cpuset
Initializing cgroup subsys cpu
Linux version 2.6.27.19-5-default (geeko@buildhost) (gcc version 4.3.2 [gcc-4_3-
branch revision 141291] (SUSE Linux) ) #1 SMP 2009-02-28 04:40:21 +0100
...
```

Watch for error messages on the console. 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 as root** to the new system using the new IP address or host name.

The SSH keys that were copied are identical to those of the golden 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 golden image except for the IP address and host name.

Next you can 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 using the LOGOFF command:

```
=> # cp log
```

### 9.4 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 as root to the cloner. The `clone.sh` script should be in your PATH in the directory `/usr/local/sbin/`. You can verify this with the `which` command:

```
# which clone.sh
/usr/local/clone.sh
```

The script takes two pairs of parameters:

- **from `<user ID>`** The source Linux
- **to `<user ID>`** the target Linux system

Both the source and target user IDs must be logged off. The script reads the parameter file on the LNXMAINT 192 disk (the cloner's 191 disk) to obtain information necessary to give the new Linux virtual server an identity. It calls CP `FLASHCOPY` using the `vmcp` module/command to try to copy the golden image’s 100 and 103 disks. If `FLASHCOPY` fails, the script falls back to copying the disk using the Linux `dasdfmt` and `dd` commands. It then regenerates SSH keys and adds the server's public key to the cloner's known_hosts file.

**Note:** The `clone.sh` script changes only the IP address, the host name and the domain. The Netmask is not changed.

The script then boots the new Linux using the `XAUTOLOG` command.

It should take less than a minute to clone with `FLASHCOPY` support and 10-20 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 from s11gold to linux01
Checking that S11GOLD exists and is not logged on ...
Invoking CP command: QUERY S11GOLD
HCPCQU045E S11GOLD not logged on
Error: non-zero CP response for command 'QUERY S11GOLD': #45
Checking that LINUX01 exists and is not logged on ...
Invoking CP command: QUERY LINUX01
HCPCQU045E LINUX01 not logged on
Error: non-zero CP response for command 'QUERY LINUX01': #45
Device already is offline
Done
Setting device 0.0.0191 online
Done
S11GOLD  PARM=S11 D1 V       65          9           1 11/28/2009 18:45:47
```
LINUX01  PARM-S11 D1 V         65          9          1 12/01/2009  8:38:49

WARNING!!!: Mindisks 100 and 101 will be copied to LINUX01
New host name will be:  gpok225.endicott.ibm.com
New IP address will be:  9.60.18.225
Other network data is retrieved from LINUX01 PARM-S11 on 191 disk
Are you sure you want to overwrite these disks (y/n): y

In the section above, it is verified that the user IDs S11GOLD and LINUX01 exist and are logged off. The 191 disk is deactivated and reactivated to pick up any changes that may have occurred. It is then verified that the two parameter files, S11GOLD PARM-S11 and LINUX01 PARM-S11 exist. You are then asked to confirm that the disks on the target system can be written to. Answer yes with y.

Linking source and target 100 disks ...
Invoking CP command: detach 1100
HPDVT040E Device 1100 does not exist
Error: non-zero CP response for command 'DETACH 1100': #40
Invoking CP command: link S11GOLD 100 1100 rr

Invoking CP command: detach 2100
HPDVT040E Device 2100 does not exist
Error: non-zero CP response for command 'DETACH 2100': #40
Invoking CP command: link LINUX01 100 2100 mr

Copying 100 disks ...
FLASHCOPYing 1100 to 2100 ...
Invoking CP command: FLASHCOPY 1100 0 end to 2100 0 end
Command complete: FLASHCOPY 1100 0 3337 TO 2100 0 3337
Invoking CP command: det 1100
DASD 1100 DETACHED
Linking source and target 101 disks ...
Invoking CP command: detach 1101
HPDVT040E Device 1101 does not exist
Error: non-zero CP response for command 'DETACH 1101': #40
Invoking CP command: link S11GOLD 101 1101 rr

Invoking CP command: detach 2101
HPDVT040E Device 2101 does not exist
Error: non-zero CP response for command 'DETACH 2101': #40
Invoking CP command: link LINUX01 101 2101 mr

Copying 101 disks ...
FLASHCOPYing 1101 to 2101 ...
Invoking CP command: FLASHCOPY 1101 0 end to 2101 0 end
Command complete: FLASHCOPY 1101 0 3337 TO 2101 0 3337
Invoking CP command: det 1101
DASD 1101 DETACHED
Invoking CP command: det 2101
DASD 2101 DETACHED

In the section above, the script copies the source 100 and 101 disks to the target user ID using the FLASHCOPY command. If FLASHCOPY is not supported or fails for some other reason the script falls back to the dasdfmt and dd commands to copy the disks.

Mounting newly cloned image over /mnt/targetLinux ...
Setting device 0.0.2100 online
Done
Mounting /dev/dasdi1 over /mnt/targetLinux ...
Modifying cloned image under /mnt/targetLinux ...
Regenerating SSH keys in /mnt/targetLinux/etc/ssh/ ...
Adding 9.60.18.225 to known_hosts file
Setting device 0.0.2100 offline
Done
Invoking CP command: DETACH 2100
DASD 2100 DETACHED

In the section above the script mounts the newly copied root file system over a mount point, /mnt/targetLinux/. The networking information is modified and the SSH keys are regenerated. Then the disk is detached.

Invoking CP command: XAUTOLOG LINUX01
Command accepted
Successfully cloned S11GOLD to LINUX01
You should be able to ping 9.60.18.225 within one minute

In the final section, the target user ID is logged on using XAUTOLOG. Because the PROFILE EXEC detects that the ID is logged on in a disconnected mode, Linux is IPLed from minidisk 100. The new system should be on the network in about 30-45 seconds.

Note: If the clone.sh script fails you can 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 the LNXMAINT 192 disk.
- The target 100 and 101 minidisks are sized identically.

9.5 Creating three more virtual servers

So far you have installed Linux manually twice onto S11CLONE and S11GOLD. You have created a new user ID LINUX01 and cloned golden image to it. Now it is time to clone three more times to have one system for each of the virtual servers described in the remaining chapters.

The following steps are involved:
- “Defining three more user IDs” on page 156
- “Testing logging on to a new user ID” on page 159
- “Creating three new parameter files” on page 157
- “Granting user IDs access to VSWITCH” on page 158
- “Testing logging on to a new user ID” on page 159

9.5.1 Defining three more user IDs

Define three more user IDs for Linux virtual servers in the USER DIRECT file named LINUX02 - LINUX04. You will need to use the DASD volumes you formatted in “Formatting DASD for minidisks” on page 145: two for each virtual server. You can repeat the definition of LINUX01 three times with the block copy ""3 prefix command. For example:

```bash
===> x user direct
====> /user linux01
...
""3 *
02142 USER LINUX01 LNX4VM 256M 1G G
02143 INCLUDE LNXDFLT
02144 OPTION APPLMON
```
This will create three more copies of the LINUX01 user definition. Modify them to have a user ID of LINUX02 - LINUX04, and give each correct DASD labels:

```
*  USER LINUX02 LNX4VM 256M 1G
   INCLUDE LNXDFLT
   OPTION APPLMON
   MDISK 100 3390 6677 3338 TM63CF MR LNX4VM LNX4VM LNX4VM
*  USER LINUX03 LNX4VM 256M 1G
   INCLUDE LNXDFLT
   OPTION APPLMON
   MDISK 100 3390 3339 3338 TM63DO MR LNX4VM LNX4VM LNX4VM
*  USER LINUX04 LNX4VM 256M 1G
   INCLUDE LNXDFLT
   OPTION APPLMON
   MDISK 100 3390 0001 3338 TM63DI 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:

```
====> top
====> /alloc
USER $ALLOC$ NOLOG
...
   MDISK A09 3390 000 001 TM63CF R
   MDISK A10 3390 000 001 TM63DO R
   MDISK A11 3390 000 001 TM63DI R
...
====> file
```

Check for overlaps and the single gap. Use the QUIT command to exit the USER DISKMAP file:

```
=> diskmap user
=> x user diskmap
====> pre off
====> 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:

```
=> directxa user
z/VM USER DIRECTORY CREATION PROGRAM - VERSION 5 RELEASE 1.0
EOJ DIRECTORY UPDATED AND ON LINE
```

You have now created three new user IDs that can be cloned to.

### 9.5.2 Creating three new parameter files

A new parameter must be created for each of the user IDs with the proper networking information. Perform the following steps:
1. Logoff of MAINT and logon to LNXMAINT.

2. Copy the LINUX01 parameter file three times:
   ```
   => copy linux01 parm-s11 d linux02  
   => copy linux01 parm-s11 d linux03  
   => copy linux01 parm-s11 d linux04  
   ```

3. Edit each of the three files replacing the appropriate network values. For example, in the LINUX02 PARM-S11, only the TCP/IP address and DNS name need to be modified as all other network and other values are the same:
   ```
   => x linux02 parm-s11 d  
   ramdisk_size=65536 root=/dev/ram1 ro init=/linuxrc TERM=dumb  
   HostIP=9.60.18.246 Hostname=gpok246.endicott.ibm.com  
   Gateway=9.60.18.129 Netmask=255.255.252.0  
   Broadcast=9.60.18.255 Layer2=0  
   ReadChannel=0.0.0600 WriteChannel=0.0.0601 DataChannel=0.0.0602  
   Nameserver=9.0.2.11 Portname=whatever Portno=0  
   Install=nfs://9.60.18.224/nfs/sles11/dvd1  
   UseVNC=1 VNCPassword=12345678  
   InstNetDev=osa OsaInterface=qdio OsaMedium=eth Manual=0  
   ====> file  
   => x linux03 parm-s11 d  
   ...
   ```

You should now have three new parameter files.

9.5.3 Granting user IDs access to VSWITCH

Modify the PROFILE EXEC on AUTOLOG1 191 to grant access to the VSWITCH for the three 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:
```
=> link autolog1 191 1191 mr  
=> acc 1191 f
```

Edit the PROFILE EXEC:
```
=> x profile exec f  
...  
/* Grant access to VSWITCH for each Linux user */  
'CP SET VSWITCH VSW1 GRANT S11CLONE'  
'CP SET VSWITCH VSW1 GRANT S11GOLD'  
'CP SET VSWITCH VSW1 GRANT LINUX01'  
'CP SET VSWITCH VSW1 GRANT LINUX02'  
'CP SET VSWITCH VSW1 GRANT LINUX03'  
'CP SET VSWITCH VSW1 GRANT LINUX04'  

/* XAUTOLOG each Linux user that should be started */  
'CP XAUTOLOG S11CLONE'  
'CP XAUTOLOG LINUX01'  
'CP XAUTOLOG LINUX02'  
'CP XAUTOLOG LINUX03'  
'CP XAUTOLOG LINUX04'  
====> file
```

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

```bash
=> set vswitch vsw1 grant linux03
Command complete
=> set vswitch vsw1 grant linux04
Command complete
```

Verify that the user IDs have access with the `QUERY VSWITCH ACCESSLIST` command:

```bash
=> query vswitch vsw1 acc
```

... Authorized userids:

- LINUX01
- LINUX02
- LINUX03
- LINUX04
- SYSTEM
- S11CLONE
- S11GOLD
- RDEV: B440.P00 VDEV: B440 Controller: DTCVSW2
- RDEV: B424.P00 VDEV: B424 Controller: DTCVSW1 BACKUP

You have now defined three new user IDs suitable for cloning the golden image to.

### 9.5.4 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:

```bash
LOGON LINUX02
NIC 0600 is created; devices 0600-0602 defined
z/VM Version 6 Release 1.0, Service Level 0901 (64-bit),
built on IBM Virtualization Technology
There is no logmsg data
FILES:   NO RDR,   NO PRT,   NO PUN
LOGON AT 08:44:26 EST WEDNESDAY 12/02/09
z/VM V6.1.0    2009-11-19 13:47
DMSACP723I A (191) R/O
DMSACP723I C (592) R/O
DIAG swap disk defined at virtual address 300 (64989 4K pages of swap space)
DIAG swap disk defined at virtual address 301 (129981 4K pages of swap space)
Do you want to IPL Linux from minidisk 100? y/n
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:

```bash
=> q osa
OSA 0600 ON NIC 0600 UNIT 000 SUBCHANNEL = 0002
   0600 DEVTYPE OSA       CHPID 00 0SD
...
```

```bash
=> q da
DASD 0100 3390 TM63CF R/W  3338 CYL ON DASD 63CF SUBCHANNEL = 0000
DASD 0101 3390 TM63DD R/W  3338 CYL ON DASD 63DD SUBCHANNEL = 0001
DASD 0190 3390 610RES R/O  107 CYL ON DASD 6280 SUBCHANNEL = 0009
DASD 0191 3390 TM6289 R/O  300 CYL ON DASD 6289 SUBCHANNEL = 000C
DASD 0190 3390 TV6283 R/O  146 CYL ON DASD 6283 SUBCHANNEL = 000A
DASD 019E 3390 TV6283 R/O  250 CYL ON DASD 6283 SUBCHANNEL = 000B
DASD 0300 9336 (VDSK) R/W  524288 BLK ON DASD VDSK SUBCHANNEL = 000E
DASD 0301 9336 (VDSK) R/W  1048576 BLK ON DASD VDSK SUBCHANNEL = 000F
DASD 0592 3390 TV6284 R/O  70 CYL ON DASD 6284 SUBCHANNEL = 000D
```

Logoff of LINUX02.
Congratulations, you have cloned one Linux virtual server and defined three 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.

9.6 Reviewing system status

Step back now and view your system from a DASD point of view. An example is shown in Figure 9-1. The DASD used by the examples in book are 12 3390-3 (8 for the /VM system and 4 for LNXMAINT and the majority of S11CLONE) and 4 3390-9s for the golden image and the remaining Linux virtual servers.

The possible role of administrators and users is suggested on the right side of this figure.

Figure 9-1   Linux virtual server system - DASD view and role view

These disk allocations can also be viewed by the **QUERY DASD** command (from MAINT):

```
=> q da
DASD 6280 CP OWNED 610RES   70
DASD 6281 CP OWNED TV6281   1
DASD 6282 CP OWNED TV6282   0
DASD 6283 CP OWNED TV6283   87
DASD 6284 CP OWNED TV6284   27
DASD 6285 CP OWNED TP6285   0
DASD 6286 CP OWNED TP6286   0
DASD 6287 CP OWNED TP6287   0
DASD 6289 CP SYSTEM TM6289   3
DASD 6290 CP SYSTEM TM6290   1
```
DASD 6293 CP SYSTEM TM6293  1
DASD 6294 CP SYSTEM TM6294  1
DASD 63CE CP SYSTEM TM63CE  1
DASD 63CF CP SYSTEM TM63CF  2
DASD 63D0 CP SYSTEM TM63D0  0
DASD 63D1 CP SYSTEM TM63D1  0

The next step will be to start cloning virtual servers for different uses.
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 163
- “Creating a virtual LDAP server” on page 164
- “Creating a virtual file and print server” on page 171
- “Creating a virtual application development server” on page 175

These Linux virtual servers can be thought of as **virtual appliances** once they have been cloned and **personalities** have been added to them.

### 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, “Cloning SLES 11” on page 145.

#### 10.1.1 Installing Apache RPMs

This section describes how to install the Apache Web Server RPMs.

1. **SSH** into the IP address of the new LINUX01 server.
2. Install the following Apache RPMs using the `yast -i` command:
   ```bash
   # yast -i apache2-prefork apache2 apache2-doc apache2-example-pages
   ```
3. 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 using the `rpm -qa` command:
   ```bash
   # rpm -qa | grep apache
   apache2-doc-2.2.10-2.18
   apache2-prefork-2.2.10-2.18
   apache2-example-pages-2.2.10-2.18
   ```
10.1.2 Testing Apache

Start the Apache Web server to verify it is installed successfully.

1. Start the apache server using the `service` command and set it to start at boot time using the `chkconfig` command:

   ```
   # service apache2 start
   Starting httpd2 (prefork) done
   # chkconfig apache2 on
   ```

2. 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. In this example the virtual server running on LINUX01 has a DNS name of gpok225.endicott.ibm.com:

   ```
   http://gpok225.endicott.ibm.com
   ```

3. You should see a test page with the two 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 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`.

10.1.4 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 using 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 clone Linux and configure a 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 golden 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:

- “Cloning a Linux” on page 165
- “Configuring the LDAP server” on page 165
- “Adding an LDAP user” on page 167
- “Setting another virtual server to use the LDAP server” on page 168

10.2.1 Cloning a Linux

From a root session on the cloner, clone from the golden image (S11GOLD user ID) to LINUX02 using the `clone.sh` script:

```bash
# clone.sh from s11gold to linux02
...
```

You should now have a new virtual server running.

10.2.2 Configuring the LDAP server

To configure the OpenLDAP server, the `yast` tool is recommended.

1. Start an SSH session as root to the new server.
2. Invoke the `yast` command. The YaST Control Center should appear:
3. Use the down arrow key to move to Network Services on the left side. Use the Tab or right arrow key to move to the right side and select LDAP Server. Press Enter.

```bash
# yast
```

4. You should see the following message. Press Enter to accept the default of Install and some RPMs will be installed:

```
These packages need to be installed: openldap2
```

5. The LDAP Server Configuration panel should appear. Accept the default of Yes by using the tab key to Next and pressing Enter.
6. In the **TLS settings** panel, accept all defaults of using TLS and press **Next**. This will ensure that LDAP communications are encrypted.

7. You should see the **New Database** panel as shown in the following example. The **Base DN** (in this example `dc=endicott,dc=ibm,dc=com`) should be correct for your DNS domain. Set the LDAP administrator password and press **Next**.

```
New Database
Basic Database Settings
Database Type
hdb
Base DN
dc=endicott,dc=ibm,dc=com
Administrator DN
cn=Administrator
[ ] Append Base DN
LDAP Administrator Password
******
Validate Password
******
Database Directory
/var/lib/ldap
[ ] Use this database as the default for OpenLDAP clients
```

8. The **LDAP Server Configuration Summary** panel should appear now, this time with one database listed identified by the Base DN (`dc=endicott,dc=ibm,dc=com` in this example). Move the cursor to **Finish** and press **Enter**.

```
LDAP Server Configuration Summary
+--------------------------------------------------------------------------+
| Startup Configuration                                                     |
|                                                                          |
| Start LDAP Server: Yes                                                   |
| Register at SLP Service: No                                              |
|                                                                          |
| Create initial Database with the following Parameters                   |
|                                                                          |
| Database Suffix: dc=endicott,dc=ibm,dc=com                               |
| Administrator DN: cn=Administrator,dc=endicott,dc=ibm,dc=com               |
+--------------------------------------------------------------------------+
```
9. The panel *Saving LDAP Server Configuration* should appear. The database will be created and the LDAP server configured.

10. You should be returned to the *YaST Control Center*. Move the cursor to *Quit* and press Enter.

11. Verify that the LDAP server is running with the `service` command and that it is set to start in run levels 3 and 5 with the `chkconfig` command:

```
# service ldap status
Checking for service ldap: running
# chkconfig --list ldap
ldap                      0:off  1:off  2:off  3:on  4:off  5:on  6:off
```

You have now cloned a new virtual server and configured it to run OpenLDAP.

### 10.2.3 Adding an LDAP user

When the golden image was installed, it was recommended that a non-root user ID be added. In this example, it was named `mikemac`.

1. Verify that this user exists with the `id` command and see that there is an entry in the `/etc/passwd` file with the `grep` command:

```
# id mikemac
uid=1000(mikemac) gid=100(users) groups=100(users),16(dialout),33(video)
# grep mikemac /etc/passwd
mikemac:x:1000:100::/home/mikemac:/bin/bash
```

2. Delete this local user using the `userdel` command so it can be added to LDAP later.

```
# userdel mikemac
no crontab for mikemac
# id mikemac
id: mikemac: No such user ...
```

3. An LDIF (LDAP Interchange Format) file is created to add an organizational unit named *People* and a user ID named `mikemac`. Create a similar file for your system's values.

```
# cd /var/lib/ldap
# vi initial.ldif // create the input file ...
# cat initial.ldif
dn: ou=People,dc=endicott,dc=ibm,dc=com
ou: People
objectClass: top
objectClass: organizationalUnit

dn: uid=mikemac,ou=People,dc=endicott,dc=ibm,dc=com
uid: mikemac
cn: mikemac
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
loginShell: /bin/bash
```
uidNumber: 501
gidNumber: 100
homeDirectory: /home/mikemac

4. Add the contents of the LDIF file to the LDAP server with the `ldapadd` command (the line wraps, but it is one command):

```
# ldapadd -x -h localhost -D "cn=Administrator,dc=endicott,dc=ibm,dc=com" -f initial.ldif -W
Enter LDAP Password:
adding new entry "ou=People,dc=endicott,dc=ibm,dc=com"
adding new entry "uid=mikemac,ou=People,dc=endicott,dc=ibm,dc=com"
```

5. Search for the new user ID just added with the `ldapsearch` command:

```
# ldapsearch -x uid=mikemac
# extended LDIF
#
# LDAPv3
# base <> with scope subtree
# filter: uid=mikemac
# requesting: ALL
#

# mikemac, People, endicott.ibm.com
dn: uid=mikemac,ou=People,dc=endicott,dc=ibm,dc=com
uid: mikemac
cn: mikemac
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
loginShell: /bin/bash
uidNumber: 501
gidNumber: 100
homeDirectory: /home/mikemac

# search result
search: 2
result: 0 Success
numResponses: 2
numEntries: 1
```

This shows that the user ID exists in the LDAP database. Now you may want to set the password with the `ldappasswd` command:

```
# ldappasswd -x -D "cn=Administrator,dc=endicott,dc=ibm,dc=com" -W -S
"uid=mikemac,ou=People,dc=endicott,dc=ibm,dc=com"
New password:
Re-enter new password:
Enter LDAP Password:
Result: Success (0)
```

You have now added a new LDAP user using an LDIF file and have set its password.

## 10.2.4 Setting another virtual server to use the LDAP server

Now that you have a virtual LDAP server, you may want to point another virtual server to it so you will have a centralized user database. If you have been following along in this book you
should have created a Web server running on the LINUX01 user ID. To point it to an LDAP server is fairly easy. In this section you will perform the following steps:

- “Testing that the LDAP client is not working” on page 169
- “Using YaST to modify the LDAP authentication client” on page 169
- “Testing the LDAP client” on page 170

**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.

Search for the LDAP user ID that you added earlier to the virtual LDAP server. In this example it is mikemac.

```
# ldapsearch -x uid=mikemac
ldap_sasl_interactive_bind_s: Can't contact LDAP server (-1)
```

The `ldapsearch` command cannot resolve the LDAP user because it cannot contact the LDAP server.

Delete the non-root user (mikemac in this example) from the local file system with the `userdel` command:

```
# userdel mikemac
no crontab for mikemac
```

**Using YaST to modify the LDAP authentication client**

The `yast` system administration interface can be used to configure the LDAP authentication client.

1. Invoke the `yast` command. The YaST Control Center should appear.
2. Select **Network Services** on the left side and **LDAP Client** on the right. Press **Enter**:

```
# yast

+---------------------+ +--------------------------------------------------+
¦ Software            ¦ ¦DNS and Hostname                                  ¦
¦ Hardware            ¦ ¦Hostnames                                         ¦
¦ System              ¦ ¦Kerberos Client                                   ¦
¦ Network Devices     ¦ ¦LDAP Browser                                      ¦
|Network Services     ¦ |LDAP Client                                      |
|Security and Users   ¦ |LDAP Server                                       |
|Miscellaneous        ¦ |Mail Transfer Agent                               |
|                    ¦ |NFS Client                                        |
...                   
```

3. On the panel that follows,
   
a. Use the Tab key to move to **Use LDAP** and press the **space bar** to select that choice.

b. Enter the IP address (or DNS name) 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.

c. Deselect LDAP TLS/SSL

```
+User Authentication-------------------------------+
|                                                   |
| ( ) Do Not Use LDAP                               |
```
4. Use the Tab key to move to Finish and press Enter. You should get the following prompt. Press Enter to continue:

These packages need to be installed:
- pam_ldap
- nss_ldap
- pam_ldap-32bit
- nss_ldap-32bit

5. Accept OK if you get a warning window.

6. Press the Quit button on the main window to quit YaST.

Your Web server virtual Linux should now also be using OpenLDAP for user and group ID resolution and authentication.

**Testing the LDAP client**

Try the `id` command against the new LDAP user:

```bash
# id mikemac
uid=501(mikemac) gid=100(users) groups=100(users)
```

Note that the UID is 501 in this example (from the LDIF file), not the value 1000 (from the Linux installation).

Try the `ldapsearch` command again:

```bash
# ldapsearch -x uid=mikemac
# extended LDIF
#
# LDAPv3
# base <dc=endicott,dc=ibm,dc=com> (default) with scope subtree
# filter: uid=mikemac
# requesting: ALL
#
# mikemac, People, endicott.ibm.com
dn: uid=mikemac,ou=People,dc=endicott,dc=ibm,dc=com
uid: mikemac
cn: mikemac
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
loginShell: /bin/bash
uidNumber: 501
gidNumber: 100
homeDirectory: /home/mikemac
```
You should also be able to start an SSH session to the virtual Web server using the LDAP user.

You may also want to set the golden image to authenticate with the LDAP server. In this fashion, all virtual servers cloned after that will be able to utilize a centralized authentication server.

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 171
- “Installing necessary RPMs” on page 171
- “Configuring Samba configuration file” on page 172
- “Adding a Samba user” on page 172
- “Starting Samba at boot time” on page 173
- “Testing your changes” on page 173

10.3.1 Cloning a Linux virtual server

Start an SSH session to the cloner as root, clone a basic virtual server. In this example the user ID LINUX03 is used.

```
# clone.sh from s11gold to linux03
...
```

SSH in to the new virtual server.

10.3.2 Installing necessary RPMs

Add the following RPMs with the `yast -i` command:

```
# yast -i samba yast2-samba-server samba-doc samba-winbind cups cups-drivers
```

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:

```
# rpm -qa | egrep "samba|cups"
cups-drivers-1.3.9-2.31
cups-libs-32bit-1.3.9-8.14
yast2-samba-client-2.17.11-1.35
```
The Samba and CUPS RPMs are now installed.

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 = SLES11 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 determines 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).

The following example shows adding the user mikemac to the smbpasswd file. In this example the user exists in the local file system. Perform the following steps:

1. Use the id command to verify the local user:

   ```bash
   # id mikemac
   uid=1000(mikemac) gid=100(users) groups=16(dialout),33(video),100(users)
   ```

2. To create a new Samba user, the smbpasswd -a command is used. If there are policy error messages, they can be ignored.

   ```bash
   # smbpasswd -a mikemac
   New SMB password:
   Retype new SMB password:
   account_policy_get: tdb_fetch_uint32 failed for field 1 (min password length), returning 0
   ...
   Added user mikemac.
   ```

3. You can see that the last smbpasswd command added mikemac to the file smbpasswd:

   ```bash
   # grep mikemac smbpasswd
   mikemac:1000:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX:A3C80B3B61B1209705705C67C03FC3C:[:U
   :]:LCT-4B1C0BC9:
   ```
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 10.2.4, “Setting another virtual server to use the LDAP server” on page 168, because the Samba schema must first be added to LDAP.

10.3.5 Starting Samba at boot time

Samba consists of two daemons \texttt{nmbd} and \texttt{smbd}. They can be started for the current session with the \texttt{service} command for both the \texttt{nmb} and \texttt{smb} daemons:

```
# service nmb start
Starting Samba NMB daemon done
# service smb start
Starting Samba SMB daemon done
```

The following \texttt{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 with the \texttt{service} command:

```
# service nmb status
Checking for Samba NMB daemon running
# service smb status
Checking for Samba SMB daemon running
```

You can test getting a Samba share from a Windows desktop.

1. Go to any Windows Explorer window (such as \textit{My Computer}) and select \texttt{Tools} -> \texttt{Map Network Drive}.

2. Use the Universal Naming Convention (UNC) to specify the Samba server and share name as shown in the upper left corner of Figure 10-1 on page 174. In this example the UNC is \texttt{\textbackslash 9. \textbackslash 60. \textbackslash 18. \textbackslash 247 \textbackslash sharedoc}.

3. You may need to click \texttt{Connect using different user name}, if the sample user ID and password are different on your desktop computer from the values you set on the Samba server.

4. Click \texttt{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.
You should now have Samba configured and running with one new share available.

If you prefer command line, you can map a network drive with the DOS `NET USE` command:

```
c:\> net use * \9.60.18.247\sharedoc
Drive T: is now connected to \9.60.18.247\sharedoc.
```

The command completed successfully.

**Note:** you may be prompted for the Samba user ID and password

Another useful DOS command to interrogate the Samba server is the `NET VIEW` command. For example:

```
c:\> net view \9.60.18.247
Shared resources at \9.60.18.247
```

```
Samba 3.2.7-11.6-2057-SUSE-CODE11

<table>
<thead>
<tr>
<th>Share name</th>
<th>Type</th>
<th>Used as</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>groups</td>
<td>Disk</td>
<td>All groups</td>
<td></td>
</tr>
<tr>
<td>mikemac</td>
<td>Disk</td>
<td>Home Directories</td>
<td></td>
</tr>
<tr>
<td>profiles</td>
<td>Disk</td>
<td>Network Profiles Service</td>
<td></td>
</tr>
<tr>
<td>sharedoc</td>
<td>Disk T:</td>
<td>SLES11 on System z documentation</td>
<td></td>
</tr>
<tr>
<td>users</td>
<td>Disk</td>
<td>All users</td>
<td></td>
</tr>
</tbody>
</table>
```

The command completed 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 robust 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 cloner clone a basic virtual server to LINUX04.

```
# clone.sh from sl1gold to linux04
```

SSH in to the new virtual server as root.

You can install all the application development tools described in this section with the following command:

```
# yast -i python perl tcl php gcc gdb make java-1_6_0-ibm
```

These RPMs use approximately 195MB in the /usr/ file system.

If you want to install only certain tools, each specific RPM or group of RPMs is described in the sections that follow.

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
libgcc43-4.3.3_20081022-11.18
gcc43-4.3.3_20081022-11.18
libgcc43-32bit-4.3.3_20081022-11.18
gcc-4.3-62.198
```

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
non-root user developer1 is used. The files readFile.c and writeFile.c are compiled into the executable files:

```bash
# 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:

```bash
$ 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:

```bash
# 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:

```bash
# yast -i make
```

## 10.4.4 Java development tools

SLES 11 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. The following command installs both:

```bash
# yast -i java-1_5_0-ibm
```

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:

```bash
# 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.

### 10.4.5 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

You have now cloned and installed four virtual Linux **appliances**.
Chapter 11. 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 DASD” on page 181
- “Adding a logical volume” on page 183
- “Extending an existing logical volume” on page 187

11.1 Adding DASD

The following process describes how to add additional DASD to a Linux guest. The overall steps are:

- “Adding minidisks to LINUX02” on page 181
- “Making the new minidisks available” on page 182
- “Formatting and partitioning the minidisks” on page 182
- “Creating a logical volume and file system” on page 183
- “Updating the file system table” on page 186

11.1.1 Adding minidisks to LINUX02

Following are the high level steps to add two new 3390-3-sized minidisks to LINUX02:

- Determine the labels of the two volumes that will be added.
- Add minidisk statements to define minidisks. In this example two minidisks at virtual addresses 102 and 103 are defined of size 1669 cylinders to the LINUX02 user ID.
- 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.

Following is the updated directory entry:

USER LINUX02 LNX4VM 256M 1G G
INCLUDE LNXDFLT
OPTION APPLMON
MDISK 100 3390 6677 3338 TM63CF MR LNX4VM LNX4VM LNX4VM
MDISK 101 3390 0001 3338 TM63D0 MR LNX4VM LNX4VM LNX4VM
MDISK 102 3390 6677 1669 TM63D1 MR LNX4VM LNX4VM LNX4VM
MDISK 103 3390 8346 1669 TM63D1 MR LNX4VM LNX4VM LNX4VM

11.1.2 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`):

```
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.0100   active      dasda     94:0    ECKD  4096   2347MB    600840
0.0.0101   active      dasdb     94:4    ECKD  4096   2347MB    600840
0.0.0300   active      dasdc     94:8    FBA   512    256MB     524288
0.0.0301   active      dasdd     94:12   FBA   512    512MB     1048576
```

Bring disks 102 and 103 online with the `dasd_configure` command and verify they are available with the `lsdasd` command:

```
# dasd_configure 0.0.0102 1
Configuring device 0.0.0102
Setting device online
# dasd_configure 0.0.0103 1
Configuring device 0.0.0103
Setting device online
```

```
# lsdasd
Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
==============================================================================
0.0.0100   active      dasda     94:0    ECKD  4096   2347MB    600840
0.0.0101   active      dasdb     94:4    ECKD  4096   2347MB    600840
0.0.0300   active      dasdc     94:8    FBA   512    256MB     524288
0.0.0301   active      dasdd     94:12   FBA   512    512MB     1048576
0.0.0102   active      dasde     94:16   ECKD  4096   1173MB    300420
0.0.0103   active      dasdf     94:20   ECKD  4096   1173MB    300420
```

You should be ready to format the new minidisks.

11.1.3 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:

```
# for i in e f
do
    dasdfmt -b 4096 -y -f /dev/dasd$i &
done
```
When the jobs are finished you should see messages similar to the following:

- Finished formatting the device.
- Rereading the partition table... ok
- Finished formatting the device.
- Rereading the partition table... ok

Now, use the `fdasd` command with the `-a` flag to create a single partition from each minidisk:

```bash
# fdasd -a /dev/dasde
auto-creating one partition for the whole disk...
writing volume label...
writing VTOC...
rereading partition table...
```

```bash
# fdasd -a /dev/dasdf
auto-creating one partition for the whole disk...
writing volume label...
writing VTOC...
rereading partition table...
```

The minidisks are now ready for you to use.

If you are creating a new logical volume, see 11.2.1, “Creating a logical volume and file system” on page 183. If you are extending an existing logical volume, skip ahead to 11.3, “Extending an existing logical volume” on page 187.

## 11.2 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 DASD” on page 181
- “Creating a logical volume and file system” on page 183
- “Updating the file system table” on page 186

### 11.2.1 Creating a logical volume and file system

The overall steps involved in creating a logical volume are:

- Create physical volumes from the two partitions
- Create a single volume group
- Create a single logical volume
- Make a file system from the logical volume

Figure 11-1 on page 184 shows a block diagram of the logical volume manager reflecting this example.
Figure 11-1  LVM block diagram

**Creating physical volumes from the two DASD**

The `pvcreate` command initializes partitions for use by LVM. Initialize the two new DASD partitions.

```
# pvcreate /dev/dasde1 /dev/dasdf1
No physical volume label read from /dev/dasde1
Physical volume "/dev/dasde1" successfully created
No physical volume label read from /dev/dasdf1
Physical volume "/dev/dasdf1" successfully created
```

Verify that the physical volumes were created with the `pvdisplay` command:

```
# pvdisplay /dev/dasde1 /dev/dasdf1
--- Physical volume ---
PV Name   /dev/dasde1
VG Name   homevg
PV Size   1.15 GB / not usable 1.42 MB
Allocatable yes
PE Size (KByte) 4096
Total PE    293
Free PE     293
Allocated PE 0
PV UUID     Ud2n4f-89dW-rCF2-tP4B-CBcM-cjYN-Iti2vE

--- Physical volume ---
PV Name   /dev/dasdf1
VG Name   homevg
PV Size   1.15 GB / not usable 1.42 MB
Allocatable yes
PE Size (KByte) 4096
Total PE    293
Free PE     293
```
Creating a single volume group

The `vgcreate` command can be used to create a volume group named `homevg` from the two partitions. Use the `vgdisplay homevg` command to verify the volume group was created:

```
# vgcreate homevg /dev/dasde1 /dev/dasdf1
Volume group "homevg" successfully created
# vgdisplay homevg
--- 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               2.29 GB
PE Size               4.00 MB
Total PE              586
Alloc PE / Size       0 / 0
Free  PE / Size       586 / 2.29 GB
VG UUID               P8Qu6k-unKh-dgFK-A5f2-whGo-zDLJ-vRoRVC
```

In this example, there are 586 free physical extents.

Creating a single logical volume

The `lvcreate` command is used to create a logical volume. The `-l` flag specifies to use all free extents, 586 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.

```
# lvcreate -l 586 -n homelv homevg
Logical volume "homelv" created
```

Use the `lvdisplay` command to verify. The parameter is the full path of the logical volume, not just the logical volume name:

```
# lvdisplay /dev/homevg/homelv
--- Logical volume ---
LV Name                /dev/homevg/homelv
VG Name                homevg
LV UUID                V6yuM6-Nv60-SGsK-7cQ2-xojr-mbYH-lSsDMc
LV Write Access        read/write
LV Status              available
# open                 0
LV Size                2.29 GB
Current LE             586
Segments               2
Allocation             inherit
Read ahead sectors     auto
- currently set to     256
Block device           253:4
```
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 1.41.1 (01-Sep-2008)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
150176 inodes, 600064 blocks
30003 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=616562688
19 block groups
32768 blocks per group, 32768 fragments per group
7904 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

This filesystem will be automatically checked every 38 mounts or 180 days, whichever comes first. Use `tune2fs -c` or `-i` to override.

The file system created from the logical volume is now ready to be mounted.

11.2.2 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:

```
# cp /etc/fstab /etc/fstab.orig
# vi /etc/fstab
/dev/disk/by-path/ccw-0.0.0300-part1 swap swap defaults 0 0
/dev/disk/by-path/ccw-0.0.0301-part1 swap swap defaults 0 0
/dev/disk/by-path/ccw-0.0.0100-part1 / ext3 acl,user_xattr 1 1
/dev/system-vg/opt-lv /opt ext3 acl,user_xattr 1 1
/dev/system-vg/tmp-lv /tmp ext3 acl,user_xattr 1 1
/dev/system-vg/usr-lv /usr ext3 acl,user_xattr 1 1
/dev/system-vg/var-lv /var ext3 acl,user_xattr 1 1
/dev/homevg/homelv /home ext3 defaults 0 0
proc /proc proc defaults 0 0
sysfs /sys sysfs noauto 0 0
devfs /sys/kernel/debug debugfs noauto 0 0
devpts /dev/pts devpts mode=0620,gid=5 0 0
```

Before mounting over `/home/`, check that it is empty. In this example a non-root user was created when installing the golden image, so that user has a directory under `/home/`. If you over-mount `/home/`, this directory will be covered. Move it temporarily to another location. In this example, `/tmp/` is used.

```
# ls /home
mikemac
# mv /home/mikemac /tmp
# ls /home
```
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 (created with the mke2fs command). Move the existing home directory back from /tmp/. Use the df -h command to verify that it is mounted:

```
# mount /home
# mv /tmp/mikemac/ /home
# ls /home
lost+found  mikemac
# df -h

Filesystem            Size  Used  Avail Use% Mounted on
/dev/dasda1           372M  100M  254M  29%  /
udev                  122M  140K  122M   1%  /dev
/dev/mapper/system--vg-opt--lv
  372M   17M  337M   5%  /opt
/dev/mapper/system--vg-temp--lv
  372M   17M  337M   5%  /tmp
/dev/mapper/system--vg-usr--lv
  2.0G  1.2G  787M  59%  /usr
/dev/mapper/system--vg-var--lv
  504M  57M  423M  12%  /var
/dev/mapper/homevg-homelv
  2.3G  68M  2.1G   4%  /home
```

When /home/ can be mounted with one parameter, it shows that the syntax in /etc/fstab is correct.

Since dasde and dasdf were brought online using dasd_configure command, they will stay online also after a reboot. Test a reboot to verify the new logical volume is successfully mounted over /home/.

11.3 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.

First, repeat the steps as described in 11.1, “Adding DASD” on page 181 to add a new minidisk. In this example, a minidisk at virtual address 104 is added of size 3338 cylinders. Don't forget to logoff and log back on to LINUX04 so the new directory entry is read.

When your system comes back, bring disk 104 online and check it is available:

```
# dasd_configure 0.0.0104 1
Configuring device 0.0.0104
Setting device online
# lsdasd

Bus-ID     Status      Name      Device  Type  BlkSz  Size      Blocks
------------------------------------------------------------------------
0.0.0100   active      dasda     94:0    ECKD  4096  2347MB    600840
0.0.0101   active      dasdb     94:4    ECKD  4096  2347MB    600840
0.0.0102   active      dasdc     94:8    ECKD  4096  1173MB    300420
0.0.0103   active      dasdd     94:12   ECKD  4096  1173MB    300420
0.0.0300   active      dasde     94:16   FBA   512    256MB     524288
0.0.0301   active      dasdf     94:20   FBA   512    512MB     1048576
0.0.0104   active      dasdg     94:24   ECKD  4096  1173MB    300420
```

Format the minidisk with the dasdfmt command and make a single partition with the fdasd -a command as you did before:
# dasdfmt -b 4096 -y -f /dev/dasdg
Finished formatting the device.
Rereading the partition table... ok

# fdasd -a /dev/dasdg
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:

```
# pvcreate /dev/dasdg1
No physical volume label read from /dev/dasdg1
Physical volume "/dev/dasdg1" 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.

```
# vgdisplay homevg
--- Volume group ---
  VG Name       homevg
  System ID
  Format       lvm2
  Metadata Areas  2
  Metadata Sequence No  2
  VG Access       read/write
  VG Status       resizable
  MAX LV         0
  Cur LV         1
  Open LV        1
  Max PV         0
  Cur PV         2
  Act PV         2
  VG Size       2.29 GB
  PE Size       4.00 MB
  Total PE      586
  Alloc PE / Size 586 / 2.29 GB
  Free PE / Size 0 / 0
  VG UUID     P8Qu6k-unKh-dgFK-A5f2-whGo-zDLJ-vRoRVC
```

```
# vgextend homevg /dev/dasdg1
Volume group "homevg" successfully extended
```

```
# vgdisplay homevg
--- Volume group ---
  VG Name       homevg
  System ID
  Format       lvm2
  Metadata Areas  3
  Metadata Sequence No  3
  VG Access       read/write
  VG Status       resizable
  MAX LV         0
  Cur LV         1
  Open LV        1
  Max PV         0
  Cur PV         3
  Act PV         3
  VG Size       3.43 GB
```
Note there are 293 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 of extents to add. Finally, use the `ext2online` command to increase the size of the file system while it is still mounted.

You can use the `df` command to show the file system size before and after you extend it as the following example shows:

```bash
# df -h /home
/dev/mapper/homevg-homelv  2.3G  68M  2.1G   4% /home
# lvextend -l +293 /dev/homevg/homelv
Extending logical volume homelv to 3.43 GB
Logical volume homelv successfully resized
# resize2fs /dev/homevg/homelv
resize2fs 1.41.1 (01-Sep-2008)
Filesystem at /dev/homevg/homelv is mounted on /home; on-line resizing required
old desc_blocks = 1, new desc_blocks = 1
Performing an on-line resize of /dev/homevg/homelv to 900096 (4k) blocks.
The filesystem on /dev/homevg/homelv is now 900096 blocks long.
```

Use the `df` `-h` command to show that the file system is now 1.1GB larger:

```bash
# df -h /home
Filesystem Size Used Avail Use% Mounted on
/dev/mapper/homevg-homelv  3.4G  69M  3.2G   3% /home
```

### 11.4 The X Window System

For many years UNIX-like operating systems have been using the X Window System (commonly just “X”). This system was designed to provide client/server, hardware-independent and network-enabled graphical environment. The current version is X11 which is widely used on UNIX and Linux platforms.

Confusion often arises among new X users regarding the concept of client and server, because client and server are defined from an application point of view where other protocols such as SSH, Telnet and FTP they are defined from an end user point of view. In X the server runs on the hardware with the mouse, keyboard and monitor (usually a workstation or a desktop), while the client runs on the UNIX or Linux server. Many Linux desktop users don't recognize this difference because they often run both the server and client on their desktop.

It is a common practice to connect from a PC (SSH client) to remote Linux (SSH server) and then run an X application. It runs on remote Linux (X client) and displays on local PC (X server).
The X communication protocol by its nature is not secure at all. For this reason it is often used together with SSH protocol, which tunnels X11 traffic via encrypted (and thus secure) communications.

X11 itself provides the ability to display graphics on raster display, nothing more. If the user wants to be able to move, resize and otherwise manage windows, a window manager is needed. There are many window managers available; some are lightweight while some are more robust. So using a window manager is a good idea because it provides functionality which one expects from a GUI.

When you have Linux installed on your workstation, a window manager is probably not enough. Here you want a full desktop environment with menus, icons, task bars etc such as Gnome and KDE.

When installing SLES11, there are three groups of RPMs available under Graphical Environment.

### Software Selection and System Tasks

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Base Technologies</th>
<th>Graphical Environment...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GNOME Desktop Environment</td>
<td>KDE Desktop Environment</td>
</tr>
<tr>
<td></td>
<td>X Window System</td>
<td></td>
</tr>
</tbody>
</table>

**X Window System**

The X Window System provides the only standard platform-independent networked graphical window system bridging the heterogeneous platforms in today’s enterprises: from network servers to desktops, thin clients, laptops, and handhelds, independent of operating system and hardware.

![Figure 11-2  Graphical Environments in SLES11](image)

Installing GNOME or KDE on System z is discouraged as they are resource-intensive. Installing The X Window System is also not recommended. Therefore it is recommend to unselect them all during installation.

### 11.4.1 VNC Server

As mentioned earlier, the X server is run where the mouse, keyboard and monitor are located - on the workstation. In a nutshell, VNC Server provides virtual workstation with all this
peripherals (virtual). The VNC server starts an embedded X server. Then any X-based application can send its output to this X server, regardless of if the applications is local or remote to X server.

To interact with the X server, one uses VNC client on a workstation, as described in section 3.2, “Setting up a VNC client” on page 23. The VNC server customization is described in section 8.5.1, “Configuring the VNC server” on page 135. In our experience this is all you need if you want to run X applications from time to time.

One big advantage of VNC is that it is session oriented. If communication to VNC server is lost, a new connection is reestablished to the session as it was. Also, applications in a disconnected VNC session still continue to run.

### 11.4.2 X Server on workstation

If for some reason VNC is not acceptable, it is possible to use a standard X server on a workstation. Since Linux users usually know the X Window system, an X server running on Windows is described in this section.

There are many commercial and free X Window servers available for Windows. In the following examples XliveCD is used, which provides a free X server based on Cygwin. It can be run directly from a CD without requiring installation.

http://xlivecd.indiana.edu/

Any X application will send its output to an address defined with `-display` parameter or, if not provided, to an address specified in the `DISPLAY` environment variable. If neither is provided, the local computer is used for output.

```
$ xclock
Error: Can't open display:
```

There is no display specified for `xclock` command and it will terminate.

The next example uses the X server on an Ubuntu system on the local network.

```
marian@ubuntu:~$ export DISPLAY=192.168.5.22:0
marian@ubuntu:~$ xclock &
[1] 22705
marian@ubuntu:~$ xclock
[1]+ Exit 1
```

This command failed, because the XliveCD requires an explicit command to allow remote hosts to connect to it. When the command `xhost +` is run, `xclock` can finally display on Windows as shown in Figure 11-3. Remember the program itself runs on a remote Linux.

```
marian@ubuntu:~$ xclock &
[1] 22724
```
The xhost + command allows any host to access the X Server. From a security point of view, this may not be a good idea. Even allowing just specific hosts is not enough, because X11 protocol itself is not secure. Using SSH tunneling removes this security exposure. SSH tunneling also prevents firewalls and NAT from breaking X11 communications.

It is possible to use an external SSH client which allows X11 forwarding, or SSH client embedded in XliveCD itself. Both options are shown.

**Using PuTTY**

To use PuTTY for X11 forwarding, select X11 forwarding as shown in Figure 11-4.
Chapter 11. Miscellaneous recipes

As you can see in Figure 11-5, the DISPLAY environment variable contains the special value of \texttt{localhost:10.0} which tells PuTTY to forward X11 protocol over SSH to SSH client address.

**Using embedded SSH**

It is also possible to achieve X11 forwarding with an embedded SSH client as shown below.
There are many ways how to achieve the same results. It is up to you to choose a solution which suits the purpose best.

11.5 Centralizing home directories for LDAP users

In previous versions of this book there was a section on how to create a travelling /home/ directory using LDAP, NFS and automount. In the interest of time, this section has been removed. The recipe for SLES 10 SP2 should be useful on SLES 11, though there is no promise it will work identically. See section 11.3 in the IBM Redbook z/VM and Linux on IBM System z The Virtualization Cookbook for SLES 10 SP2, SG24-7493, on the Web at:

http://www.redbooks.ibm.com/abstracts/sg247493.html

However, the following section has been added to this book. In December of 2009, the topic of how to set up a common home directory came up on the linux-390 list server. The following post by Patrick Spinler is copied, with permission, as it may be helpful to you:

11.5.1 Recommendations for centralizing home directories

"NFSv3 is not known for it's security. Consider the use of the NFS option root_squash, along with limiting the list of hosts who can connect to your home share. Only export home directories to hosts which you control, remember that anyone who has root on their box (e.g. a developer workstation) can impersonate any user to NFS. Here's the relevant /etc/exports line we use:

/export/unixdata/homedirs
  @grp_autohome_admin(rw,no_root_squash,insecure, sync) \
  @grp_autohome_hosts(rw,root_squash,insecure, sync)"
I look forward to going to NFSv4 with kerberos authentication, but we're not there yet.

Regarding automount maps in LDAP, this works very well for us with one exception. The problem is that there's a significant number of automount map schemas out there, and different OS's (and different revisions of OS's) use different ones. As we are a fairly heterogeneous environment, I found it near impossible to keep a master map in LDAP. Right now we're just keeping a `/etc/auto.master` or `/etc/auto_master` on each host.

In order to make the individual map entries work heterogeneously, I had to add several object classes and a few redundant attributes to each entry. Here's what my home directory automount map entry looks like:

```
# ap00375, auto_home, unix.example.com
dn: automountKey=ap00375,automountMapName=auto_home,dc=unix,dc=example,dc=com
automountInformation: linux01.example.com:/vol/vol2/unixhomes-5gb/75/ap00375
cn: ap00375
automountKey: ap00375
objectClass: automount
objectClass: nisNetId
objectClass: top
```

Regarding heterogeneous clients, we found AIX in particular to be the hardest of our clients to configure, and Linux the easiest. Insure on AIX that you have the latest available LDAP client package from IBM. Also be aware that AIX wants to use it's extended LDAP schema rather than RFC2307, and wants full write access to the LDAP servers from every AIX client. Despite that, it will work with RFC2307 and read only access. Solaris, like Linux, has an option to not use an LDAP proxy account at all via anonymous binding, but I never got Solaris anonymous binding to work.

I recommend making LDAP use TLS or SSL on the wire, in order to keep clear-text passwords from flying about. Both AIX and Solaris require the server public SSL certificates to be loaded on every client to do LDAP over TLS or SSL. Linux can be configured to ignore authenticating the LDAP servers' certificates and proceed with TLS/SSL anyway - this is convenient, but does open the possibility of man in the middle attacks. In our environment this isn't a big deal, but it might be in yours.

We've found POSIX group membership management to be one of our more challenging issues overall. Some older systems (e.g. solaris <= 8 or 9) enforce the old POSIX limit of no more than 16 secondary groups. Further, the primary group concept is annoying - conceptually, in any organization with modest member mobility, which primary group do they get? If one assumes that the primary group is meaningful, e.g. reflective of someone's function, role, or job, what about people who do two or more things (E.g. student *and* employee) or people who transfer, but will have a transitional period?

Our not so great compromise was to first use NIS-style netgroups via LDAP for anything we can. In particular, we use a mutation of netgroups to control individual's authorization to log in via the use of service search descriptors, and also for sudo privileges. Second in our environment all meaningful POSIX groups are secondary groups. For primary groups we adopt the linux convention of creating a separate POSIX group for each individual: e.g. userA gets a group userA as her primary group. This has the problem of a huge proliferation of groups, though, and several LDAP clients, in particular AIX, have issues with that."
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:


There are a number of z/VM monitoring tools such as CA’s VM:Monitor, IBM’s z/VM Performance Toolkit, IBM’s Tivoli OMEGAMON XE for z/VM and Linux, and products from Velocity Software. The IBM z/VM Performance Toolkit is briefly described in this section.

This chapter contains the following sections:
- “Using INDICATE and other commands” on page 197
- “The z/VM Performance Toolkit” on page 201
- “Monitoring Linux” on page 209
- “Viewing Linux data in the Performance Toolkit” on page 212

12.1 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. For more information, see the z/VM Performance Resources Web page at:

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

12.1.1 Using the INDICATE command

z/VM has some basic commands such as INDICATE. This can be useful if there are no other tools with more extended function present on the system, such as when undertaking a completely new system installation, for example. Command level tools should not be relied
upon for formal long term reporting on production systems. There are many **INDICATE**
parameters that can be included as command line options. 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,
depending on whether the issuing userid has privilege class G or 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:

```plaintext
=> 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 DORMANT-00357
6 Q0-000000(00000) E1-000000(00000)
7 Q1-000001(00000) EXPAN-002 E2-000000(00000)
8 Q2-000034(00000) EXPAN-002 E3-000000(00000)
9 PROC 0000-038% PROC 0001-038%
10 PROC 0002-038%
11 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 users ready to run virtual
  processors to keep the physical processors busy. One of the processors will be a Master,
  all of the others Alternate, and some imbalance may result from performing these
  functions. Line 2 describes paging to expanded storage. Most z/VM systems on z9 class
  machines can sustain several 1000s of this type of paging operations a second without
  any problems. z10 class machines will perform even better. 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,
  probably being measures in 100s rather than 1000s. The higher values seen tend to build
  up over time, and are sustained over periods of intense system activity, however, there
  are times the **MIGRATE** value may 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. For a system which
  has an appreciably high I/O rate, composed of reads plus writes, and a high proportion of
  reads, and a good hit ratio for those reads (tending to 90% or greater), the real, physical
  I/O avoidance can be very high, this author has seen the avoidance as high as 50% in
  some cases. Conversely, however, a high **HIT RATIO** with a low value for the **READS** rate
  should not be taken as good, (100% hit ratio, when doing only 1 I/O per second is
effectively meaningless).
Line 4 describes more storage (memory) management. The PAGING rate is important. Higher values will often impact performance. This can be at least partially offset by increasing the number of page volumes, but a more thorough examination of this problem is advisable whenever it arises. 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 still considered to be stolen. So there are scenarios where a system only has a user set comprising 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 relatively 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 additional class labelled zero. So the Column of Qx values and Ex 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. Such a system would require further investigation, possibly leading to some tuning work, or even hardware addition in extreme cases. 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
  EDLLNK4    Q3 PS  00007635/00007635  ....  -1.121 A00
  TCPIP      Q0 R01 00004016/00003336  ..I.  -.9324 A01
  APCTEST1   Q2 I0  00003556/00003512  ..I.  -.7847 A01
  EDLWK20    Q3 AP  0001495/00001462    ....  -6996 A01
  EDL        Q3 I0  00000918/00000902  ....  -2.409 A01
  EDLWK11    Q3 AP  0002323/00002299  ....  -0.183 A00
  EDLWK18    Q3 I0  0001052/0000388    ....  -0.0437 A00
  EDLWK4     Q3 AP  0004792/00002295  ....  .0055 A01
  EDLWK8     Q3 AP  0004804/00004797    ....  .0089 A02
  EDLWK16    Q3 AP  0002378/0002378    ....  .0170 A02
  EDLWK2     Q3 AP  0005544/0002956    ....  .0360 A00
  EDLWK12    Q3 AP  0004963/00002348    ....  .0677 A01
  EDLWK6     Q3 I0  00000750/00000302  ....  .0969 A02
  EDLWK3     Q3 AP  0005098/0005096    ....  .0999 A02
  EDLWK17    Q3 AP  0004786/0004766    ....  .1061 A01
  EDLWK9     Q3 AP  0002372/0002334    ....  .1107 A02
  EDLWK5     Q3 I0  0002376/0002376    ....  .1205 A01
  EDLWK14    Q3 AP  00002426/00002323   ....  .1238 A02
  EDLLB19    Q3 I0  0001226/0001100    ....  .1309 A02
  EDLWK19    Q3 AP  00002322/00002298   ....  .1705 A00
  EDLWK15    Q3 AP  0002839/0002781    ....  .2205 A02
  EDLWK1     Q3 AP  0002969/0002935    ....  .2491 A02
```

This is another class E command and displays the virtual processors associated with a given userid (a single virtual machine may have multiple virtual processors) what queue (dispatch list, eligible list, limit list) they are in and what state they are in. This is a snapshot in time. Again you want to check this output to make sure there are no virtual machines in the eligible list. Normal virtual processors in the dispatch list will be Qx (x=1,2,3). Eligible list would be marked as Ex. The third column in the example also gives state of virtual processor. This can
be helpful to get a idea of how the virtual processors might be constrained. Virtual processors that are actually running in the snapshot period 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, a single snapshot cannot be regarded as indicative.

12.1.2 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

```bash
===> help // for basic help
===> help menus // for menu of all z/VM help menus
===> 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:

```bash
===> q n
FTPSERVE - DSC , LINUX04 - DSC , LINUX03 - DSC , LINUX02 - DSC
LINUX01 - DSC , S11CLOVE - DSC , DTCVSW2 - DSC , DTCVSW1 - DSC
VMSERVR - DSC , VMSERVU - DSC , VMSERV - DSC , TCPIP - DSC
OPERSYMP - DSC , DISKACNT - DSC , EREP - DSC , OPERATOR - DSC
MAINT - L0003
VSM - TCPIP
```

Determining storage or memory

To see how much central and expanded storage (memory) are installed and allocated to a system use the query storage and query xstor commands. For example:

```bash
===> q stor
STORAGE = 16G CONFIGURED = 16G INC = 256M STANDBY = 0 RESERVED = 0
===> q xstor
XSTORE= 2048M online= 2048M
XSTORE= 2048M userid= SYSTEM usage= 0% retained= 0M pending= 0M
XSTORE MDC min=0M, max=0M, usage=0%
XSTORE= 2048M userid= (none) max. attach= 2048M
```

Determining processors or CPUs

To see how many processors (CPs, IFLs, CPUs) you have allocated at system level, use the query processors command. For example:

```bash
===> q proc
PROCESSOR 00 MASTER CP
PROCESSOR 01 ALTERNATE CP
PROCESSOR 02 ALTERNATE CP
PROCESSOR 03 ALTERNATE CP
```
Determining software level
To determine what level of CP your system is at, use the QUERY CPLEVEL command. For example:

```plaintext
==> q cplevel
z/VM Version 6 Release 1.0, service level 0901 (64-bit)
Generated at 12/15/09 13:41:46 EST
IPL at 12/15/09 13:50:21 EST
```

Determining system cylinder allocation
The QUERY ALLOC MAP command shows you the system allocation of spool, paging and directory space. For example:

```plaintext
==> q alloc map
q alloc map
EXTENT     EXTENT                         % ALLOCATION
VOLID  RDEV      START        END  TOTAL IN USE   HIGH USED TYPE
------ ---- ---------- ---------- ------ ------ ------ ---- --------------
610RES 6280          1         20     20      1      2   5% DRCT ACTIVE
TV6281 6281          1       3338 600840  91592 100907  15% SPOOL
TV6282 6282          1       3338 600840      0      0   0% PAGE
TP6285 6285          0       3338 601020      0      0   0% PAGE
TP6286 6286          0       3338 601020     17     21   1% PAGE
```

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. The following list gives the short form of these commands without any of the associated output shown:

```plaintext
==> q da
==> q da free
==> q osa
==> q osa free
==> q v all
```

12.2 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 Guide, SC24-6156, z/VM Performance Toolkit Reference, SC24-6157, on the Web starting at the z/VM 5.4 bookshelf:
  [http://www-03.ibm.com/systems/z/os/zos/bkserv/zvmpdf/#zvm61](http://www-03.ibm.com/systems/z/os/zos/bkserv/zvmpdf/#zvm61)

  Search for Toolkit on that page. The Program Directory for Performance Toolkit for VM, GI10-0785-00,
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 202
- “Using the z/VM Performance Toolkit” on page 206

### 12.2.1 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.

1. Query which priced products are enabled with the `QUERY PRODUCT` command:

   ```
   ==> q product
   Product  State    Description
   6VMDIR10 Disabled 00/00/00.00:00:00.$BASEDDR DIRECTORY MAINTENANCE FL 610
   6VMPTK10 Disabled 00/00/00.00:00:00.$BASEDDR PERFORMANCE TOOLKIT FOR VM
   6VMRAC10 Disabled 00/00/00.00:00:00.$BASEDDR RACF for VM
   6VMRSC10 Disabled 00/00/00.00:00:00.$BASEDDR RSCS Networking Version 6 Release 1 Modification 0
   ```

2. To enable the z/VM Performance Toolkit, logon to `MAINT` and enter the following command:

   ```
   ==> 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.

3. At IPL time the `SYSTEM CONFIG` file is modified by having a line appended to the end. Verify this has been added by the `SERVICE` command with the following commands:

   ```
   ==> link * cfl cfl rr
   ==> acc cfl f
   ==> x system config f
   ===> bot
   ===> -z
   ===> pre off
   ...
   PRODUCT PRODID 6VMPTK10 STATE ENABLED DESCRIPTION '12/17/09.15:35:41.MAINT PE RFKIT Minidisk Install and Service'
   ```

The Performance Toolkit is now enabled.
12.2.2 Configuring Web Browser support

Once the product is enabled, the TCPIP profile must be modified to enable Web access to the Performance Toolkit. The following example sets the port to 80, the default for a Web browser:

1. Logon to TCPMAINT. Edit the TCPIP configuration file. In this example it is POKTST61 TCPIP D file (assuming you modified this file name earlier - the default name is PROFILE TCPIP) and search for the string reserve ports. This is where z/VM TCP/IP ports are reserved.

```plaintext
=> x poktst61 tcpip d
====> /port
```

2. Add the following line under the PORT entries:

```plaintext
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 DHCPD               ; DHCP Server
 ; 69  UDP TFTPD               ; TFTPD (Trivial FTP) Server
 ; 69  UDP TFTPD               ; TFTPD (Trivial FTP) Server
   80  TCP PERFSVM             ; Performance Toolkit
 ; 111 TCP PORTMAP             ; Portmap Server

... 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 a position to reIPL the system, you can do that (shutdown reipl iplparms cons=sysc)

3. 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:

```plaintext
=> netstat clients
...
Client: PERFSVM Authorization: {none}
Notes Handled: none
Last Touched:  0:01:22
Vmcf error count: 0
```

The entry for PERFSVM should be at the end of the output.

12.2.3 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:

```plaintext
FCX001                 Performance Toolkit for VM                Autoscroll 12
FCX8AS5001 Performance Toolkit for VM FL610
Monitor event started -- recording is activated
Monitor sample started -- recording is activated
FCXPMN446E Incomplete monitor data: SAMPLE CONFIG size too small
```

2. Press **F12 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 the VMLINK 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 59, then it will be lnx4vm.

```bash
==> 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:

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

4. Copy the default configuration files, which are on PERFSVM’s D disk, to your A disk:

```bash
==> copy * * d = = a
```

5. The main configuration file is FCONX $PROFILE. Edit that file and search for the string VMCF.

```bash
==> x fconx $profile
====> /vmcf
```

This should take you to line 175 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 port 81 to 80 on the fourth line
- Add the text FC MONCOLL LINUXUSR ON after the fourth line. This tells the Performance Toolkit to collect Linux performance data.

The modified lines should be as follows. Save your changes with the FILE subcommand:

```bash
* Following command activates VMCF data retrieval interface
FC MONCOLL VMCF ON
* Following command activates Internet interface
FC MONCOLL WEBSERV ON TCPIP TCPIP 80
FC MONCOLL LINUXUSR ON
* Following command activates Internet interface with SSL
*C MONCOLL WEBSERV ON SSL TCPIP TCPIP 81 IDTEST RACF
...
====> file
```

6. Create a remote data retrieval authorization file with your z/VM system identifier (replace POKTST61 with your system identifier):

```bash
==> x fconrmt authoriz
====> a 2
POKTST61 PERFSVM S&FSERV
POKTST61 MAINT DATA CMD EXCPMSG
```

7. Create a system identification file that links your z/VM system and PERFSVM to a special resource name called FCXRES00. (replace POKTST61 with your system identifier):

```bash
==> x fconrmt systems
====> a
POKTST61 PERFSVM ESA N FCXRES00
```

8. Create a Linux system definition file. Use port 8803 for Linux performance data. Add the TCP/IP addresses of your Linux systems. The following example shows adding five Linux virtual servers (replace IP addresses with values for your enterprise):

```bash
==> x fconx linuxusr a
S11CLONE 9.60.18.224:8803
LINUX01 9.60.18.225:8803
LINUX02 9.60.18.246:8803
LINUX03 9.60.18.247:8803
```
9. Edit the PROFILE EXEC file and uncomment the five MONITOR SAMPLE and the two MONITOR EVENT statements. Also increase the size of the of the SAMPLE CONFIG and EVENT CONFIG:

```bash
==>
x profile exec a

Before:
...
/*** Once you have PERFKIT enabled and running uncomment the ***/
/*** following comments ***/
/* 'CP MONITOR SAMPLE ENABLE PROCESSOR' */
/* 'CP MONITOR SAMPLE ENABLE STORAGE' */
/* 'CP MONITOR SAMPLE ENABLE USER ALL' */
/* 'CP MONITOR SAMPLE ENABLE I/O ALL' */
/* 'CP MONITOR SAMPLE ENABLE APPLDATA ALL' */
/* 'CP MONITOR EVENT ENABLE STORAGE' */
/* 'CP MONITOR EVENT ENABLE I/O ALL' */

'PERFKIT' /* Invoke the PERFKIT module @FC012BD*/
Exit

After:
...
/*** Once you have PERFKIT enabled and running uncomment the ***/
/*** following comments ***/
'CP MONITOR SAMPLE ENABLE PROCESSOR'
'CP MONITOR SAMPLE ENABLE STORAGE'
'CP MONITOR SAMPLE ENABLE USER ALL'
'CP MONITOR SAMPLE ENABLE I/O ALL'
'CP MONITOR SAMPLE ENABLE APPLDATA ALL'
'CP MONITOR EVENT ENABLE STORAGE'
'CP MONITOR EVENT ENABLE I/O ALL'

/* Increase the size of the SAMPLE CONFIG and EVENT CONFIG */
'CP MONITOR SAMPLE CONFIG SIZE 1500'
'CP MONITOR EVENT CONFIG SIZE 68'
'CP MONITOR START PARTITION 324'

'PERFKIT' /* Invoke the PERFKIT module @FC012BD*/
Exit
===> file

12.2.4 Increasing the size of the MONDCSS DCSS

The DCSS named MONDCSS shipped with z/VM 6.1 is not large enough. To enlarge DCSS, first determine where the current MONDCSS is located by entering the following command:

```bash
===> q nss name mondcss map
FILE FILENAME FILETYPE MINSIZE BEGPAG ENDPAG TYPE CL #USERS PARMREGS VMGROUP
0011 MONDCSS CPDCSS N/A 09000 09FFF SC R 00001 N/A N/A
```

In this example, the DCSS starts at x9000 and ends at x9FFF. Before starting the Performance Toolkit, it is recommended that you increase the size of the DCSS named MONDCSS. First, delete the old MONDCSS by issuing:

```bash
===> purge nss name mondcss
NO FILES PURGED
0001 FILE PENDING PURGE
```
Then redefine the DCSS larger with the following `DEFSEG` and `SAVESEG` commands:

```
=> defseg mondcss 2200-4fff sc rstd
HCPNSD440I Saved segment MONDCSS was successfully defined in fileid 0376.
```

```
=> saveseg mondcss
```

You should now be ready to run the Performance Toolkit.

### 12.2.5 Starting the z/VM Performance Toolkit

To start the Performance Toolkit enter the following command from the `PERFSVM` user ID:

```
=> perfkit
```

`PERFSVM` connected on path 0006, for S&F-Coll

```
FCXBAS500I Performance Toolkit for VM FL610
FCXAPP530I Connected to *IDENT for resource FCXRES00
FCXAPF530I Connected to *IDENT for resource FCXSYSTEM
FCXTCP571I Connected to TCP/IP server TCPIP on path 0003
FCXTCR571I Connected to TCP/IP server TCPIP on path 0004
FCXAPP527I User PERFSVM connected on path 0007
FCXAPC535I Connected to resource FCXRES00 on path 0006, for S&F-Coll
FCXTCP575I WebServer host IP address is 9.60.18.249:00080
FCXTCR575I TCPRequest host IP address is 9.60.18.249:41780
FCXTCP590I TCPRequest interface activated
FCXTCP590I WebServer interface activated
Monitor event started -- recording is activated
Monitor sample started -- recording is activated
```

Disconnect from `PERFSVM` now.

```
Command => disc
```

The Performance Toolkit should now be configured and running.

### 12.2.6 Using the z/VM Performance Toolkit

The Performance Toolkit can be used with 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. For example:

   `http://9.60.18.249`

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.
3. Enter any valid user ID and password and click **Submit**. In this example MAINT is 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 12-2 on page 208.
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
FCXBAS550I Performance Toolkit for VM FL530
Monitor event started -- recording is activated
Monitor sample started -- recording is activated
...
FCX001     Performance Toolkit for VM
FCXBAS550I Performance Toolkit for VM FL530
HCPMOF6229E Monitor event collection is already active.
HCPMOG6229E Monitor sample collection is already active.

Command ==>> monitor
```
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:

- User resource usage
- User paging load
- User wait states
- User configuration
- Linux systems
- Benchmark displays

For example to drill down into the Benchmark submenu screen, enter the following command:

```
Command ===> 33
```

Then type S over the period on the left side of the submenu screen in the row corresponding to the report you wish to see

### 12.3 Monitoring Linux

Measurements can show resource consumption of the Linux guest as measured and dispatched by the VM host. It is also possible to measure performance data from within the Linux guest itself. To monitor Linux performance data at this level, a data gatherer process must be running within each Linux guest you wish to monitor. There are different ways of gathering this data. Data can be gathered in the kernel and as a user application. Novell
SuSE SLES9, SLES10, SLES 11 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.

12.3.1 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 with a browser, or, if you have access to the Internet you can use an FTP client. SLES 11 is a 64-bit distribution:

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 cloner or any other virtual server.

1. Change directory to /usr/local/src/:

   # cd /usr/local/src

2. If your Linux system has access to the Internet, you can download the appropriate tar file with the wget command:

   --17:26:26--
   Resolving ftp.software.ibm.com... 9.17.252.40
   Connecting to ftp.software.ibm.com[9.17.252.40]:21... connected.
   Logging in as anonymous ... Logged in!
   ...
   100%[====================================>] 1,097,125 538.93K/s
   17:26:32 (537.12 KB/s) - `rmfpms_s390x_kernel26.tgz' saved [1097125]

3. For RHEL 4 and RHEL 5 use the following wget command:


4. Untar the file with the tar command and change to the rmfpms/ directory:

   # tar xzf rmfpms_s390x_kernel26.tgz
   # ls
   rmfpms  rmfpms_s390x_kernel26.tgz
   # cd rmfpms/
   # ls
   .rmfpms_config  README  bin  enable_autostart
   .rmfpms_config_autostart  autostart_rmfpms  doc  rmfpms
5. You should now be able to start `rmfpms` in the `bin/` directory with the following command:

```
# 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!
```

6. 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 12-4.

![Browser view of rmfpms interface](image)

You can also register Linux images with the Performance Toolkit - see section 12.4, “Viewing Linux data in the Performance Toolkit” on page 212.

### 12.3.2 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 cloner 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.

1. 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
```

2. There is no output so no modules with the string `appldata` are loaded. Load those modules with the `modprobe` command and verify they have been loaded:

```bash
# modprobe appldata_mem
# modprobe appldata_os
# modprobe appldata_net_sum
```

3. Now if you repeat the `lsmod` command, you should see the following:

```bash
# lsmod | grep appldata
appldata_net_sum       20064  0
appldata_os            22536  0
appldata_mem           20576  0
```

4. 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.

5. 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.

### 12.4 Viewing Linux data in the Performance Toolkit

After the system has had some time to collect data, you should be able to use the Performance Toolkit to view Linux performance data. There should be entries in the `FCONX LINUXUSR` file as completed in 12.2.2, “Configuring Web Browser support” on page 203. To view that data, drill down into menu 29, Linux systems. This can be done either from the browser interface or the 3270 interface as shown in the following figure.
Then type $ over the period on the left side of the submenu screen in the row corresponding to the report you wish to see. You should see a new report screen with the Linux guest systems CPU overview.
Reference Material

This appendix refers to additional material that can be downloaded from the Web.

A.1 Related books

The following publications can be used as information sources:

- *Linux for zSeries and S/390 Device Drivers, Features, and Commands*, LNUX-1403

- *SUSE LINUX Enterprise Server* documentation - on CD1 or DVD1 in docu/ directory:
  - *SUSE LINUX Enterprise Server* INSTALLATION AND ADMINISTRATION
  - *SUSE LINUX Enterprise Server*: ARCHITECTURE-SPECIFIC INFORMATION
  - *SUSE LINUX Enterprise Server*: START-UP GUIDE

- *z/VM* documentation - start at
  - *z/VM* Guide for Automated Installation and Service
  - *z/VM* CP Messages and Codes
  - *z/VM* TCP/IP Messages and Codes
  - The Program Directory for Performance Toolkit for VM
  - *z/VM* CP Commands and Utilities Reference
  - *z/VM* CP Planning and Administration
  - *z/VM* Getting Started with Linux on System z9 and zSeries
  - *z/VM* TCP/IP Planning and Customization
  - *z/VM Performance Toolkit Guide*, SC24-6156-00
  - *z/VM Performance Toolkit Reference*, SC24-6157-00

- *Redbooks* - start at:
  http://www.redbooks.ibm.com/
  - *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®:
- 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 12-1 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 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 to the system are defined here (assuming a directory maintenance product is 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 TCP/IP address, OSA resources, subnet mask and gateway. It is initially 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 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. 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 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**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>Add a line</td>
</tr>
<tr>
<td>a&lt;n&gt;</td>
<td>Add 'n' lines</td>
</tr>
<tr>
<td>c/&lt;old&gt;/&lt;new&gt;/ &lt;n&gt;</td>
<td>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'</td>
</tr>
<tr>
<td>/&lt;string&gt;</td>
<td>Search for 'string' from the current line</td>
</tr>
<tr>
<td>-/&lt;string&gt;</td>
<td>Search backwards for 'string'</td>
</tr>
<tr>
<td>all /&lt;string&gt;/</td>
<td>Show all occurrences of 'string' and hide other lines</td>
</tr>
<tr>
<td>bottom</td>
<td>Move to the bottom of the file</td>
</tr>
<tr>
<td>top</td>
<td>Move to the top of the file</td>
</tr>
<tr>
<td>down &lt;n&gt;</td>
<td>Move down 'n' lines</td>
</tr>
<tr>
<td>up &lt;n&gt;</td>
<td>Move up 'n' lines</td>
</tr>
<tr>
<td>file</td>
<td>Save the current file and exit XEDIT</td>
</tr>
<tr>
<td>ffile</td>
<td>Save the current file and exit but don't warn of overwrite</td>
</tr>
<tr>
<td>save</td>
<td>Save the current file but don't exit</td>
</tr>
<tr>
<td>quit</td>
<td>Exit XEDIT if no changes have been made</td>
</tr>
<tr>
<td>qquit</td>
<td>Exit XEDIT even if changes have not been saved</td>
</tr>
<tr>
<td>left &lt;n&gt;</td>
<td>Shift 'n' characters to the left</td>
</tr>
<tr>
<td>right &lt;n&gt;</td>
<td>Shift 'n' characters to the right</td>
</tr>
<tr>
<td>get &lt;file&gt;</td>
<td>Copy file and insert past the current line</td>
</tr>
<tr>
<td>:&lt;n&gt;</td>
<td>Move to line 'n'</td>
</tr>
<tr>
<td>?</td>
<td>Display last command</td>
</tr>
<tr>
<td>=</td>
<td>Execute last command</td>
</tr>
<tr>
<td>x &lt;file&gt;</td>
<td>Edit 'file' and put it into the XEDIT &quot;ring&quot;</td>
</tr>
<tr>
<td>x</td>
<td>Move to the next file in the ring</td>
</tr>
</tbody>
</table>

**Prefix Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>Add one line</td>
</tr>
<tr>
<td>a&lt;n&gt;</td>
<td>Add 'n' lines</td>
</tr>
<tr>
<td>c</td>
<td>Copies one line</td>
</tr>
<tr>
<td>cc</td>
<td>Copies a block of lines</td>
</tr>
<tr>
<td>d</td>
<td>Deletes one line</td>
</tr>
<tr>
<td>dd</td>
<td>Deletes a block of lines</td>
</tr>
<tr>
<td>f</td>
<td>Line after which a copy (c) or a move (m) is to be inserted</td>
</tr>
<tr>
<td>p</td>
<td>Line before which a copy (c) or a move (m) is to be inserted</td>
</tr>
<tr>
<td>i</td>
<td>Insert a line</td>
</tr>
<tr>
<td>i&lt;n&gt;</td>
<td>Insert 'n' lines</td>
</tr>
<tr>
<td>m</td>
<td>Move one line</td>
</tr>
<tr>
<td>mm</td>
<td>Move a block of lines</td>
</tr>
<tr>
<td>&quot;</td>
<td>Replicate a line</td>
</tr>
<tr>
<td>&quot;&lt;n&gt;</td>
<td>Replicate a line 'n' times</td>
</tr>
<tr>
<td>&quot;&quot;</td>
<td>Replicate a block of lines</td>
</tr>
</tbody>
</table>

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)
jkil; cursor movement
A add text at the end of the line
<nn>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
:q! quit and discard changes
:q save (write & quit)
:<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
Appendix B. Source code

This section lists source code associated with this book. The following sections are included:

- Appendix B.1, “Obtaining and using the Web material” on page 219
- Appendix B.2, “z/VM REXX EXECs and XEDIT macros” on page 219
- Appendix B.3, “Linux code” on page 223

B.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-S11.tgz

Download the tar file to your NFS server and use it as is described in section 7.5.1, “Copying files to the cloner” on page 107. After untarring the file, you will have a directory named virt-cookbook-S10SP2. Under that directory are the following files and directory:

- README.txt  The main README file
- clone.sh     The script to clone the golden image to a target user ID
- vm/          A directory containing files used on z/VM

B.2 z/VM REXX EXECs and XEDIT macros

This section lists three z/VM REXX EXECs, one XEDIT macro and a sample parameter file.

B.2.1 The CPFORMAT EXEC

Following is the code for the EXEC that formats multiple disks using CPFMTXA (described in section 4.6.1, “Formatting the paging volumes” on page 47):
B.2.2 The CHPW610 XEDIT macro

Following is the code for the XEDIT macro that changes all passwords in the z/VM 5.4 USER DIRECT file:

```almond
/*+------------------------------------------------------------------*/
/* CHPW610 XEDIT - change all passwords in z/VM 6.1 USER DIRECT file */
/*+------------------------------------------------------------------*/

/*+------------------------------------------------------------------*/
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 HEREOUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES
-------------------------------------------------------------------*/

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 curline on 3'
'command set serial off'
'command set scale off'
'command set case m i'
'command set pre off'
'command set v 1 80'
'command top'

/* change user ID passwords */
'command c/USER MAINT MAINT/USER MAINT' newPass'/**'
'command c/USER AVSMV AVSMV/USER AVSMV' newPass'/**'
'command c/USER TSAFVM TSAFVM/USER TSAFVM' newPass'/**'
'command c/USER GCS GCS/USER GCS' newPass'/**'
'command c/USER GCSX A GCSX/USER GCSX' newPass'/**'
'command c/USER AUDITOR AUDITOR/USER AUDITOR' newPass'/**'
'command c/USER AUTOLOG1 AUTOLOG1/USER AUTOLOG1' newPass'/**'
'command c/USER AUTOLOG2 AUTOLOG2/USER AUTOLOG2' newPass'/**'
'command c/USER BLDCMS BLDCMS/USER BLDCMS' newPass'/**'
'command c/USER BLDNUC BLDNUC/USER BLDNUC' newPass'/**'
'command c/USER BLDRAF BLDRAF/USER BLDRAF' newPass'/**'
'command c/USER BLDSERB BLDSERB/USER BLDSERB' newPass'/**'
```

220 The Virtualization Cookbook for SLES 11
'command c/USER CMS1 CMS1/USER CMS1' newPass'/*'
'command c/USER CMSBATCH CMSBATCH/USER CMSBATCH' newPass'/*'
'command c/USER DISKACNT DISKACNT/USER DISKACNT' newPass'/*'
'command c/USER EREP EREP/USER EREP' newPass'/*'
'command c/USER IBMUSER IBMUSER/USER IBMUSER' newPass'/*'
'command c/USER LGLOPR LGLOPR/USER LGLOPR' newPass'/*'
'command c/USER MIGMAINT MIGMAINT/USER MIGMAINT' newPass'/*'
'command c/USER MONWRITE MONWRITE/USER MONWRITE' newPass'/*'
'command c/USER OPI OPI/USER OPI' newPass'/*'
'command c/USER OPERATNS OPERATNS/USER OPERATNS' newPass'/*'
'command c/USER OPERATOR OPERATOR/USER OPERATOR' newPass'/*'
'command c/USER OPERSYMP OPERSYMP/USER OPERSYMP' newPass'/*'
'command c/USER SYSADMIN SYSADMIN/USER SYSADMIN' newPass'/*'
'command c/USER SYSDUMP1 SYSDUMP1/USER SYSDUMP1' newPass'/*'
'command c/USER SYSMARK SYSMARK/USER SYSMARK' newPass'/*'
'command c/USER SYMON SYMON/USER SYMON' newPass'/*'
'command c/USER VMMADM/VMMADM/USER VMMADM' newPass'/*'
'command c/USER VMRSVM USER VMRSVM/USER VMRSVM' newPass'/*'
'command c/USER VMSERVR VMSERVR/USER VMSERVR' newPass'/*'
'command c/USER VMSERVU VMSERVU/USER VMSERVU' newPass'/*'
'command c/USER VMUTIL VMUTIL/USER VMUTIL' newPass'/*'
'command c/USER ZVMMAPLX MAINT/USER ZVMMAPLX' newPass'/*'
'command c/USER 5684042J 5684042J/USER 5684042J' newPass'/*'
'command c/USER 4OSASF40 4OSASF40/USER 4OSASF40' newPass'/*'
'command c/USER OSADMIN1 OSADMIN1/USER OSADMIN1' newPass'/*'
'command c/USER OSADMIN2 OSADMIN2/USER OSADMIN2' newPass'/*'
'command c/USER OSADMIN3 OSADMIN3/USER OSADMIN3' newPass'/*'
'command c/USER OSAMAINT OSAMAINT/USER OSAMAINT' newPass'/*'
'command c/USER OSASF OSASF/USER OSASF' newPass'/*'
'command c/USER 6VMRSC10 6VMRSC10/USER 6VMRSC10' newPass'/*'
'command c/USER RSCS RSCS/USER RSCS' newPass'/*'
'command c/USER RSCSAUTH RSCSAUTH/USER RSCSAUTH' newPass'/*'
'command c/USER RSCSDNS RSCSDNS/USER RSCSDNS' newPass'/*'
'command c/USER XCHANGE XCHANGE/USER XCHANGE' newPass'/*'
'command c/USER 6VMTCP10 6VMTCP10/USER 6VMTCP10' newPass'/*'
'command c/USER TCPIP TCPIP/USER TCPIP' newPass'/*'
'command c/USER TCMPMAINT TCMPMAINT/USER TCMPMAINT' newPass'/*'
'command c/USER ADMSERV ADMSERV/USER ADMSERV' newPass'/*'
'command c/USER DHCPD DHCPD/USER DHCPD' newPass'/*'
'command c/USER DTCSW1 DTCSW1/USER DTCSW1' newPass'/*'
'command c/USER DTCSW2 DTCSW2/USER DTCSW2' newPass'/*'
'command c/USER FTPSERVER FTPSERVER/USER FTPSERVER' newPass'/*'
'command c/USER IMAP IMAP/USER IMAP' newPass'/*'
'command c/USER IMAPAUTH IMAPAUTH/USER IMAPAUTH' newPass'/*'
'command c/USER LDAPSRV LDAPSRV/USER LDAPSRV' newPass'/*'
'command c/USER LPSERVE LPSERVE/USER LPSERVE' newPass'/*'
'command c/USER MPROUTE MPROUTE/USER MPROUTE' newPass'/*'
'command c/USER NAMESRV NAMESRV/USER NAMESRV' newPass'/*'
'command c/USER NDBPMGR NDBPMGR/USER NDBPMGR' newPass'/*'
'command c/USER NDBSRV01 NDBSRV01/USER NDBSRV01' newPass'/*'
'command c/USER PORTMAP PORTMAP/USER PORTMAP' newPass'/*'
'command c/USER REXEC REXEC/USER REXEC' newPass'/*'
'* command c/USER SMTP SMTP/USER SMTP' newPass'/*
* command c/USER SNALNKA SNALNKA/USER SNALNKA' newPass'/*
* command c/USER SNMPD SNMPD/USER SNMPD' newPass'/*
* command c/USER SNMPQE SNMPQE/USER SNMPQE' newPass'/*
* command c/USER SNMPSUBA SNMPSUBA/USER SNMPSUBA' newPass'/*
* command c/USER SSLSERV SSLSERV/USER SSLSERV' newPass'/*
* command c/USER TFTPD TFTPD/USER TFTPD' newPass'/*
* command c/USER VMKERB VMKERB/USER VMKERB' newPass'/*
* command c/USER VMNFS VMNFS/USER VMNFS' newPass'/*
* command c/USER X25IPI X25IPI/USER X25IPI' newPass'/*
* command c/USER 6VMDIRIO 6VMDIRIO/USER 6VMDIRIO' newPass'/*
* command c/USER 6VMRAC10 6VMRAC10/USER 6VMRAC10' newPass'/*
* command c/USER RACFSMF RACFSMF/USER RACFSMF' newPass'/*
* command c/USER RACFV RACFV/USER RACFV' newPass'/*
* command c/USER RACMAINT RACMAINT/USER RACMAINT' newPass'/*
* command c/USER 6VMPTK10 6VMPTK10/USER 6VMPTK10' newPass'/*
* command c/USER PERFSVM PERFSVM/USER PERFSVM' newPass'/*
* command c/USER CBDIOSPB CBDIOSPB/USER CBDIOSPB' newPass'/*
* command c/USER GSKADMIN GSKADMIN/USER GSKADMIN' newPass'/*
* command c/USER LNIXMAINT LNIXMAINT/USER LNIXMAINT' newPass'/*

/* change mindisk passwords */
* command c/ALL WRITE MULTIPLE/ALL' newPass newPass'/*
* command c/RADMSERV WADMSERV MADMserv/’newPass newPass newPass'/*
* command c/RAUDITOR WAUDITOR MAUDITOR/’newPass newPass newPass'/*
* command c/RCONTROL WCONTROL/'newPass newPass newPass'/*
* command c/RLOG1 WLOG1/’newPass newPass'/*
* command c/RLOG2 WLOG2/'newPass newPass'/*
* command c/RLOG3 WLOG3/’newPass newPass'/*
* command c/RLOG4 WLOG4/'newPass newPass'/*
* command c/RLOG5 WLOG5/’newPass newPass'/*
* command c/RLOG6 WLOG6/'newPass newPass'/*
* command c/RLOG7 WLOG7/’newPass newPass'/*
* command c/RLOG8 WLOG8/'newPass newPass'/*
* command c/RLOG9 WLOG9/’newPass newPass'/*
* command c/RLOG10 WLOG10/’newPass newPass'/*
* command c/RLOG11 WLOG11/’newPass newPass'/*
* command c/RLOG12 WLOG12/’newPass newPass'/*
* command c/RLOG13 WLOG13/’newPass newPass'/*
* command c/RLOG14 WLOG14/’newPass newPass'/*
* command c/RLOG15 WLOG15/’newPass newPass'/*
* command c/RLOG16 WLOG16/’newPass newPass'/*
* command c/RLOG17 WLOG17/’newPass newPass'/*
* command c/RLOG18 WLOG18/’newPass newPass'/*
* command c/RLOG19 WLOG19/’newPass newPass'/*
* command c/RLOG20 WLOG20/’newPass newPass'/*
* command c/RLOG21 WLOG21/’newPass newPass'/*
* command c/RLOG22 WLOG22/’newPass newPass'/*
* command c/RLOG23 WLOG23/’newPass newPass'/*
* command c/RLOG24 WLOG24/’newPass newPass'/*
* command c/RLOG25 WLOG25/’newPass newPass'/*
* command c/RLOG26 WLOG26/’newPass newPass'/*
* command c/RLOG27 WLOG27/’newPass newPass'/*
* command c/RLOG28 WLOG28/’newPass newPass'/*
* command c/RLOG29 WLOG29/’newPass newPass'/*
* command c/RLOG30 WLOG30/’newPass newPass'/*
B.2.3  PROFILE EXEC for Linux user IDs

This section lists the code for the PROFILE EXEC that is shared among Linux user IDs from the LNXMAINT 192 disk.

/* PROFILE EXEC for Linux virtual servers */
'CP SET RUN ON'
'CP SET PF11 RETRIEVE FORWARD'
'CP SET PF12 RETRIEVE'
'ACC 592 C'
'SWAPGEN 300 524288' /* create a 256M VDISK disk swap space */
'SWAPGEN 301 1048576' /* create a 512M VDISK disk swap space */
'PIPE CP QUERY userid() | var user'
parse value user with id . dsc .
if (dsc = 'DSC') then /* user is disconnected */
  'CP IPL 100'
else /* user is interactive -> prompt */
do
  say 'Do you want to IPL Linux from minidisk 100? y/n'
  parse upper pull answer .
  if (answer = 'Y') then 'CP IPL 100'
end /* else */

B.3  Linux code

This section lists the code for the /sbin/clone.sh script that clones from the golden Linux image to a target virtual machine.

#!/bin/sh
#
# 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 11"
# on the Web at: http://www.linuxvm.org/present
#
#--------------------------------------------------------------------------------
# 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

function help()
# give help
#
{ echo "Usage: clone [options] from <sourceID> to <targetID>" echo "" echo " Clone Linux from sourceID 100 and 101 minidisks to targetID" echo " options:" echo " -v or --verbose: verbose" echo "" echo "Example: clone.sh from s11gold to linux01" exit 1 }

function processArguments()
# Parse command line arguments
# Args: The arguments passed in to the script
#
{ verbose="off" sourceID="none" targetID="none" while ( (( "$#" )) ); do case $1 in -v|--verbose) verbose="on" ;; from) shift sourceID=`echo $1 | tr '[a-z]' '[A-Z]'` # fold source ID to upper case ;; to) shift targetID=`echo $1 | tr '[a-z]' '[A-Z]'` # fold target ID to upper case ;; esac shift done if [ $sourceID = "none" ]; then # source user ID was not passed echo "Error: Source Linux user ID not supplied" help fi if [ $targetID = "none" ]; then # target user ID was not passed echo "Error: Target Linux user ID not supplied" help fi }

function CPcmd()
# echo a CP command and invoke it via the vmcp module/command
# Arg1-n: the command to issue
# Return: the command's return code

{ echo "Invoking CP command: $@
# parse output to get return code: awk -F# splits line at '#' with rc at end
output=`vmcp $@ 2>&1`
echo "$output"
retVal=0
retVal=`echo $output | grep "Error: non-zero CP response" | awk -F# '{print $2}'`
return $retVal
}

function checkID()
# Verify user ID exists and is logged off
# Arg 1: The user ID to check

{ userID=$1
echo "Checking that $userID exists and is not logged on ..."
CPcmd QUERY $userID
rc=$?
case $rc in
  0) # user ID is logged on or disconnected
    echo "$userID user ID must be logged off"
    exit 2
  ;;
  3) # user ID does not exist
    echo "$userID user ID does not exist"
    exit 3
  ;;
  45) # user ID is logged off - this is correct
    ;;
  *) # unexpected
    echo "Return code of $rc unexpected from QUERY $userID"
    echo "$userID user ID must exist and be logged off"
    exit 4
esac
}

function prepareIPaddr()
# Set the variable "newIPaddr" by adding a backslash before any "."s
# Arg 1: The IP address to be modified

{ newIPaddr=`echo $1 | sed -e 's:\.:\\.:g'`
}

function prepareVaddr()
# Prepare an address by folding to lower case and prepending leading zeros
# to make it 4 digits
# Arg 1: The vaddr to be modified
# Return:
# The new value is written to the global variable newVaddr

{
newVaddr=`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 $leadingZeros ]; do
    newVaddr="0$newVaddr"
    i=$[i+1]
done

function copyDisk()
# Use FLASHCOPY to copy a disk, if it fails, fall back to dasdfmt then dd
# Arg 1: Source vaddr
# Arg 2: Target vaddr
{
    source=$1
    target=$2
    echo ""
    echo "FLASHCOPYing $source to $target ..."
    CPcmd FLASHCOPY $source 0 end to $target 0 end
    if [ $? != 0 ]; then
        echo "FLASHCOPY failed, falling back to dasdfmt and dd ..."
        chccwdev -e $source
        if [ $? != 0 ]; then exit 7; fi
        chccwdev -e $target
        if [ $? != 0 ]; then exit 8; fi
        sleep 1
        srcDev=/dev/$(egrep ^0.0.$source /proc/dasd/devices | awk '{ print $7 }')
        if [ "$?" != 0 ]; then exit 5; fi
        tgtDev=/dev/$(egrep ^0.0.$target /proc/dasd/devices | awk '{ print $7 }')
        if [ "$?" != 0 ]; then exit 6; fi
        echo "dasdfmt-ing $tgtDev ..."
        dasdfmt -y -b 4096 -f $tgtDev
        if [ "$?" != 0 ]; then exit 9; fi
        echo "dd-ing $srcDev to $tgtDev ..."
        dd bs=4096 if=$srcDev of=$tgtDev
        if [ "$?" != 0 ]; then exit 10; fi
        echo "disabling and re-enabling $target ..."
        sync
        chccwdev -d $target
        if [ $? != 0 ]; then exit 11; fi
        chccwdev -e $target
        if [ $? != 0 ]; then exit 12; fi
    fi
}

function getNetworkInfo()
# Bring 191 minidisk online to check for two parameter files
{
    # recycle 191 to pick up latest changes
    chccwdev -d 191
    chccwdev -e 191
    rc=$?
    if [ $rc != 0 ]; then # unable to enable 191 disk
        echo "unable to enable 191, rc from chccwdev = $rc"
        exit 13
    fi
sleep 1 # TODO: get the command that waits until the disk is enabled.
CMSdisk=`lsdasd | grep 0191 | awk '{ print $3 }'`
.cmsfslst -d /dev/$CMSdisk | grep -i $sourceID | grep PARM-S11
rc=$?
if [ $rc != 0 ]; then
    echo "Error: $sourceID PARM-S11 not found on 191 minidisk. Exiting"
    exit 14
fi
.cmsfslst -d /dev/$CMSdisk | grep -i $targetID | grep PARM-S11
rc=$?
if [ $rc != 0 ]; then
    echo "Error: $targetID PARM-S11 not found on 191 minidisk. Exiting"
    exit 15
fi

# get information about target
export local $(/dev/$CMSdisk $targetID.PARM-S11)
targetHostname=$Hostname
targetIP=$HostIP
targetDNS=$Nameserver
targetGW=$Gateway
targetMask=$Netmask
targetBroadcast=$Broadcast
prepareVaddr $ReadChannel
targetReaddev=$newVaddr
prepareVaddr $WriteChannel
targetWritedev=$newVaddr
prepareVaddr $DataChannel
targetDatadev=$newVaddr

# get information about source
export local $(/dev/$CMSdisk $sourceID.PARM-S11)
sourceHostname=$Hostname
prepareIPaddr $HostIP
sourceIP=$newIPaddr
prepareIPaddr $Nameserver
sourceDNS=$newIPaddr
prepareIPaddr $Gateway
sourceGW=$newIPaddr
prepareIPaddr $Netmask
sourceMask=$newIPaddr
prepareIPaddr $Broadcast
sourceBroadcast=$newIPaddr
prepareVaddr $ReadChannel
sourceReaddev=$newVaddr
prepareVaddr $WriteChannel
sourceWritedev=$newVaddr
prepareVaddr $DataChannel
sourceDatadev=$newVaddr

# --- askAreYouSure() # Ask "Are you sure?" - if not, then exit
function askAreYouSure()
{
    echo "WARNING!!: Mindisks 100 and 101 will be copied to $targetID"
    echo "New host name will be: $targetHostname"
    echo "New IP address will be: $targetIP"
    read 
    if [ $? -ne 0 ]; then
        echo "Exiting"
        exit 16
    fi
}

# --- main script ---

# --- main script ---
echo "Other network data is retrieved from $targetID PARM-S11 on 191 disk"
echo -n "Are you sure you want to overwrite these disks (y/n): "
read ans
if [ $ans != "y" ]; then
echo "Aborting clone per user input"
exit 16
fi

#+--------------------------------------------------------------------------+
function copySystem()
  # For each of two mindisks 100 and 101:
  # -} Link disk
  # -} Enable disk
  # -} Copy disk
  +--------------------------------------------------------------------------+
  {
    echo "Linking source and target 100 disks ..."
    CPcmd detach 1100
    CPcmd link $sourceID 100 1100 rr
    if [ $? != 0 ]; then exit 17; fi
    CPcmd detach 2100
    CPcmd link $targetID 100 2100 mr
    if [ $? != 0 ]; then exit 18; fi
    echo "Copying 100 disks ..."
    copyDisk 1100 2100
    CPcmd det 1100
    # don't detach 2100 yet because it has to be modified

    echo "Linking source and target 101 disks ..."
    CPcmd detach 1101
    CPcmd link $sourceID 101 1101 rr
    if [ $? != 0 ]; then exit 19; fi
    CPcmd detach 2101
    CPcmd link $targetID 101 2101 mr
    if [ $? != 0 ]; then exit 20; fi
    echo "Copying 101 disks ..."
    copyDisk 1101 2101
    CPcmd det 1101
    CPcmd det 2101
  }

#+--------------------------------------------------------------------------+
function modifyClone()
  # Mount newly copied system and modify networking info
  +--------------------------------------------------------------------------+
  {
    targetVaddr=2100
    targetDir="/mnt/targetLinux"         # directory of target Linux system
    echo "Mounting newly cloned image over $targetDir ..."
    if [ ! -d $targetDir ]; then
      mkdir $targetDir
      rc=$?
      if [ $rc != 0 ]; then
        echo "Error: mkdir $targetDir failed with $rc"
        exit 21
      fi
    fi
    # recycle target mount point
    # chccwdev -d $targetVaddr
chccwdev -e $targetVaddr
rc=$?
if [ $rc != 0 ]; then
   echo "Fatal error: chccwdev -e $targetVaddr failed with $rc"
   CPcmd DET $targetVaddr
   exit 22
fi
sleep 1
tDev=/dev/$(/egrep ^0.0.$targetVaddr /proc/dasd/devices | awk '{ print $7 }')1
if [ "$?" != 0 ]; then exit 23; fi

echo "Mounting $tDev over $targetDir ..."
mount $tDev $targetDir
rc=$?
if [ $rc != 0 ]; then
   echo "Error: 'mount $tDev $targetDir' failed with $rc"
   lsdasd
   CPcmd DET $targetVaddr
   exit 24
fi

echo "Modifying cloned image under $targetDir ..."

sed --in-place -e "s/$sourceHostname/$tagetHostname/g" $targetDir/etc/HOSTNAME
sed --in-place -e "s/$sourceIP/$targetIP/g" -e "s/${sourceHostname%%.*}/${tagetHostname%%.*}/g" $targetDir/etc/hosts
sed --in-place -e "s/$sourceGW/$targetGW/g" $targetDir/etc/sysconfig/network/routes
sed --in-place -e "s/$sourceBroadcast/$targetBroadcast/g" $targetDir/etc/sysconfig/network/ifcfg-eth0
sed --in-place -e "s/$sourceDNS/$targetDNS/g" $targetDir/etc/resolv.conf

# Delete SSH keys - sshd will recreate them at first boot
echo "Removing SSH keys"
rm $targetDir/etc/ssh/ssh_host*

# Remove any old entry, then copy clone's public key to known_hosts file
echo "Removing $targetIP from known_hosts file"
cd /root/.ssh
grep -v $targetIP known_hosts > known_hosts.temp
mv known_hosts.temp known_hosts

# clean up
sleep 1
sync # sync disks
umount $targetDir
sync
chccwdev -d $targetVaddr
CPcmd DETACH $targetVaddr
return 0

} # main()

processArguments $0
if [ $verbose = "on" ]; then set -vx; fi # turn on debug
checkID $sourceID # user ID must exist and be logged off
checkID $targetID # user ID must exist and be logged off
getNetworkInfo # get info from parm files
askAreYouSure # confirm disks will be overwritten
copySystem # copy source disks to target
modifyClone # modify newly copied system
CPcmd XAUTOLOG $targetID                  # bring new clone to life
if [ $verbose = "on" ]; then set +vx; fi  # turn off debug
echo "Successfully cloned $sourceID to $targetID"
echo "You should be able to ping $targetIP within one minute"
exit 0
Index

Symbols
$ALLOC$, z/VM user ID  54, 91, 118, 147, 157
/etc/fstab  186
/etc/samba/smb.conf file  172
/etc/samba/smbpasswd file  172
/proc/sys/appldata/ directory  212
/usr/share/doc/ directory  172

Numerics
3270 emulators  25, 40
520RES  35

A
Aduva’s Onstage  3
Albert Einstein  3
Alt-Tab sequence  28
Apache  163
Apache resources  164
appldata_mem module  212
appldata_net_sum module  212
appldata_os module  212
appletviewer command  177
application development resources  178
application development server  175
APPLMON option  211
ATTACH command  33
Attachmate Extra!  25
AUTOLOG1, z/VM user ID  216

B
Bill of materials  7

C
C/C++ compiler  176
C/C++ development  176
CHANGEPW XEDIT  59
CHANGEPW XEDIT macro  60
Changes to current book  xi
chapters
    in the book  ix
chkconfig command  86, 110, 139, 173
choices made
    in the book  3
CHPW52 XEDIT  59
    source code  220
clone.sh script  154, 171, 175
cloning
    Six more virtual servers  156
cmsfs package  112
cmsfscat command  113
cmsfslst command  113
Common UNIX Printing System (CUPS)  171

Computer Aided Design (CAD)  176
Computer Associates
    VM Direct  3
    VM Monitor  197
    VM Secure  59
Control Program  3
Controller  ix
    applying service  111
    Configuration  107
    Copying files to  107
    Installing  95
    Turn off unneeded services  110
    Verify your changes  115
conventions
    command retrieval  10
    in the book  9
    naming backup files  9
    volume labeling  9
CP  3
CPACCESS command  43
CPFMTXA command  47
CPFORMAT EXEC  47
    using it  48
CPSYNTAX  51
CPSYNTAX command  43
CUPS  175

D
DASD
    3390  4
    activating in a SLES10 install  123
    ECKD  4
    formatting in a SLES10 install  123
    labelling convention  9
dasdfmt command
    using in a for loop  182
dd command  154
desktop  19
df command  144
DIRECTXA command  54, 60, 91, 118, 157
DirMaint  3
Disconnect Timeout  41
Disconnect_Timeout  42
DISKMAP command  53
distributions
    31-bit and 64-bit  x

E
egrep command  171
ext3 file system
    creating  186

© Copyright IBM Corp. 2010. All rights reserved.
The Virtualization Cookbook for SLES 11

F
fdasd command 183
Features statement 42
FILELIST command 93
FLASHCOPY command 90, 154
font conventions xi
FORMAT command 55
formatting DASD fast 47
FTP
  site fix 80 subcommand 93
FTP server 45

G
gcc 176
gdb 177
GNU compiler collection (gcc) 176
GNU debugger (gdb) 177
Golden image
  Configuration 135
  Verifying changes 144

H
hardware resources 7
HELP command 200
HMC 28
  CD-ROM/DVD 30
  Load icon 35
  Operating System Messages 46
  Single Object Operations 29
  SYSG 35
  System Console (SYSC) 46
Hummingbird Host Explorer 25
hypervisor 3

I
IBM Personal Communications 25
IBM RACF 59
IBM ShopzSeries 69
IBM Tivoli Provisioning Manager 3
IBM z/VM Performance Toolkit 197
  monitor subcommand 208
INDICATE command 197
INDICATE LOAD command 198
INDICATE QUEUES EXP command 199
infrastructure
  block diagram 4
design 4
initial RAMdisk 92
install source 108
INSTPLAN 32
INSTVM EXEC 37
Integrated 3270 Console 28, 32
IPL CMS 37
IPWIZARD command 38

J
jar command 178
Java 177
Java development
tool 177
Java Runtime Environment (JRE) 177
jdb command 178

K
kernel 92

L
LAMP 175
LDAP client 169
LDAP server
  manual install 165
Idapsearch command 169
Lightweight Directory Access Protocol (LDAP) 164
Linux
  manual installation 4
  monitoring 197, 211
  monitoring from the kernel 211
Linux distribution 164, 175–176
Linux PC 83
Linux RMF PM Data Gatherer (rmfpms) 210
LINUX01 146, 163
LNXDFTLT profile 90
LNXMAINT 52
  copying files to 56
Load icon 35
logical partition ix, 1
logical volume 91
  adding 183
LPAR
  requirements 7
Ismod command 212
lvcreate command 185
LVM
  block diagram 184
  physical extents 185

M
MAINT 2CC 216
MAINT CF1 216
make command 177
Map Network Drive 173
Master image ix
  Installation 120
  Remove unnecessary RPMs 138
  Turn off unneeded services 138
MDISK statement 53
Minidisk cache (MDC) 198
Minidisk passwords 59
mke2fs command 186
modprobe command 212
monitor stream support 211
monitoring Linux 210–211
mount command 106
MySQL 175
Network File System (NFS) 83
networking resources 8
NFS server
configuration on Linux PC 86
nmbd daemon 173

OpenLDAP
pointing virtual server to LDAP server 168
overlaps
searching for 53

paging space 47
paging volumes 47
parameter file 92
  copying 157
  copying for new user ID 148
passwords
planning 12
Performance Toolkit 201–202
PERFSVM user ID 203
Perl 176
philosophy of the book 3
PHP 175–176
physical extents 185
planning 7
Printing
configuration 175
PROFILE EXEC
  AUTOLOG1's 57, 91, 119, 147, 216
  LNXMAINT's 55
  PERFSVM's 208
PROFILE LNXDFLT 90
PROFILE TCPPIP 44
Programming 67
PTF 69
PUT2PROD command 37
PuTTY 19
  customizing 19
  larger screen 22
pvcreate command 184
pvdisplay command 184
Python 175–176

QUERY VSWITCH ACCESSLIST command 159
QUERY VSWITCH command 92
QUERY XSTOR command 200
Quick3270 25

RAMdisk 92
rcapache2 command 164
rcnfsserver command 116
rcnmb command 173
Reader, z/VM's 92
reboot command 115, 143
Recommended 67
Red Hat RHEL
  on the PC NFS server 87
RELEASE command 43
rmfpms 210
rmfpms command 211
root password 133

Samba 171
  configuration 172
  integrating with LDAP 173
saved segments
  rebuilding 65
SCSI/FCP disks 4
SET SIGNAL command 57
SET SRM command 57
SET VSWITCH GRANT command 92, 120, 148, 158
share /usr/ file system 3
SHUTDOWN command 46
SHUTDOWN REIPL command 52, 58
Single Object Operations 29
site fix 80, FTP subcommand 93
SLES 10 SP2
  install source 108
SLES10 89
  activating DASD in an install 123
  applying service 140
  bootstrap files 92
  formatting DASD in an install 123
  installation 121
  Installing and configuring 89, 117
Java 177
  parameter file 148
  partitioning DASD in an install 123
  Preparing for Online Update 138
  second part of an install 133
SLES10 EXEC 94, 121
smbd daemon 173
smbpasswd command 172
software resources 8
SSH 135
  Configuring keys 141
  key-based authentication 141
  protocol 2 19–20
Standalone Program Loader 35
Standard Development Kit (SDK) 177
The Virtualization Cookbook for SLES 11

Virtual server
application development 175
File and print 171
LDAP 164
Web server 163
virtual servers 3
virtual switch (VSWITCH) 1, 41
vmcp command 115
VNC 121
configuring on master image 135
volume group 185
VSWITCH 1, 44, 46, 158

Windows workstation 83
worksheets
examples in the book 13

X
XAUTOLOG command 44, 57, 154
XEDIT
** prefix command 90, 156
? subcommand 10
add subcommand 42
BOTTOM subcommand 42
CHANGE subcommand 60
FILE subcommand 42
PREFIX OFF subcommand 53
search subcommand (/) 42
XEDIT command 40
XEDIT PROFILE 40

Z
z/VM
adding paging volumes 47
backing up to tape 61
cons=sysg 35
conventions 9
CPFMTXA command 47
customizing shutdown 57
customizing startup 57
DIRECTXA command 54
Disconnect Timeout 41
DISKMAP command 53
FORMAT command 55
FTP server 45
HELP command 200
HIT RATIO 198
INDICATE command 197
INDICATE QUEUES EXP command 199
INSTPLAN EXEC 32
INSTVM EXEC 37
Integrated 3270 Console 28
IPWIZARD command 38
latest PTFs 76
MIGRATE rate 198
monitoring 197
PAGING rate 199
PUT2PROD command 37
QUERY ALLOC MAP command 201
QUERY CPLEVEL command 201
QUERY DASD command 201
QUERY NAMES command 200
QUERY OSA command 201
QUERY PROCESSORS command 200
QUERY STORAGE command 200
QUERY VIRTUAL ALL command 201
QUERY XSTOR command 200
reader 92
RELEASE command 43
Restoring from tape 66
security issues 59
Servicing 67
SET SIGNAL command 57
Standalone Program Loader 35
STEAL value 199
strengths 1
system name 42
TCP/IP configuration 38
TCP/IP at IPL time 43
userid() function 94
virtual switch 1
VSWITCH 1
XEDIT PROFILE 40
XSTORE rate 198
z/VM CPACCESS command 43
z/VM system
backup Copy 61
z/VM user IDs
default password 38