ibm.com/redbooks

# Installing Linux for z Systems on zPDT
## A Short Cookbook

Bill Ogden

z Systems

Redbooks

International Technical Support Organization

# Installing Linux for z Systems on zPDT: A Short Cookbook

February 2016

SG24-8330-00

**Note:** Before using this information and the product it supports, read the information in “Notices” on page v.

**First Edition (February 2016)**

This edition applies to Version 1, Release 6, of the zPDT product.

# Contents

**Notices** . . . v
Trademarks . . . vi

**IBM Redbooks promotions** . . . vii

**Preface** . . . ix
Authors . . . ix
Now you can become a published author, too! . . . ix
Comments welcome . . . x
Stay connected to IBM Redbooks . . . x

**Chapter 1. Introduction** . . . 1
1.1 Abbreviations . . . 2
1.2 The base Linux system . . . 4
1.3 Configurations used . . . 5

**Chapter 2. Base installation** . . . 7
2.1 Prerequisites . . . 8
2.2 Base installation steps . . . 8
2.2.1 Base Linux installation . . . 9
2.3 zPDT installation . . . 10
2.4 z Systems disk planning . . . 11
2.5 zPDT device map . . . 12
2.5.1 LAN interface . . . 14
2.5.2 Initial zPDT startup . . . 15
2.6 Typical operation with zPDT . . . 16

**Chapter 3. Linux for z Systems (SUSE)** . . . 17
3.1 Installing SLES 11 service pack 3 . . . 18
3.1.1 Booting the installer and installation kernel . . . 19
3.1.2 Using VNC to work with YaST2 . . . 23
3.1.3 Completing the installation . . . 25
3.1.4 Additional work . . . 27
3.1.5 Stopping the system . . . 27
3.1.6 FTP . . . 27
3.1.7 More DASD . . . 27
3.1.8 Network expansion . . . 29
3.2 SLES 12 . . . 30
3.2.1 Booting the installer and installation kernel . . . 31

**Chapter 4. Linux for z Systems (Red Hat)** . . . 37
4.1 Installing RHEL 6.7 . . . 38
4.1.1 Installation status . . . 40
4.1.2 Additional steps . . . 41
4.1.3 Shutdown . . . 41
4.1.4 Adding additional disk volumes . . . 41
4.2 Installing RHEL 7.1 . . . 42
4.2.1 System status . . . 45

**Chapter 5. General topics** . . . 47

 

5.1 Memory usage . . . . . . . . . . 48
5.2 Disk backups . . . . . . . . . . 48
5.3 zPDT updates . . . . . . . . . . 49
5.4 Base Linux updates . . . . . . . . . . 49
5.5 Disk addresses . . . . . . . . . . 50

**Related publications** . . . . . . . . . . 51
IBM Redbooks . . . . . . . . . . 51
Online resources . . . . . . . . . . 51

# Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
*IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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

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

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided “AS IS”, without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

# Trademarks

THIS PAGE INTENTIONALLY LEFT BLANK

# Preface

This IBM® Redbooks® publication describes the basic installation processes of Linux for z Systems™ on an IBM zPDT® base. It is intended for readers who are not familiar with IBM z Systems or with the zPDT product. This book assume the reader is familiar with Linux on Intel-compatible platforms.

This book provides basic introductions to necessary z Systems and zPDT topics, and proceeds in a cookbook manner. This book is not intended for readers who are already familiar with these topics.

## Authors

This book was produced in the International Technical Support Center (ITSO), Poughkeepsie. The author was Bill Ogden:

**Bill Ogden** is a retired Senior Technical Staff Member who continues to work part time with projects that he enjoys. These include working with new mainframe users and entry-level systems.

The following people contributed substantially to this project:

**Keith VanBenschoten**, IBM Poughkeepsie, is the technical leader who provided zPDT test systems, and the zPDT installation processes and tools.

**Theodore Bohizic,** IBM Poughkeepsie, an IBM Senior Technical Staff Member, is a key zPDT developer.

**Marc van der Meer,** IBM Netherlands, is a Senior IT specialist with IBM Systems Lab Services. Marc helped unravel specific details involved in installing various Linux for z Systems distributions.

## Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

**ibm.com**/redbooks/residencies.html

 

## Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  **ibm.com**/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

## Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

1

# Introduction

This chapter covers the abbreviations and products, Linux system, and configuration that are used in the examples in this book. It contains the following sections:

- Abbreviations
- The base Linux system
- Configurations used

## 1.1 Abbreviations

This book discusses several products, each of which is known by an abbreviated name:

- IBM z Systems personal development tool (zPDT).
- SUSE Linux Enterprise Server (SLES) 11.3 and 12 for z Systems. SLES 11.3 and 12 can be used with base hardware other than z Systems, but this book only describes the z Systems distribution unless noted otherwise.
- Red Hat Enterprise Linux (RHEL) 6 fand 7 or z Systems. RHEL 6 and 7 can be used with base hardware other than z Systems, but this book only describes the z Systems distribution unless noted otherwise.

This book also describes IBM z Systems™. The formal name has recently been changed from System z, but you are likely to see both names used. The Linux distributions might have "S/390®" or "System/390" in their names. This is the name of an older series of IBM machines. For purposes of this book, these are the same as z Systems.

There are other Linux distributions that are intended for IBM z Systems use, but they are not discussed in this book.

Novell (SLES) and Red Hat (RHEL) provide extensive documentation for their products. This book describes the basic installation of one of these distributions on zPDT. The initial details of such installations differ somewhat from installation on a large z Systems mainframe, and these differences are not described in the standard documentation.

As stated, this document is intended as a simple cookbook for users who are new to z Systems and zPDT. It ignores many of the finer details of Linux for z Systems, and concentrates on helping you install your first attempt at this combination. It provides concise instructions for installing Linux for z Systems on a zPDT base. zPDT provides the z Systems architecture required for Linux for z Systems.

The general overview is shown in Figure 1-1.

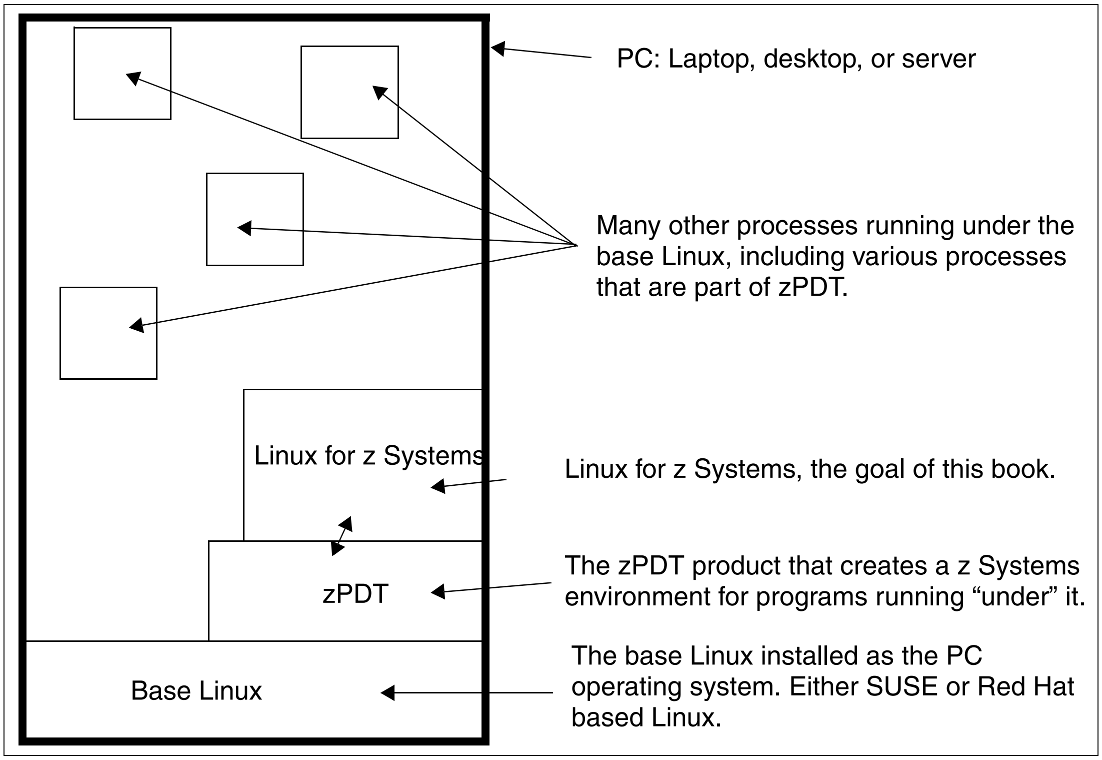


*Figure 1-1 Overview of zPDT Environment*

Linux for z Systems installed this way will not provide the performance that the base Linux distribution exhibits. The reason for installing Linux for z Systems in this manner is to provide an inexpensive base for developing and testing applications intended for Linux for z Systems. The zPDT product produces the same z Systems functions that are found in the latest z Systems, with minor exceptions.

For more information about zPDT, see *zPDT Guide and Reference*, SG24-8205. The material in this book has been extracted material from the larger publication to help new users get started with Linux for z Systems on top of a zPDT base.

The zPDT product is provided by IBM in two ways:

- A product for independent software vendors (ISVs). This product is known simply as zPDT. It also includes use of prepackaged IBM z/OS®, z/VM®, and z/VSE® systems that are not relevant to a simple Linux for z Systems installation.
- A product known as IBM Rational® Development and Test (RD&T). This product also includes usage of a slightly different prepackaged z/OS system.

The terms and conditions for obtaining these two products are different, but the functionality of the zPDT included in both products is essentially the same.

zPDT is a licensed product, and the licenses are controlled by *tokens*. A token is similar to a USB memory key (USB flash drive), and is connected to a USB port on the PC.[1] A standard token can provide licenses for 1, 2, or 3 z Systems processors (z Systems *cores*). The ISV version of zPDT provides a 1090 token, whereas the RD&T version provides a 1091 token. The 1090 and 1091 numbers are IBM machine types. For the purposes of this book, they are functionally equivalent.

[1] Token licenses can also be obtained from a remote *token server*. This is described in SG24-8205.

More advanced Linux for z Systems installations can operate under the IBM z/VM operating system. Figure 1-2 shows a example zPDT environment.

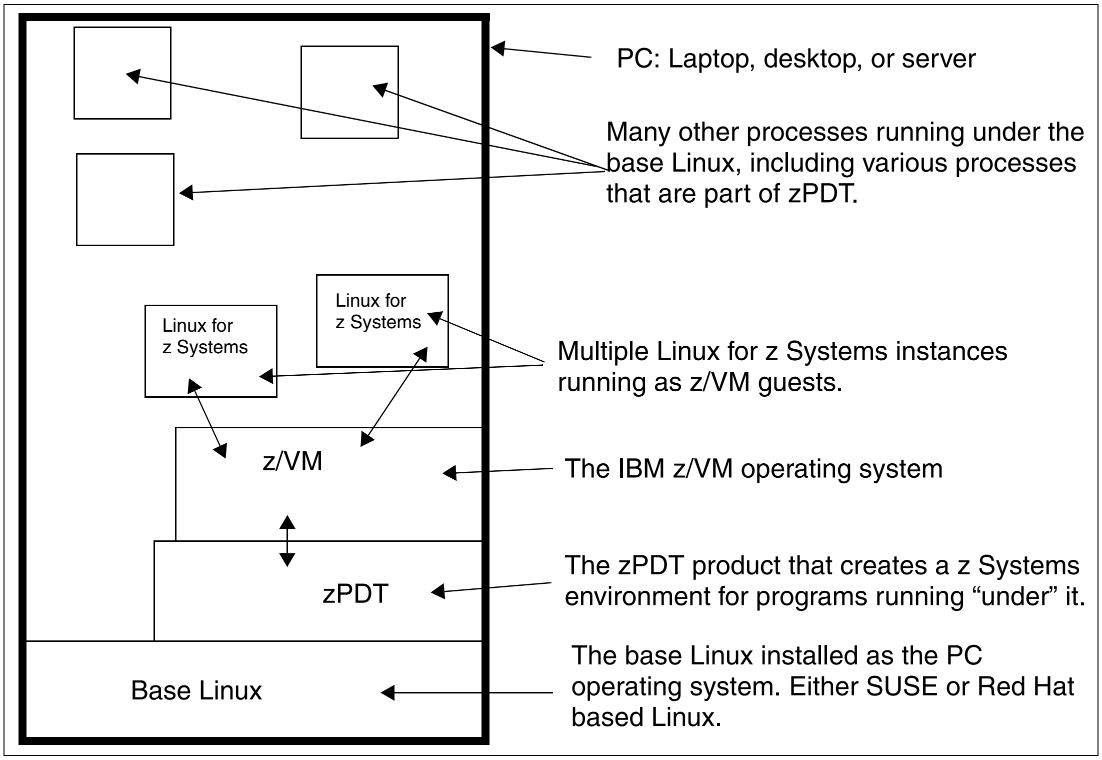


*Figure 1-2 zPDT, z/VM, and Linux for z Systems*

This book concentrates on the simpler case of a single Linux for z Systems running directly on zPDT. This is sometimes known as running in an *LPAR*. Other IBM publications discuss the more advanced option of running under z/VM.

**Restriction:** At the time of writing, there was a known problem using Fedora 22 and Fedora 23 with zPDT tokens. The problem is with the token drivers, not Fedora. If you select Fedora for the base Linux, an earlier distribution is needed. Check for a resolution with your zPDT provider if you want to use one of these distributions for your base Linux.

## 1.2 The base Linux system

All the work described here runs on a personal computer that is “Intel-compatible” running a Linux operating system. In this book, this Linux is the “base Linux” system. *Do not confuse it* with the Linux for z Systems that are the “target” of this book. The base Linux must be either Novell (SLES 12 or openSUSE) or Red Hat (RHEL 6 or Fedora) due to zPDT requirements. The choice of the base Linux operating system is independent of the choice of the target Linux for z Systems.

The specific system that was used when exploring the topics in this book was a Lenovo W520 (with 16 GB memory) running under openSUSE 11.3.

## 1.3 Configurations used

The test environment involved installing recent distributions of the two Linux for z Systems distributions that the book covers and the previous major distribution level of each. In some respects, the older versions were easier to install and use. The following distributions were installed:

- SLES release 11, service pack 3
- SLES release 12
- RHEL release 6.7
- RHEL release 7.1

Each installation started with an .iso file that was downloaded from an authorized source.

All of the examples in this book were performed with user ID `ibmsys1` on the base Linux system. You can use any user ID[2], but this one is used it consistently throughout the documentation.

### Quick overview of relevant files

For each of the four Linux for z Systems installations, three disk volumes were created: One volume for the target system residence (with all the normal Linux directories) and two smaller volumes that could be used for data file systems. These two smaller volumes allow exploration of the different ways they can be recognized, formatted, and used by the four different Linux for z Systems distributions. The concepts of "volumes" is explained in more detail in Chapter 2, "Base installation" on page 7. For the moment, a "volume" can be considered a file in the base Linux system. Figure 1-3 shows the files in the base Linux installation.

```
/home/ibmsys1/DVD/SLE-12-Server-DVD-s390x-GM-DVD1.iso
/home/ibmsys1/DVD/SLES-11-SP3-DVD-s390x-DVD1.iso
/home/ibmsys1/DVD/RHEL-6.7-20150702.0-Server-s390x-dvd1.iso
/home/ibmsys1/DVD/RHEL-7.1-20150219.1-Server-s390x-dvd1.iso

/z/SLES11A     (10 GB)   z Systems address 2A0
/z/SLES11B     (2 GB)    z Systems address 2A1
/z/SLES11C     (2 GB)    z Systems address 2A2
/z/SLES12A     (10 GB)   z Systems address 2B0
/z/SLES12B     (2 GB)    z Systems address 2B1
/z/SLES12C     (2 GB)    z Systems address 2B2
/z/RHEL6A      (10 GB)   z Systems address 2C0
/z/RHEL6B      (2 GB)    z Systems address 2C1
/z/RHEL6C      (2 GB)    z Systems address 2C2
/z/RHEL7A      (10 GB)   z Systems address 2D0
/z/RHEL7B      (2 GB)    z Systems address 2D1
/z/RHEL7C      (2 GB)    z Systems address 2D2

/home/ibmsys1/ISO            (an empty directory used as a mount point later)
```

*Figure 1-3 Relevant files on base Linux*

The example DVD directory and the /z file system were created on the base Linux system. The names and locations used are typical. The Linux file names for the target volumes are arbitrary choices. The four iso files in the figure are the downloads used to install Linux for z

[2] The base Linux user ID selected for using zPDT must not be more than eight characters long.

Systems. You would probably need only one of these when installing a single Linux for z Systems, and you would need only one set of the `/z/xxxxx` files.

The z Systems address Figure 1-3 on page 5 is explained in Chapter 2, “Base installation” on page 7. You probably want to install on one of these four target Linux for z Systems distributions. The examples include all four so that we can later boot whichever one we want at any time. The example uses 10 GB volumes for the target Linux residence devices. In practice, the target Linux systems used between 3 and 4 GB. Nothing was placed on the other volumes during the installations.

If you are installing one of these distributions, you do not need to create the volumes used for other distributions. The relevant files are described here to avoid any confusion later.

The ISO directory is used as a mount point for the Linux .iso distribution, as described later.

2

# Base installation

There are many ways to install a base Linux, many options for using zPDT, and many ways to configure and install Linux for z Systems. Many arbitrary choices are made in this book to simplify installation for first-time users. If you are not familiar with zPDT or with z Systems usage, initially follow the steps outlined in this document. After you are more comfortable with the components, you can customize your system in many ways.

This chapter assumes that you are familiar with Linux running on a PC. It does not go into lengthy explanations about simple PC Linux use or common Linux commands.

This chapter includes the following sections:

- Prerequisites
- Base installation steps
- zPDT installation
- z Systems disk planning
- zPDT device map
- Typical operation with zPDT

## 2.1 Prerequisites

You need the following components to get started:

- A zPDT token: Either a 1090 or 1091. You need to know how many licenses the token provides. This is part of the *model number* that is associated with the token. For example, a 1090-L03 token provides up to three licenses. The token must be *activated*, which is normally done by whoever provides the token.
- You need a PC. It must have more cores than the number of licenses in the token. This computer should have *at least* 8 GB of memory. Generally it should have *at least* a 100 GB hard disk drive, and possibly more disk space depending on your intended applications.
- You must have a base Linux distribution. zPDT requires SUSE (openSUSE or SLES) or Red Hat (Fedora or RHEL) Linux. Do not use other distributions. Generally, select a supported distribution (SLES or RHEL). The selection of a base Linux is independent of the Linux for z Systems distribution that you will use. For the examples in this book, the base Linux was openSUSE 13.1. However, any recent SLES, RHEL, openSUSE, or Fedora system can be used.
- You must have the zPDT package.
- You must decide on an initial z Systems configuration, and create a zPDT device map (*devmap*) to define this configuration.
- You must have a Linux for z Systems distribution. The examples assume these are in the format of a single .iso file. These distributions might be known as Linux for S/390[1], which is the series name for the IBM mainframes before z Systems. The specific examples deal with SLES 11 SP3, SLES 12, RHEL 6.7, and RHEL 7.1. This book assumes that you are using one of these levels or a slightly later version of these.

The following steps, explained in more detail later, are involved in installing Linux for z Systems on zPDT. The steps are usually executed in the order presented here:

1. Install the base Linux. As mentioned earlier, this must be a SUSE or Red Hat distribution.
2. Install zPDT.
3. Plan your disk usage for z Systems.
4. Create a zPDT device map.

The following general steps are discussed in later chapters:

1. IPL (the z Systems equivalent of *booting*) the initial installation programs from the DVD files.
2. Connect a VNC connection to the installation programs and complete the initial Linux for z Systems setup.
3. Connect to your Linux for z Systems with VNC (graphics) or SSH (text mode) and continue your setup and operation. This process might involve, for example, revising network connections and adding more disk volumes.

## 2.2 Base installation steps

The goal of this book is a simple installation process to help the new user get started with Linux for z Systems. This stage uses a minimum number of disks, no LVM, and an especially

[1] Or possibly even Linux for S/370, which was an older IBM mainframe series.

simple LAN configuration. Later chapters describe modest upgrades to this initial configuration.

Select a user ID for your base Linux system. You need to operate as *root* for the initial zPDT installation steps. Note carefully which steps require *root*. **Do not** use *root* to run zPDT after completing the installation steps. The examples use `ibmsys1` as the base Linux user ID for everything connected with zPDT usage, but this name is arbitrary.

## 2.2.1 Base Linux installation

The example used openSUSE 13.1 (64-bit version) as the base Linux. The PC was a 16 GB Lenovo W520 with a 150 GB hard disk. You can install whatever options you want in your base Linux. The following are needed for the example:

- 32-bit support is required for the token drivers. This support is installed automatically with openSUSE 13.1, so no special action was needed in teh example. Action might be needed on other base systems.
- You need a plan for your Linux disk layout. Segregate your (future) z Systems emulated volumes in a separate file system on your base Linux. The simple layout used in the example is as follows:
  - A partition (maybe 10-20 GB) to be mounted as root. All of the base Linux is installed here, taking around 4 GB.
  - A swap partition: The example uses 4 GB, which is probably more than is needed.
  - A partition for z Systems volumes: In the example, this partition is arbitrarily mounted at /z. The size depends on what you intend to do with z Systems applications. The partition should be at least 20 GB, but will probably be much larger.

  The advantage of a separate partition for z Systems volumes is that you can reinstall a base Linux (and format its partition, if needed) without losing the z Systems volumes.
- Install an FTP program. The example uses vsftp.
- Select a desktop manager. *Gnome* was used until it became too graphic, and then Xfce was used as a simple desktop manager. The associated editor is *leafpad*. The selection of the desktop manager for the base Linux does not affect zPDT operation.
- Create an appropriate user ID. The examples always use `ibmsys1`, but this is arbitrary. However, the user ID cannot be more than eight characters.[2] It is best to always use the same base Linux user ID for all functions related to zPDT. Do not use root for this, except as directed when installing zPDT.

After installing the base Linux, be sure to test the FTP server function. It will be needed later. Your base Linux must accept an FTP connection initiated elsewhere. You might find it best to turn off the firewall.

> **Note:** When describing Linux commands, this book uses a # (hash sign) to indicate that you are operating as *root*, and $ (dollar sign) to indicate that you are not operating as *root*. Your actual Linux command line is probably longer than this. This simple prompt indicator is used for clarity.

[2] The base Linux user ID used to start zPDT becomes the LPAR name for the emulated z Systems and LPAR names are limited to eight characters.

## 2.3 zPDT installation

zPDT is distributed as a single file. There are different files for 1090 tokens and 1091 tokens, so be certain that you obtain the correct distribution. The file name is something like `z1090-1-6.49.15.x86_64`. The 1091 version has a longer name, but is similar to this example.

The single file contains the following items:

- An sntl-sud rpm at the correct level (a driver for the zPDT token)
- A zpdt-shk-server rpm at the correct level (another token program)
- The primary z1090 (or z1091) rpm for SUSE Linux
- The primary z1090 (or z1091) rpm for Red Hat Linux
- An *installer program* that displays a license and then installs the rpms. The correct rpm (Red Hat or SUSE) is automatically selected for your base Linux system. You cannot install zPDT by directly using an **rpm** command.

As shown, the single distributed zPDT file contains two versions of zPDT: One for Red Hat and one for SUSE. This division is necessary due to different library levels in the two versions.

Proceed with zPDT installation as follows. Note the use of the # Linux prompt characters to indicate when *root* authority is required:

1. Move the installation file to a convenient directory, such as `/tmp`. If you obtained the zPDT installation file through FTP or another download method, you might already have placed it in `/tmp`.
2. Begin the installation[3]:

```
(log in as ibmsys1)
$ su                                       (change to root)
# cd /run/media/ibmsys1/ROM                (if you install from a CD)
# cp z1090-1-6-49.15.x86_64  /tmp          (use the correct name)
# cd /tmp                                  (the file is in /tmp)
```

3. Run the installer program[4]:

```
# chmod u+x z1090-1-6.49.15.x86_64  (make file executable, if not already)
# ./z1090-1-6.49.15.x86_64          (verify the exact file name first)
```

4. Scroll through the license that is displayed and reply to the question at the end. The various rpms are then installed automatically. The zPDT installer program performs the following tasks, removing previously installed versions of these programs as needed:
   a. Two prerequisite rpms are installed.
   b. The z1090 or z1091 rpm is installed, mostly in `/usr/z1090/bin` (`/usr/z1091/` is not used).
   c. A set of `man` files is loaded into `/usr/z1090/man`.
   d. A `/usr/z1090/uim` directory is created.
5. You must change two Linux files: */etc/sysctl.conf* and the `.bashrc` file for your user ID. You can edit these files directly, with the details shown in the SG24-8205 document, or you can run two zPDT scripts:

```
# /usr/z1090/bin/aws_sysctl     (You must be root to use this command)
$ /usr/z1090/bin/aws_bashrc     (You must not be root to use this command)
```

[3] This example assumes that ROM is the title of the CD or DVD. You must determine the title of your CD or DVD.
[4] The "./" characters before the file name tell Linux to run this file from the current directory.

6. Both these command scripts will prompt you through their execution. Take the default options. The `sysctl.conf` parameters affect the maximum size of the emulated z Systems created by zPDT. The default values are good for most initial users.[5] The changes to your `.bashrc` file are primarily to add the zPDT directory to your search paths.

> **Note:** After completing these steps, you should seldom need *root* authority (on the base Linux system) when working with zPDT.

7. Close and reopen your Linux command window (to pick up the changes to the `.bashrc` file) and issue the **`z1090instcheck`** command. It makes a few checks in your zPDT installation and reports any problems.

   You can install zPDT without having a zPDT token. However, you cannot start zPDT operation without one. The token must be *activated*: That is, it must have at least one current license. Normally your zPDT provider takes care of activation.

A new zPDT release is installed exactly as described here. The new release is written over the old one.

## 2.4 z Systems disk planning

Linux for z Systems can use "normal" z Systems disk drives, which are known as CKD drives,[6] or it can use SAN disks. zPDT does not support z Systems SAN disks. The examples in this document use only emulated CKD drives. The current IBM device type for CKD drives is 3390 and so the book often mentions 3390 drives.[7]

Mainframe users often describe disks as direct access storage devices (DASDs). Originally there were other types of DASD, but today there are only disks (or SSD devices that appear as disks). However, the term DASD is still often used.

The last "real" 3390 drive was manufactured about 20 years ago. Since that time, 3390 drives have been emulated on other base disk hardware. There have been various emulations that included various features such as various RAID designs, large caches, and multiple external connections for mirroring and real-time copying. However, at the z Systems operating system level, all these versions are simply 3390 drives using the standard interface functions of the 3390s.

CKD terminology can be confusing for someone new to z Systems usage. A 3390 drive ("volume") consists of tracks and cylinders. A track holds up to about 56K bytes, depending on the size of the data blocks in the track. The data blocks can be variable sizes, a much different design than the fixed-size sectors on PC disks. Linux generally formats the 3390 tracks into 4K blocks and treats them as sectors. A track holds 12 such blocks. A 3390 has 15 tracks per cylinder. Thus, when formatted for Linux use, a cylinder holds 12*15 or 180 blocks, which provides 732,280 usable bytes.

The track and cylinder concept goes back to physical disk drive designs with the first S/360 machines in 1964. The modern 3390 drives still emulate this architecture[8], although they no longer have the same physical construction.

[5] If you need an emulated z Systems larger than about 14 GB or if you have more than about 100 emulated I/O devices, you might need to change the sysctl.conf values.

[6] CKD is for Count, Key, and Data and describes the disk architecture used by z/OS and all its predecessors. More recently, z/VM and z/VSE have supported SCSI drives, but z/OS uses only CKD drives.

[7] The term "3390 volume" is more often used than "3390 drive" but has the same meaning.

[8] Customers have made large investments in programs and data that use CKD architecture. As a result, it is still retained as the "normal" type of disk drives for z Systems systems.

Physical 3390 drives (and there might still be a few in operation from the mid 1990s) were available in four models:

- 3390-1 ("model 1") with 1113 cylinders (or about 750 MB)[9]
- 3390-2 ("model 2") with 2226 cylinders (or about 1500 MB)
- 3390-3 ("model 3") with 3339 cylinders (or about 3 GB)
- 3390-9 ("model 9") with 10,017 cylinders (or almost 9 GB)

These are considered the standard sizes, and a 3390-3 is often a common unit of measurement when discussing disk space. Now that 3390 construction has moved to emulation rather than physical designs, it is possible to emulate a 3390 with any number of cylinders. These sizes are sometimes expressed in units of 1113 cylinders. For example, a 3390-22 would have 22*1113=24,486 cylinders or about 16 GB. *Model 22* is an informal terminology and not everyone agrees on the exact workings of this terminology.

Your Linux for z Systems installation requires at least one 3390 drive. Use a model 10, which has 10*1113=11130 cylinders or about 8 GB when formatted for Linux use. This is larger than needed, but it provides a reasonable example. It is possible to spread Linux out over a number of 3390 drives, but these examples are as simple as possible.

An emulated 3390 is a single file in the base Linux system. It has a normal Linux file name and is created by the zPDT **alcckd** command[10]:

```
$ alcckd /z/LINUX1 -d3390-10
```

This command creates an emulated 3390 volume in the base Linux file `/z/LINUX1`.[11] The directory location and file name are arbitrary. Use meaningful names when possible. The 3390 volume is not yet formatted for Linux use. This part is done as part of Linux for z Systems installation. As described in "Configurations used" on page 5, the example has additional smaller drives. These drives can be created with a command such as:

```
$ alcckd /z/LINUX2 -d3390-3
```

Create at least one 3390 volume to help test your zPDT startup using the sample command shown above. You can delete the new Linux file when you no longer need it. You can create and format more 3390 volumes after you install Linux for z Systems. Stop zPDT when adding volumes.

## 2.5 zPDT device map

zPDT produces emulated z Systems. The device map ("devmap") provides the z Systems specifications for zPDT. A devmap is a flat file in the base Linux that you create with your preferred Linux editor. It is usually placed in the home directory of the base Linux user ID that controls zPDT. The devmap file name is arbitrary, but names such as `devmap1` are common. Devmap statements can be written in uppercase, lowercase, or both. However, Linux path and file names must be in the correct case.

The following is a sample devmap used for the initial zPDT startup test[12]:

```
[system]
memory 3000m
```

[9] There are a few more cylinders on all the drive models to allow for spare and diagnostic uses. The 1113 cylinders (and multiples of this amount) are the ones available for data.

[10] The **alcckd** command is entered in a command window on the base Linux system.

[11] In other zPDT documentation, uppercase names are used for emulated 3390 volumes. However, doing so is not required.

[12] The specific devmaps that were used for each distribution are listed later.

```
processors 3                        #hash symbols indicate comments

[manager]
name awsckd 22
device 02A0 3390 3990 /z/LINUX1      #These are your volumes that you created
device 02A1 3390 3990 /z/LINUX2      #with the alcckd command

[manager]                           #this creates your first LAN
name awsosa 33 --path=a0 --pathtype=OSD --tunnel_intf=y
device 0400 osa osa
device 0401 osa osa
device 0402 osa osa
```

The [system] and [manager] stanzas must have the square brackets as part of the [system] and [manager] keywords. This initial devmap defines a z Systems with 3000 MB memory (about 3 GB) and three processors ("cores"). This configuration requires a 1090 (or 1091) token with at least three zPDT licenses.

The [manager] stanzas are for zPDT device managers. The *awsckd* device manager emulates 3390 disk drives and the *awsosa* device manager emulates z Systems LAN interfaces. The number immediately after the device manager name (22 and 33 in the example) is an arbitrary number (up to four hex digits) that must be different for each [manager] stanza. Some device managers require extra parameters. The awsosa parameters are described later.

The general format for a device statement is shown in Figure 2-1.

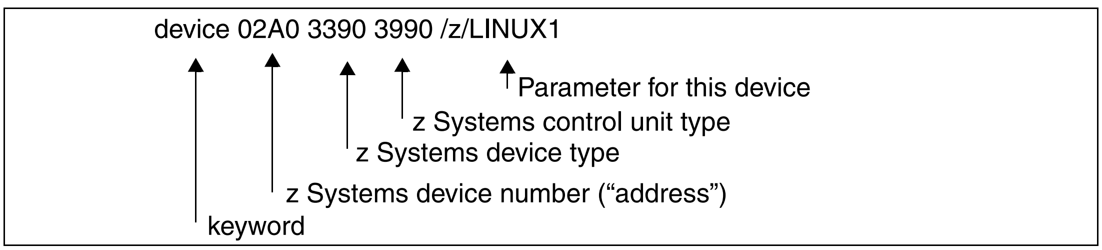


*Figure 2-1 Device statement format*

A z Systems *device number* (most commonly known as an "address") can be the result of a rather complex arrangement on a large z Systems and can have implications for the z Systems operating system to be used. However, for Linux for z Systems on zPDT it is just a three or four digit hexadecimal number that must be different for each device. In some cases, as with OSA devices, the device numbers are in ascending sequence. The 02A0 and 02A1 in this example are arbitrary, but you might start with these numbers in your devmap.

The z Systems *device type* must conform to the device manager type. In this example, the type is 3390 for the awsckd device manager and OSA for the awsosa device manager. The z Systems *control unit type* is three or four characters that are not used currently. Use the control unit types shown in the examples.

The *device parameter* (if needed) provides details for a specific device. In this example, the 3390 disk drive at "address" 02A0 is emulated using base Linux file `/z/LINUX1`. This emulated volume is created in the base Linux file with the **`alcckd`** command.

You will have one device statement for each z Systems device you access (usually emulated disk drives). However, a TCP/IP Ethernet interface requires three "devices" as shown.

Linux for z Systems makes the z Systems device addresses appear more complex. Instead of 02A0 for a disk drive in the example, it appears as 0.0.02A0. The leading “0.0.” can have other numbers with a large z Systems configuration, but will always be “0.0.” for a zPDT system. Also, 02A0 is often written as 2A0 (without the leading zero) unless there is a specific need for four characters in a parameter.

### 2.5.1 LAN interface

The base PC Linux has a restriction partly illustrated in the high-level diagram in Figure 2-2. In this case, the base Linux operates with TCP/IP in the normal way. The z Systems operating system (running under zPDT) has its own TCP/IP. The two can independently share a PC LAN adapter by using different IP addresses to communicate with the outside.[13] However, the two TCP/IP “stacks” cannot communicate with each other through this single LAN adapter.

To work around this restriction, create a pseudo LAN device that connects the base Linux TCP/IP to the z Systems TCP/IP. This is known as a “tunnel” or “tap” device. The devmap example creates the tunnel and, for the example devmap, does *not* create a LAN connection to the LAN adapter.

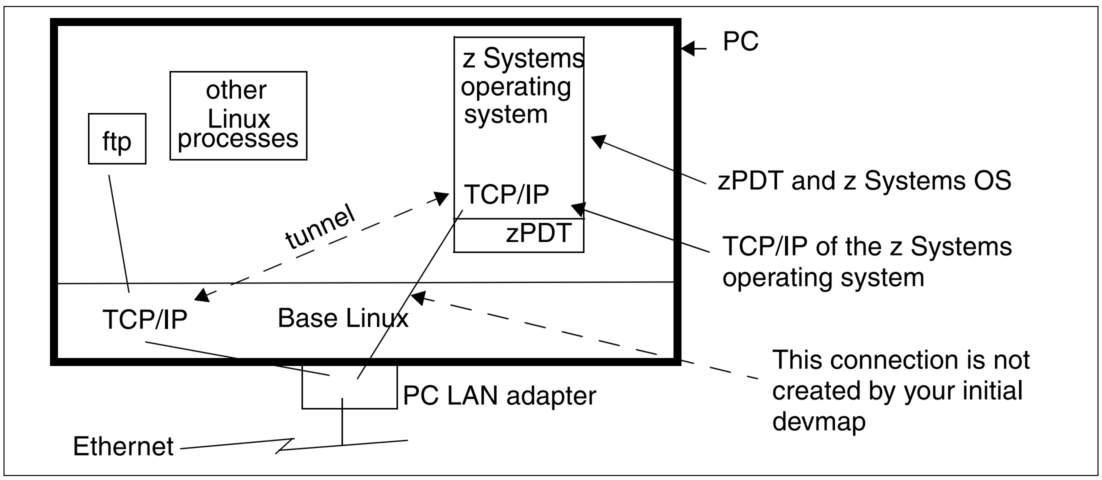


*Figure 2-2 Initial LAN overview under zPDT*

The **`--path=a0 --pathtype=OSD --tunnel_intf=y`** parameters on the `awsosa` device manager statement create the tunnel. The --path parameter specifies a channel path (CHPID). It is usually A0 for the first tunnel. If desired, you can verify this with the zPDT command **`find_io`**, which is best used after you start zPDT for the first time. The output of **`find_io`** might be confusing, but you want a line that looks something like the this example:

```
A0    tap0     UP,RUNNING     3a:eb:9d:25:2c:a2   10.1.1.1
```

The A0 might be different in your case. The default IP address of the tunnel (on the base Linux side) is 10.1.1.1. Assign an appropriate address on the z Systems side. The examples in this book use 10.1.1.2.[14]

[13] The zPDT connection to the LAN adapter actually uses low-level interfaces in the base Linux.

[14] You can specify a different base Linux IP address by using more parameters on the awsosa statement. These are explained in the SG24-8205 book.

### Another LAN interface

If your PC has an externally connected LAN interface, you can define it in the devmap with a second OSA [manager] stanza. To do this, you must know the relevant path name. Use the **find_io** command discover the path name:

```
F0    enp0s25  UP, NOT RUNNING f0:de:f1:a6:03:17   *
F1    wlp3s0   UP, RUNNING     00:24:d7:7f:e1:00   192.168.1.107
...
A0    tap0     UP,RUNNING      3a:eb:9d:25:2c:a2   10.1.1.1
```

On the notebook computer used for the examples, the enp0s25 interface was for wired Ethernet (not connected) and the wlp3s0 interface was for wireless (and was connected). Therefore, the path for the second LAN was F1:

```
[manager]                             #this creates your second LAN
name awsosa 33 --path=f1 --pathtype=OSD   #pathtype=OSD is required
device 0404 osa osa                   #use different addresses from
device 0405 osa osa                   #what was used for the first LAN
device 0406 osa osa
```

The second LAN interface is optional at this point in the installation process.

## 2.5.2 Initial zPDT startup

There is nothing to run under zPDT because you have not installed Linux for z Systems yet, but starting and stopping zPDT without errors is a reasonable first test. This test assumes that you have created a devmap in your home directory. The examples assume that the Linux user ID is `ibmsys1`.

Go to your home directory and enter this command:

```
$ awsstart devmapz
```

where *devmapz* is whatever file name you assigned to your devmap. You should then see messages similar to these:

```
IBM z Systems Personal Development Tool (zPDT)
   Licenses Materials - Property of IBM
   5799-ADE
   (c) Copyright IBM Corp. 2007, 2013. All Rights Reserved.
z1090, version 1-6.49.13, build date 05/20/15 for Linux on SUSE 64 bits.
AWSSTA014I Map file name specified: devampz
AWSSTA214I zPDT started in directory '/home/ibmsys1'
OSA code level = 0x4301
AWSDSA010I AWSOSA is ready for chpid: 0xA0 device: 0x400
AWSDSA010I AWSOSA is ready for chpid: 0xA0 device: 0x401
AWSDSA010I AWSOSA is ready for chpid: 0xA0 device: 0x402
AWSEMI314I CPU 0 zPDT License Obtained
AWSSTA059I System Initialization Complete
AWSSTA012I All configured subsystems started
```

`Do not close the base Linux command window used to start zPDT`. Any asynchronous messages from zPDT are sent only to this window. While zPDT is running, enter the **find_io** command. After starting zPDT, you would normally IPL ("boot") your z Systems operating system. However, you cannot do this yet because it has not been installed.

You cannot make dynamic changes to the devmap. It is read only when zPDT is started. However, you can have many devmaps and select a particular devmap when starting zPDT.

## 2.6 Typical operation with zPDT

After Linux for System z is installed under zPDT, operation is fairly simple. Before using it, you need to know two key pieces of information:

1. The "address" (or "device number") of the boot volume for your Linux for z Systems installation. The devmap sample has two disk drives at addresses 02A0 and 02A1. Do not assume that the boot volume is the first one. You need to know the specific address for it.
2. The IP address for your Linux for z Systems installation (or domain name if a name server has been set up). You need this address for an SSH connection to your Linux for z Systems. (If you are working from a command terminal in the base Linux and used IP address 10.1.1.2 for the "other side" of the tunnel, then you can use this IP address with your **ssh** command.)

The startup process involves the following steps:

1. Start the underlying Linux system and open a command window.
2. Enter the command **awsstart** *devmapx* (where *devmapx* is whatever file name you assigned to your devmap.
3. After the zPDT startup messages appear, enter the command **ipl** *nnnn* (where *nnnn* is the address of the emulated 3390 volume that contains the boot files for your Linux for z Systems installation.)
4. Watch the Linux for z Systems startup messages appear in the base Linux terminal. When they stop, use **ssh** to log in to Linux for z Systems.
5. Do not close the base Linux window because any asynchronous messages from zPDT appear here.
6. When you want to stop Linux for z Systems, issue whatever shutdown commands are appropriate for that distribution (such as **shutdown -h now**) and watch the shutdown messages in the base Linux terminal window. When these messages have ceased, enter an **awsstop** command. If you accidentally closed the base Linux window, you can enter the **awsstop** command from any window on the base Linux system.

**Note:** The start-up terminology can be a bit confusing. In the context of this description, "IPL", "boot", and "load" all have the same meaning. **IPL** (Initial Program Load) was a button on all the early S/390 and S/370 systems. More modern z Systems machines have a **Load** icon that serves the same purpose. Early non-mainframe computers often needed bootstrap code and this became boot for current machines. The general meaning is the same for all three terms, although the exact mechanism used is different for different hardware platforms.

Messages from zPDT to the base Linux command window used to start zPDT usually start with the letters *AWS*. You can ask for more information about zPDT messages with the zPDT **msgInfo** command:

```
$ msgInfo AWSDSA010I
$ msgInfo DSA010       (you can omit the leading AWS and trailing character)
```

# 3

# Linux for z Systems (SUSE)

This chapter covers installing SLES 11 Service Pack 3 and SLES 12. Later distributions (such as SLES 12 Service Pack 1) will probably have a similar installation process to SLES 12. There is no need for you to install both distributions. Both distributions are featured to show minor changes and to illustrate the general processes involved in both instances.

This chapter includes the following sections:

- Installing SLES 11 service pack 3
- SLES 12

## 3.1 Installing SLES 11 service pack 3

For this example, the distribution, a single file named SLES-11-SP3-DVD-s390x-GM-DVD1.iso, was placed in a subdirectory in the base Linux home directory named DVD. The path name was /home/ibmsys1/DVD.

The distribution iso file was mounted on a mount point created in the home directory named ISO:

```
# mount -o loop /home/ibmsys1/DVD/SLES-11-SP3-DVD-s390x-GM-DVD1.iso ISO
   mount: /dev/loop0 is write protected, mounting read-only
$ ls ISO
```

The *loop* function in the **mount** command takes the single .iso file (as input) and explodes it into all the component files at the mount point. The **ls** command displayed 29 file or directory names, starting with ARCHIVES.gz. This output verifies that the iso file and the mount function worked correctly.

An example devmap was created in the base Linux home directory named devSLES11:

```
[system]
memory 3000m
processors 3                          #hash symbols indicate comments

[manager]
name awsckd 22
device 02A0 3390 3990 /z/SLES11A      #This volume is about 8.5 GB usable space
device 02A1 3390 3990 /z/SLES11B      #These are smaller volumes
device 02A2 3390 3990 /z/SLES11C

[manager]                             #this creates your first LAN
name awsosa 33 --path=a0 --pathtype=OSD --tunnel_intf=y
device 0400 osa osa
device 0401 osa osa
device 0402 osa osa

[manager]                             #this creates your second LAN
name awsosa 33 --path=f1
device 0404 osa osa                   #use different addresses from
device 0405 osa osa                   #what was used for the first LAN
device 0406 osa osa
```

This devmap assumes that you have created the three 3390 volumes earlier using **alcckd** commands. Your path name in the OSA definitions might differ, and you can skip defining the second LAN shown here. The 3390 file names (such as /z/SLES11A) are arbitrary and are not visible to Linux for z after it is running.

The initial installation process can be divided into three parts:

1. IPL ("boot") the installer and installation kernel.
2. Use VNC to communicate with YAST for the primary Linux configuration and installation.
3. IPL the system again to allow the installation processes to complete.

Each of these steps is described in detail in the following text.

### 3.1.1 Booting the installer and installation kernel

You can use these steps to boot the installer and installation kernel:

1. Start zPDT with your devmap and boot the installation process by using a special zPDT command:

   ```
   $ awsstart devSLES11
      (The usual startup messages appear)
   $ ipl_dvd ISO/suse.ins              (note the underscore in the command)
   AWSMID020I Comment from .ins file: 'SUSE Linux for zSeries.....
   AWSMID021I Enter 'Y' to continue or 'N' to cancel the installation
   Y                                    (you enter Y at this point)
   ```

   Several screens of messages (and some pauses during the messages) are displayed before the first option menu is displayed:

   ```
   Main Menu
     1) Start Installation
     2) Settings
     3) Expert
     4) Exit or Reboot
   ```

2. Normal messages to be sent to the emulated z Systems from a base Linux command window use the **oprmsg** command. For example, use **oprmsg 1** to send a “1” to the z Systems program. However, issuing many **oprmsg** commands is laborious and leads to errors. You can use the Linux command **alias +=oprmsg** to provide a shorter sequence, and you can then use **+ 1** to send the “1” character. Note the space after the plus sign. The following text is the script from the initial Linux for z Systems installation. Note that *ibmsys1@linux-4f4a:~* is the prompt for user `ibmsys1` in the base Linux system.

   A listing of the first part of the installation is included here:

   > **Attention:** In the interest of readability, some of the lines in the following text are truncated or shortened.
   >
   > You will experience multi-second pauses at various points in the startup. Be patient.

   ```
   ibmsys1@linux-4f4a:~ alias +=oprmsg                        < create alias
   ibmsys1@linux-4f4a:~> awsstart devSLES11                      < start zPDT

   IBM z Systems Personal Development Tool (zPDT)
     Licensed Materials - Property of IBM
     5799-ADE
     (C) Copyright IBM Corp. 2007,2013   All Rights Reserved.

   z1090, version 1-6.49.13, build date - 05/29/15 for Linux on SuSE 64bit
   AWSSTA014I Map file name specified: devmapz
   AWSSTA204I zPDT started in directory '/home/ibmsys1'.
   AWSSTA146I Starting independent 1090 instance 'ibmsys1'
   OSA code level = 0x4301
   AWSDSA010I AWSOSA is ready for chpid: 0xA0 device: 0x400
   AWSDSA010I AWSOSA is ready for chpid: 0xA0 device: 0x401
   AWSDSA010I AWSOSA is ready for chpid: 0xA0 device: 0x402
   AWSEMI314I CPU 0 zPDTA License Obtained
   AWSSTA059I System initialization complete
   AWSSTA012I All configured subsystems started
   ibmsys1@linux-4f4a:~> ipl_dvd ISO/suse.ins
   ```

```
AWSMID022I The installation directory is ISO

AWSMID020I Comment from INS file: '* SuSE Linux for zSeries
Installation/Rescue System'
AWSMID021I Enter 'Y' to continue or 'N' to cancel the installation
y                                                   <no + prefix needed here
Memory loaded, 0x8A6E00 Bytes at address 0x0
Memory loaded, 0x4 Bytes at address 0x1040C
Memory loaded, 0x4 Bytes at address 0x10414
Memory loaded, 0xF95AB9 Bytes at address 0x1000000
Memory loaded, 0x47 Bytes at address 0x10480
ibmsys1@linux-4f4a:~> OPRMSG:   Initializing cgroup subsys cpuset
OPRMSG:   Initializing cgroup subsys cpu
OPRMSG:   Linux version 3.0.76-0.11-default (geeko@buildhost) (gcc version
4.3.
                  (multiple Linux initialization lines omitted)
OPRMSG:   >>> SUSE Linux Enterprise Server 11 installation program v3.3.91
(c) 1996-2010 SUSE Linux Products GmbH <<<
                     (more lines omitted)
OPRMSG:   >>> Linuxrc v3.3.91 (Kernel 3.0.76-0.11-default) <<<
OPRMSG:   Main Menu
OPRMSG:   1) Start Installation
OPRMSG:   2) Settings
OPRMSG:   3) Expert
OPRMSG:   4) Exit or Reboot
+ 1                                                          <select option
1
ibmsys1@linux-4f4a:~> OPRMSG:   1
OPRMSG:   Start Installation
OPRMSG:
OPRMSG:   1) Start Installation or Update
OPRMSG:   2) Boot Installed System
OPRMSG:   3) Start Rescue System
+ 1                                                          <select option
1
ibmsys1@linux-4f4a:~> OPRMSG:   1
OPRMSG:   Choose the source medium.
OPRMSG:
OPRMSG:   1) DVD / CD-ROM
OPRMSG:   2) Network
+ 2                                                          <select option
2
ibmsys1@linux-4f4a:~> OPRMSG:   2
OPRMSG:   Choose the network protocol.
OPRMSG:   1) FTP
OPRMSG:   2) HTTP
OPRMSG:   3) HTTPS
OPRMSG:   4) NFS
OPRMSG:   5) SMB / CIFS (Windows Share)
OPRMSG:   6) TFTP
+ 1                                                          <select option
1
ibmsys1@linux-4f4a:~> OPRMSG:   1
OPRMSG:   Detecting and loading network drivers
OPRMSG:   Choose the network device.
```

```
OPRMSG:   1) IBM OSA Express Network card (0.0.0400)
OPRMSG:   2) IBM OSA Express Network card (0.0.0401)
OPRMSG:   3) IBM OSA Express Network card (0.0.0402)
+ 1                                                         <select option
1
ibmsys1@linux-4f4a:~> OPRMSG:   1
OPRMSG:   Please choose the physical medium.
OPRMSG:   1) Ethernet
OPRMSG:   2) Token Ring
+ 1                                                         <select option
1
   (For the example configuration (and devmap) you want to use the
   tunnel for TCP/IP with the read channel at address 0.0.0400,
   the write channel at 0.0.0401, and data channel 0.0.0402.
   No port number is used.)
ibmsys1@linux-4f4a:~> OPRMSG:   1
OPRMSG:   Enter the relative port number
+                                                           <a null entry
ibmsys1@linux-4f4a:~> OPRMSG:
OPRMSG:   Device address for read channel
OPRMSG:    0.0.0400!>
+                                                           <a null entry
ibmsys1@linux-4f4a:~> OPRMSG:
OPRMSG:   Device address for write channel
OPRMSG:    0.0.0401!>
+                                                           <a null entry
ibmsys1@linux-4f4a:~> OPRMSG:
OPRMSG:   Device address for data channel
OPRMSG:    0.0.0402!>
+                                                           <a null entry
ibmsys1@linux-4f4a:~> OPRMSG:
OPRMSG:   Portname to use
+                                                           <a null entry
ibmsys1@linux-4f4a:~> OPRMSG:
OPRMSG:   Enable OSI Layer 2 support?
OPRMSG:   1) Yes
OPRMSG:   2) No
+ 2                                                         <no layer 2
ibmsys1@linux-4f4a:~> OPRMSG:   2
OPRMSG:   (Port 0)
OPRMSG:   qeth.2c6def: register layer 3 discipline
OPRMSG:   qdio: 0.0.0402 OSA on SC 4 using AI:1 QEBSM:0 PCI:1 TDD:1 SIGA: W
AP
OPRMSG:   qeth.736dae: 0.0.0400: Device is a OSD Express card (level: 4301)
OPRMSG:   with link type OSD_1000 (portname: )
OPRMSG:   qeth.47953b: 0.0.0400: Hardware IP fragmentation not supported on
eth
OPRMSG:   qeth.d7fdb4: 0.0.0400: VLAN enabled
OPRMSG:   qeth.e90c78: 0.0.0400: Multicast enabled
OPRMSG:   qeth.5a9d02: 0.0.0400: IPV6 enabled
OPRMSG:   qeth.184d8a: 0.0.0400: Broadcast enabled
OPRMSG:   qeth.9c4c89: 0.0.0400: Outbound TSO not supported on eth0
OPRMSG:   Automatic configuration via DHCP?
OPRMSG:   1) Yes
OPRMSG:   2) No
```

```
+ 2                                                        <no DHCP here
ibmsys1@linux-4f4a:~> OPRMSG:   2
OPRMSG:   Enter your IPv4 address.
OPRMSG:   Example: 192.168.5.77/24
+ 10.1.1.2                                     <address for new Linux for
z
ibmsys1@linux-4f4a:~> OPRMSG:   10.1.1.2
OPRMSG:   Enter your netmask. For a normal class C network, this is usually
OPRMSG:   255.255.255.0
OPRMSG:    255.255.255.0!>
+                                                          <null entry
ibmsys1@linux-4f4a:~> OPRMSG:
OPRMSG:   Enter the IP address of the gateway. Leave empty if you don't need
on
+                                                          <null entry
ibmsys1@linux-4f4a:~> OPRMSG:
OPRMSG:   Enter your search domains, separated by a space:
+                                                    <null entry, long
pause
ibmsys1@linux-4f4a:~> OPRMSG:
OPRMSG:   Enter the IP address of your name server. Leave empty or enter
"+++" if you don't need one
+                                                          <null entry
ibmsys1@linux-4f4a:~> OPRMSG:
OPRMSG:   Enter the IP address of the FTP server
OPRMSG:   >
+ 10.1.1.1                                             <base Linux tunnel
ibmsys1@linux-4f4a:~> OPRMSG:   10.1.1.1
OPRMSG:   Enter the directory on the server
+ ISO                                                  <your copy of iso
DVD
ibmsys1@linux-4f4a:~> OPRMSG:   /mnt/disk
OPRMSG:   Do you need a username and password to access the FTP server?
OPRMSG:   1) Yes
OPRMSG:   2) No
+ 1                                                  <for use on base
Linux
ibmsys1@linux-4f4a:~> OPRMSG:   1
OPRMSG:   Enter the user name with which to access the FTP server
+ ibmsys1                                      <userid for ftp on base
Linux
ibmsys1@linux-4f4a:~> OPRMSG:   ibmsys1
OPRMSG:   Enter the password for the FTP server
+ xxxxxx                                 <password for ibmsys1 on base
Linux
ibmsys1@linux-4f4a:~> OPRMSG:
OPRMSG:   Use a HTTP proxy?
OPRMSG:   1) Yes
OPRMSG:   2) No
+ 2                                                        <no proxy
needed
ibmsys1@linux-4f4a:~> OPRMSG:   2
OPRMSG:   Loading Installation System (1/6) (24640 kB) -   0%
                         (lines omitted)
OPRMSG:   Select the display type.
```

```
OPRMSG:   1) X11
OPRMSG:   2) VNC
OPRMSG:   3) SSH
OPRMSG:   4) ASCII Console
+ 2                                                             <use VNC
ibmsys1@linux-4f4a:~> OPRMSG:   2
OPRMSG:   Enter your VNC password
+ xxxxxxxx                                            <must be minimum 8
characters
ibmsys1@linux-4f4a:~> OPRMSG:
OPRMSG:   starting hald...
OPRMSG:    ok
OPRMSG:   starting syslogd (logging to /dev/tty4)...
OPRMSG:    ok
OPRMSG:   starting klogd... ok
OPRMSG:   starting slpd to announce VNC...
OPRMSG:   starting yast...
OPRMSG:
OPRMSG:   starting VNC server...
OPRMSG:   A log file will be written to: /var/log/YaST2/vncserver.log ...
OPRMSG:   ***
OPRMSG:   ***         You can connect to <host>, display :1 now with
vncviewer
OPRMSG:   ***         Or use a Java capable browser on http://<host>:5801/
OPRMSG:   ***
OPRMSG:   (When YaST2 is finished, close your VNC viewer and return to this
window.)
OPRMSG:   Active interfaces:
OPRMSG:   eth0      Link encap:Ethernet  HWaddr 02:A0:A0:A0:A0:A0
OPRMSG:             inet addr:10.1.1.2  Bcast:10.1.1.255  Mask:255.255.255.0
OPRMSG:   --
OPRMSG:   lo        Link encap:Local Loopback
OPRMSG:             inet addr:127.0.0.1  Mask:255.0.0.0
OPRMSG:
OPRMSG:   *** Starting YaST2 ***
```

### 3.1.2 Using VNC to work with YaST2

You can use steps like these to use VNC to work with YaST2:

1. Move to another command window in your base Linux system. Leave the command window with the initial text (as shown above) open.
2. In the new command window, enter the following command[1]:

   ```
   $ vncviewer 10.1.1.2:1
   $ password: xxxxxxxx                  (VNC password you assigned earlier)
   ```

   This example assumes that you assigned `10.1.1.2` as the IP address of your target Linux for z Systems installation. Remember that the `10.1.1.1` is the default IP address of the tunnel on your base Linux.
3. This command should result in a graphics window opening on your base Linux desktop, with the target system YaST2 running in a semi-text manner. This section does not include

[1] As before, "$" indicates the command line for a non-root user on the base Linux. You might have a different name for your VNC client program.

all of the screens involved. Instead, it presents a condensed version of your interactions for this portion of the installation. You might make different choices at various points.

4. Some of the update operations are slow. Wait a second or two when the VNC window is updating to be certain you are seeing the complete update. A busy indicator is displayed while processing that looks like the example shown in Figure 3-1.

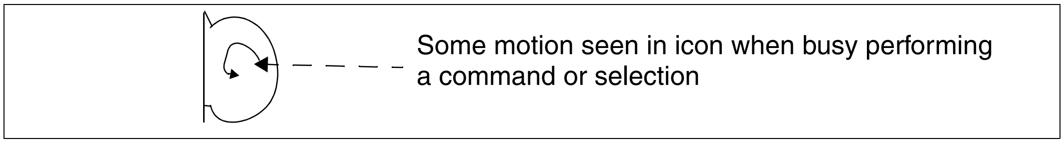


*Figure 3-1 System busy icon (in VNC window)*

The output similar to the following will be displayed:

```
Lanuage: English (US)
Keyboard Layout: English (US)
   Agree to License terms
<next>
Select Configure DASD Disks
DASD Disk Management
   (select first disk with cursor (0.0.02A0) and double click)
Perform Action: Activate
Not Formatted: Format Now. <Continue>
   (Formatting takes some time. Generally, do not format several disks in
   parallel. The progress bar displayed is confusing.)
   (Use the same steps to format additional DASD volumes you may have
   defined.)
<next>        (Some transitions are slow. Be patient.)
<next>
System Probing
Select New Installation
<next>
Set time zone.
<next>
Installation Settings
Possibly select (click) Partitioning (optional)
   (use the whole 02A0 disk volume for Linux and do not partition it. This
   default results in:  (based on your initial 9 GB 3390 volume)
         /dev/dasda1   2 GB swap
         /dev/dasda2   about 6 GB GB ext3
   New users might want to practice installing Linux for z Systems several
   times; if so, initially use the default partition arrangement on one
   volume.)
Possibly select (click) Software (optional)
   (accept the defaults.)
<install>
Confirm licenses <I agree>
Confirm Installation <install>
   (The example installation took about 30 minutes.)
   Finishing Basic Installation
   System will reboot now.
```

5. However, the system does not reboot. Instead, it loads a *disabled wait*, as seen in the initial command window where zPDT was started. You must reboot manually, as described in 3.1.3, “Completing the installation” on page 25. The VNC window closes automatically.

## 3.1.3 Completing the installation

Linux for z Systems is now bootable, but has not completed all of the installation processes. For reasons that were not clear, you sometimes need to stop and restart zPDT to finish the installation. To do so, you can use steps like these:

1. Enter the following commands:

```
$ awsstop
   (wait for “Shutdown Complete” message)
$ a wsstart devSLES11
```

2. Your first boot (IPL) of the new system requires special care, as shown in the following text. Go to your first command window on the base Linux and enter the following zPDT command:

```
$ ipl 02A0
```

   This should result in the following interaction:

```
ibmsys1@linux-4f4a:~> OPRMSG:   zIPL v1.15.0-0.136.13 interactive boot menu
OPRMSG:    0. default (SLES11_SP3)
OPRMSG:    1. SLES11_SP3
OPRMSG:    2. Failsafe_1
OPRMSG:    3. ipl
OPRMSG:   Please choose (default will boot in 10 seconds):
+ 3                                                        <do this quickly
   (Do not let this default to option 0. You must select option 3 within 10
   seconds.)
OPRMSG:   Booting ipl...
                 (A long series of startup messages omitted here,
                  including some long pauses.)
OPRMSG:   ***
OPRMSG:   ***  Please return to your X-Server screen to finish installation
OPRMSG:   ***
OPRMSG:   starting VNC server...
OPRMSG:   A log file will be written to: /var/log/YaST2/vncserver.log ...
OPRMSG:   ***
OPRMSG:   ***        You can connect to <host>, display :1 now with
vncviewer
OPRMSG:   ***        Or use a Java capable browser on http://<host>:5801/
OPRMSG:   ***
OPRMSG:   (When YaST2 is finished, close your VNC viewer and return to this
window.)
```

3. Open another command window of your base Linux system and start a VNC viewer again:

```
$ vncviewer 10.1.1.2:1
```

4. Perform the following actions in the YaST2 displayed by VNC:

```
Set and confirm a password for the root user
<next>
Set a host and domain name. (use “SLES11” and “site” and do not allow DHCP
to change the name.)
<next>                         (long pause while setting network
configuration)
disable the firewall for your initial tests.
<next>
skip the network test.
```

```
<next>                        (followed by several screen updates. Be
patient.)
<next>                        (creating a certificate and more pauses)
User authentication method: Local
New local user: define ibmsys1 as a new local user to receive system mail.
   (You should select your own name. The example uses ibmsys1 as the primary
   user on both the base Linux and Linux for z Systems. This could be
   confusing to some people.)
<next>
Writing the system configuration
Release notes
<next>
Hardware Configuration
<next>
Installation Completed.
uncheck the Clone This System For AutoYaST option.
<Finish>
```

This process caused the VNC window to close, and some shutdown and startup messages to appear in the zPDT command window followed by a login prompt.

5. Start the vncviewer in a base Linux command window:

```
$ vncviewer 10.1.1.2:1
```

This command (after a pause) creates a fully graphic Linux window.

6. Log in, start YaST, and define a new user. You can connect by using **ssh** (to 10.1.1.2) but not with Telnet. You can use an ASCII mode login from the base console window that you used to start zPDT, but this requires entering **oprmsg** (or the "+" shortcut used in the example) for every line and is not practical. The example used a **df -h** command to observe that Linux for z Systems used 3.2 GB disk space (all in the root directory in the example configuration).
7. Do not attempt to start YaST from the base console window you used to start zPDT. The z Systems console functions that it provides are basic. When it detects a test interface, YaST attempts to use a full-screen mode that does not work in this case. Instead, make an **ssh** connection and use the text-mode YaST through it.
8. Using YaST through a VNC window, **shutdown** Linux for z Systems. Doing so displays a Power Manger Still Running message.
9. Enter **Shutdown Anyway**. This action produced the followign output in the zPDT window:

```
INIT: Switching to runlevel: 0
(more shutdown messages)
The system will be halted immediately.
```

10. Stop zPDT with the **awsstop** command in the base Linux command window. Then start zPDT again with a **awsstart devSLES11** command. The startup menu appeared again:

```
OPRMSG:    0. default (SLES11_SP3)
OPRMSG:
OPRMSG:    1. SLES11_SP3
OPRMSG:    2. Failsafe_1
OPRMSG:    3. ipl
OPRMSG:
OPRMSG:   Please choose (default will boot in 10 seconds):
```

11. This time allow it to default to option 0. After several startup messages, you receive a login prompt in the zPDT starting window.
12. Start a VNC viewer and perform any additional system setup through this interface.

## 3.1.4 Additional work

Many postinstallation jobs need to be completed. This book only addresses topics that are related to use with zPDT. For information about the jobs not detailed in the following sections, see *The Virtualization Cookbook for IBM z/VM 6.3, RHEL 6.4, and SLES 11 SP3*, SG24-8147.

## 3.1.5 Stopping the system

Typically, you use the VNC desktop to stop Linux for z Systems. Wait for the `The System will be halted immediately` message in the base Linux window, then enter **awsstop** to stop zPDT. It is sometimes necessary to stop and restart zPDT when you want to restart Linux for z Systems.

If you inadvertently closed the base system Linux window that was used to start zPDT, you will not see the `The System will be halted immediately` message. In this case, wait about a minute before entering the **awsstop** command in another base Linux window.

## 3.1.6 FTP

You can use these steps to allow an FTP link:

1. Use YaST to install vsftp. The system remembers the FTP link to `10.1.1.1` and installed it from the `.iso` files mounted there.
2. Configure vsftp, again through YaST. Elect it to have it start through `xinetd`.
3. The default mode is anonymous only. The YaST windows for FTP allow you to select anonymous users, authenticated users, or both.
4. By default, uploading is disabled. This can be enabled in the same window.
5. The Expert Settings window allows enabling of SSL, with several options.

If the installation process described in this chapter was followed, you should be able to make the following connections:

```
(from Linux for z Systems): $ ftp 10.1.1.1       (connect to base Linux)
```

and

```
(from the base Linux)     : $ ftp 10.1.1.2       (connect to Linux for z)
```

## 3.1.7 More DASD

Almost all users require more disk space than what was created during the installation described above. Complete steps like these to add more space:

1. Decide on the number and size of 3390 volumes involved. You can create a single very large 3390 volume and create multiple partitions. However, Linux for z Systems DASD are limited to three partitions. At the other extreme, you can generate many small 3390 volumes with a single Linux partition on each one.

   There is no general recommendation about your disk requirements. As a starting point, consider several moderate-size 3390 volumes, perhaps 10-20 GB each, with one or two partitions per volume. Your first Linux for z Systems installations will be like a trial, making it a good time to experiment with disk layouts.

2. Create the 3390 volumes with the zPDT command **alcckd**:

   ```
   $ alcckd /z/MyNextLinuxVolume -d3390-5      (this command on your base Linux)
   ```

   This command creates a single Linux file (on the base Linux) with 5*1113 = 5565 cylinders. This configuration provides about 732K usable bytes/cylinder, or roughly 4 GB total space. The space is in raw 3390 format and is not yet usable by Linux for z Systems. Creating emulated volumes (with **alcckd**) is best done when zPDT is not running.

3. Add the new volumes to your zPDT devmap:

   ```
   ...
   [manager]
   name awsckd 22
   device 02B0 3390 3990 /z/SLES11A           #this will be the boot drive
   device 02B1 3390 3990 /z/SLES11B
   device 02BA 3390 3990 /z/MyNextLinuxVolume  #use of z directory is arbitrary
   ....
   ```

   As mentioned earlier, the device number or “address” such as 02BA for a new drive in this example is arbitrary. It has more meaning on a “real” z Systems mainframe, but this is not relevant to zPDT systems running Linux for z Systems. The addresses need not be sequential, but it is common practice to make them sequential (or almost sequential).

4. After restarting zPDT and IPLing Linux for z Systems, use the YaST DASD option. The newly added device should be visible as channel `0.0.02BA` (based on the devmap example). Select the device and **Perform Action** → **Activate**. This should offer to format the volume. Select **Continue** to do so. As a result of formatting, the device should now have a name, such as `/dev/dasdd`. Select **<next>** to exit from the DASD Management function. This can take several seconds.

5. Decide on mount points in Linux for z Systems for the new volumes or partitions. For new users, generally use mount points that are not part of the root file system. The following mount points are created in the example home directory:

   ```
   $ cd ~                        (go to home directory)
   $ mkdir mountpoint1           (not very clever names, but you get
   $ mkdir mountpoint2           the general idea)
   ```

6. Find the **Partitioner** function in YaST; select **Yes** to continue. Double-click the new device name (`/dev/dasdd` in this example). A screen is displayed to **Add**, **Edit**, **Move**, **Resize**, and otherwise manipulate partitions. The example involves adding two partitions, each 250 MB, and selecting `Ext3` file systems and appropriate mount points. After you select <**Finish**>, the partitions are created and added to the mount table (if you selected this option).

More disks can be added without using YaST. This process involves the **chccwdev**, **fdasd**, and **mke2fs** commands, and then editing `/etc/fstab`. As a new user on this platform, use the YaST interface.

### 3.1.8 Network expansion

The initial installation had only the “tunnel” network connection between Linux for z Systems and the base Linux. This connection was sufficient for running a VNC viewer on the base Linux. The next step is to connect to the external Internet. This connection can be done using the same PC LAN adapter that is used by the base Linux, as shown in Figure 3-2.

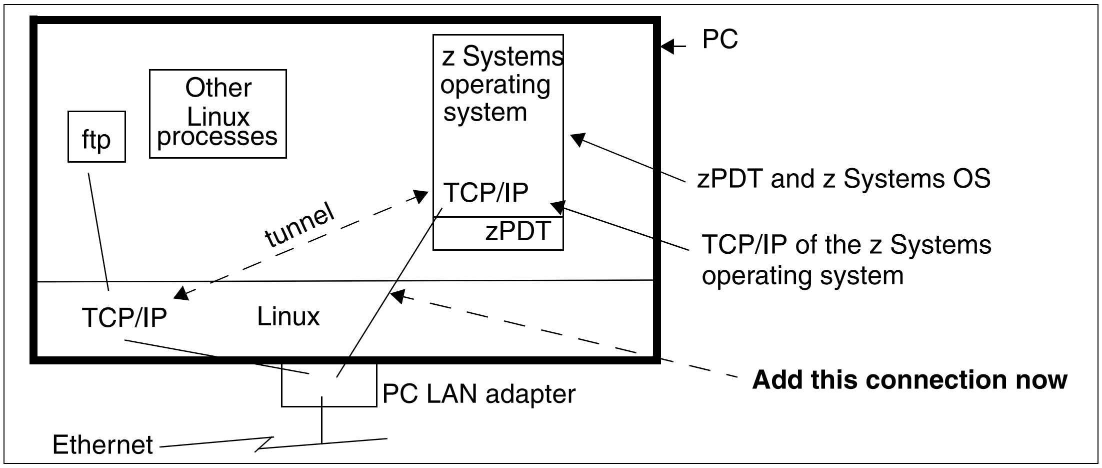


*Figure 3-2 External Ethernet connection*

In the example, the base Linux has the address `192.168.1.107`, which was assigned by DHCP on a local router. The static address `192.167.1.110` was arbitrarily used for Linux for z Systems.

#### If you have not already defined a second LAN

The first step is to add another network interface to the devmap. Remember that z Systems need three OSA addresses for a TCP/IP connection. The 400 address range is arbitrary, but generally use three consecutive addresses for each definition. A separate device manager definition is required:

```
....
[manager]                            #this creates your tunnel LAN
name awsosa 33 --path=a0 --pathtype=OSD --tunnel_intf=y
device 0400 osa osa
device 0401 osa osa
device 0402 osa osa

[manager]                             #this is for a real LAN
name awsosa 44 --path=F1 --pathtype=OSD    #this pathtype is required
device 0404 osa osa
device 0405 osa osa
device 0406 osa osa
```

As before, the `44` number is arbitrary but different than any other device manager number. The path (CHPID) number, `F1` in this example, might need to be adjusted. On the base Linux, enter a **`find_io`** command. The following results are shown:

```
$ find_io
Path Interface  State            MAC address       IPV4          IPV6
---- ---------  ---------------- -------------     ----------    --------
 F0  enp0s25    UP, NOT RUNNING  f0:de:f1:a6:0e:17 *             *
 F1  wlp3s0     UP, RUNNING      00:24:d7:7f:e1:00 192.167.1.107 ........
```

```
A0  tap0      UP, RUNNING     3a:d0:f9:6a:05:0a 10.1.1.1     ........
A1  tap1      DOWN            etc
```

On the example, the `wlp3s0` interface is the wireless interface that was connected to the local router. The `enp0s25` is an Ethernet wired interface that is not currently being used. To use the same wireless interface for Linux for z Systems, use the `F1` path in the devmap. pathtype=OSD is required in the devmap.

### Configuring and activating

You can use the following steps to configure and activate the system:

1. Update the devmap.
2. Restart zPDT.
3. IPL Linux for z Systems.
4. Connect with VNC.
5. Using YaST, select **Network Settings**. This window initially displayed the OSA Express Network cards (`0.0.0404` - `0.0.0406`) as not configured.
6. Select the first address in this range (`0404`) and select **Edit**.
7. For `Port Name`, enter a name of your choice.
8. Leave the `Port Number` blank.
9. Enter the `Read Channel` as `0.0.0404`, the `Write Channel` as `0.0.405`, and the `Control Channel` as `0.0.406`.
10. On the next window, enter `192.168.1.110` as a static address.
11. After exiting from Network Settings and YaST, an **/sbin/ifconfig** command showed both the tunnel (`10.1.1.2`, assigned the `eth0` name) and the new link (`192.167.1.110`, assigned the `eth1` name).
12. Connect to Linux for z Systems with a **vncviewer** command on a Microsoft Windows system on the same local network to verify operation.

There are many options for network setup, and this book does not cover anything more complex than the simple operation already outlined.

Remember two important facts about networking with zPDT and Linux for z Systems:

- You cannot communicate between the base Linux and Linux for z Systems by using the same network adapter on the PC. A tunnel is required for this communication.
- IP addresses outside a local network are complex topics. Do not attempt to use an address outside the `10.x.x.x` or `192.168.x.x` range unless you clearly understand those issues.

## 3.2 SLES 12

You can use the following example steps to install SLES 12:

1. Create a subdirectory in the base Linux home directory named DVD. The path name in the example is `/home/ibmsys1/DVD`.
2. Place the distribution (which is a single file) named `SLE-12-Server-DVD-s390x-GM-DVD1.iso` in this directory.
3. Make a mount point in your home directory named ISO.[2]

4. Mount the distribution iso file on this mount point[3]:

```
# mount -o loop /home/ibmsys1/DVD/SLE-12-Server-DVD-s390x-GM-DVD1.iso  ISO
   mount: /dev/loop0 is write protected, mounting read-only
$ ls ISO
```

5. The **ls** command displayed various file and directory names, starting with ARCHIVES.gz. This command verifies that the iso file and the mount function work correctly.

6. Create a devmap in your base Linux home directory named devSLES12:

```
[system]
memory 3000m
processors 3                         #hash symbols indicate comments

[manager]
name awsckd 22
device 02B0 3390 3990 /z/SLES12A
device 02B1 3390 3990 /z/SLES12B
device 02B2 3390 3990 /z/SLES12C

[manager]                            #this creates your first LAN
name awsosa 33 --path=a0 --pathtype=OSD --tunnel_intf=y
device 0400 osa osa
device 0401 osa osa
device 0402 osa osa

[manager]                            #this creates your second LAN
name awsosa 33 --path=f1
device 0404 osa osa                  #use different addresses from
device 0405 osa osa                  #what was used for the first LAN
device 0406 osa osa
```

This devmap assumes that you have created the three 3390 volumes earlier using **alcckd** commands. Your path name in the OSA definitions might differ, as described earlier, and you can skip defining the second LAN shown here.

The initial installation can be divided into three parts:

1. IPLing ("booting") the installer and installation kernel.
2. Using VNC to communicate with YAST for the primary Linux configuration and installation.
3. IPLing again to allow the installation processes to complete.

Each of these steps is described in detail in the following sections.

## 3.2.1 Booting the installer and installation kernel

Complete these steps to boot the installer and installation kernel:

1. Start zPDT with your devmap.
2. Boot the installation process using a special zPDT command:

```
$ awsstart devSLES12
   (The usual startup messages appear)
$ ipl_dvd ISO/suse.ins              (note the underscore in the command)
```

[2] This same mount point was used for installing all four Linux distributions described in this book, mounting one at a time.

[3] The **-o loop** option of the **mount** command explodes the iso file into its component files. This process avoids any need to restore the iso distribution file to a DVD.

```
AWSMID020I Comment from .ins file: 'SUSE Linux for zSeries.....
AWSMID021I Enter 'Y' to continue or 'N' to cancel the installation
Y                                     (you enter Y at this point)
```

Several screens of messages (and some pauses during the messages) are displayed before the first option menu:

```
Main Menu
  1) Start Installation
  2) Settings
  3) Expert
  4) Exit or Reboot
```

3. Normal messages to be sent to the emulated z Systems machine from a base Linux command window using the **oprmsg** command. For example, enter **oprmsg 1** to send a "1" to the z Systems program. However, entering **oprmsg** many times is laborious and leads to errors. You can use the Linux command **alias +=oprmsg** to provide a shorter sequence, and then use **+ 1** to send the "1" character. Note the space after the plus sign. The following text is the script from the initial Linux for z Systems installation. Note that `ibmsys1@linux-4f4a:~` is the prompt for user `ibmsys1` in the example base Linux system.

   A listing of the first part of the installation is included here. Some lines have been omitted to shorten the listing.

```
ibmsys1@linux-4f4a:~> alias +=oprmsg
ibmsys1@linux-4f4a:~> awsstart devSLES12
IBM z Systems Personal Development Tool (zPDT)
  Licensed Materials - Property of IBM
  5799-ADE
  (C) Copyright IBM Corp. 2007,2013   All Rights Reserved.

z1090, version 1-6.49.13, build date - 05/29/15 for Linux on SuSE 64bit


AWSSTA014I Map file name specified: devSLES12
AWSRAS181I 0 Snapdump incident(s), RAS trace and RAS log files occupy 0
bytes in /home/ibmsys1/z1090/logs
AWSRAS182I Associated files, logs and core files occupy 0 bytes in
/home/ibmsys1/z1090/logs
AWSSTA023I DEVMAP contains warnings, please use "awsckmap" for more
information.
AWSSTA204I zPDT started in directory '/home/ibmsys1'.
AWSSTA146I Starting independent 1090 instance 'ibmsys1'
OSA code level = 0x4301
AWSDCK005E Could not open file '/z/SLES12C' on device 02B2, ERRNO=2
AWSDSA010I AWSOSA is ready for chpid: 0xA0 device: 0x400
AWSDSA010I AWSOSA is ready for chpid: 0xA0 device: 0x401
AWSDSA010I AWSOSA is ready for chpid: 0xA0 device: 0x402
AWSDSA010I AWSOSA is ready for chpid: 0xF1 device: 0x404
AWSEMI314I CPU 0 zPDTA License Obtained
AWSDSA010I AWSOSA is ready for chpid: 0xF1 device: 0x405
AWSDSA010I AWSOSA is ready for chpid: 0xF1 device: 0x406
AWSEMI314I CPU 1 zPDTA License Obtained
AWSSTA059I System initialization complete
AWSSTA012I All configured subsystems started
ibmsys1@linux-4f4a:~> AWSEMI314I CPU 2 zPDTA License Obtained
ibmsys1@linux-4f4a:~> ipl_dvd ISO/suse.ins
AWSMID022I
```

```
 The installation directory is ISO/
AWSMID020I Comment from INS file: '* SuSE Linux for zSeries
Installation/....
AWSMID021I Enter 'Y' to continue or 'N' to cancel the installation
y
Memory loaded, 0xA45800 Bytes at address 0x0
Memory loaded, 0x4 Bytes at address 0x1040C
Memory loaded, 0x4 Bytes at address 0x10414
Memory loaded, 0x19D8940 Bytes at address 0x1000000
Memory loaded, 0x14 Bytes at address 0x10480
ibmsys1@linux-4f4a:~> OPRMSG:   Initializing cgroup subsys cpuset
      (omit many lines here)
OPRMSG:   >>> linuxrc 5.0.26 (Kernel 3.12.28-4-default) <<<
OPRMSG:   Main Menu
OPRMSG:   0) <-- Back <--
OPRMSG:   1) Start Installation
OPRMSG:   2) Settings
OPRMSG:   3) Expert
OPRMSG:   4) Exit or Reboot
+ 1                                                 (select option 1)
OPRMSG:   Start Installation
OPRMSG:   0) <-- Back <--
OPRMSG:   1) Installation
OPRMSG:   2) Upgrade
OPRMSG:   3) Rescue System
OPRMSG:   4) Boot Installed System
OPRMSG:   5) Network Setup
+ 1                                                 (select option 1)
OPRMSG:   Choose the source medium.
OPRMSG:   0) <-- Back <--
OPRMSG:   1) DVD / CD-ROM
OPRMSG:   2) Network
OPRMSG:   3) Hard Disk
+ 2                                              (select option 2)
OPRMSG:   Choose the network protocol.
OOPRMSG:   0) <-- Back <--
OPRMSG:   1) FTP
OPRMSG:   2) HTTP
OPRMSG:   3) HTTPS
OPRMSG:   4) NFS
OPRMSG:   5) SMB / CIFS (Windows Share)
OPRMSG:   6) TFTP
+ 1                                              (select option 1)
OPRMSG:   Choose the network device.
OPRMSG:   0) <-- Back <--
OPRMSG:   1) IBM OSA Express Network card (0.0.0400)
OPRMSG:   2) IBM OSA Express Network card (0.0.0401)
OPRMSG:   3) IBM OSA Express Network card (0.0.0402)
OPRMSG:   4) IBM OSA Express Network card (0.0.0404)
OPRMSG:   5) IBM OSA Express Network card (0.0.0405)
OPRMSG:   6) IBM OSA Express Network card (0.0.0406)
+ 1                                              (select option 1)
OPRMSG:   Enter the relative port number. (Enter '+++' to abort).
+                                                (enter a null line)
OPRMSG:   Device address for read channel. (Enter '+++' to abort).
```

```
OPRMSG:    0.0.0400!>
+                                                 (enter a null line)
OPRMSG:   Device address for write channel. (Enter '+++' to abort).
OPRMSG:    0.0.0401!>
+                                                 (enter a null line)
OPRMSG:   Device address for data channel. (Enter '+++' to abort).
OPRMSG:    0.0.0402!>
+                                                 (enter a null line)
OPRMSG:   Portname to use. (Enter '+++' to abort).
+                                                 (enter a null line)
OPRMSG:   Enable OSI Layer 2 support?
OPRMSG:   0) <-- Back <--
OPRMSG:   1) Yes
OPRMSG:   2) No
+ 2                                               (select option 2)
OPRMSG:   Examples: 192.168.5.77/24 2001:db8:75:fff::3/64.
+ 10.1.1.2/24                                     (IP for target system)
OPRMSG:   Enter your gateway IP address.
+ 10.1.1.1                                        (could also be null)
OPRMSG:   Enter your name server IP address.
+                                                 (use a null line)
OPRMSG:   Enter your search domains, separated by a space. (Enter '+++' to
+                                                 (use a null line)
OPRMSG:   Enter the name of the FTP server. (Enter '+++' to abort).
+ 10.1.1.1                                        (IP of base Linux)
OPRMSG:   Enter the directory on the server. (Enter '+++' to abort).
+ ISO                                             (mount point on base Linux)
OPRMSG:   Do you need a username and password to access the FTP server?
OPRMSG:   0) <-- Back <--
OPRMSG:   1) Yes
OPRMSG:   2) No
+ 1                                               (you need a username/pw)
OPRMSG:   Enter the user name with which to access the FTP server.
+ ibmsys1                                         (your userid on base Linux)
OPRMSG:   Enter the password for the FTP server. (Enter '+++' to abort).
+ xxxxxx                                          (password for ibmsys1 on
base)
OPRMSG:   Use a HTTP proxy?
OPRMSG:   0) <-- Back <--
OPRMSG:   1) Yes
OPRMSG:   2) No
+ 2                                               (no proxy)
OPRMSG:   Loading Installation System (1/5) -       0%
       (many messages not shown)
OPRMSG:   Select the display type.
OPRMSG:   0) <-- Back <--
OPRMSG:   1) X11
OPRMSG:   2) VNC
OPRMSG:   3) SSH
OPRMSG:   4) ASCII Console
+ 2                                               (select VNC)
OPRMSG:   Enter your VNC password. (Enter '+++' to abort).
+ xxxxxxxx                                        (minimum 8 characters!)
         .......
OPRMSG:   starting VNC server...
```

```
OPRMSG:   ***        You can connect to <host>, display :1 now with
vncviewer
OPRMSG:   ***        Or use a Java capable browser on http://<host>:5801/
OPRMSG:   (When YaST2 is finished, close VNC viewer,return to this window.)
OPRMSG:   eth0      Link encap:Ethernet  HWaddr 02:A0:A0:A0:A0:A0
OPRMSG:             inet addr:10.1.1.2  Bcast:10.1.1.255  Mask:255.255.255.0
OOPRMSG:   lo        Link encap:Local Loopback
OPRMSG:             inet addr:127.0.0.1  Mask:255.0.0.0
```

4. Open a second command window on the base Linux and enter the command **vncviewer 10.1.1.2:1**. This command opens a graphic window in your base Linux system and prompts you through the next phase of the installation. This type of operation should be familiar to experienced PC Linux users, so the steps are just summarized here:
5. Agree to the license terms.
6. Configure  DASD.
7. If your DASD volumes have not been previously formatted for Linux,[4] click **Select All**, **Perform Action**, **Activate**, and then **Continue**.
8. Register your system so that you can obtain online updates.
9. Install any add-on products.
10. Partition using the Expert partitioner or a similar tool. The following are the example partition results:
    - /dev/dasda1      200 MB     /boot/zipl
    - /dev/dasdz2      980 MB     swap
    - /dev/dasda3      6.5 GB     /
    - /dev/dasdb1      1.6 GB     /diskb        *(an arbitrary mount point*
    - (The third drive was not partitioned)
11. Disable **firewall** and **blacklist**, and open an **ssh port**.
12. Select **install**.

    The example installation took about 30 minutes. When it completed, it automatically closed the VNC window and entered a wait state:

    AWSEMI307I Warning! Disabled Wait CPU 0 = 00020001 80000000 00000000 00116A9E
13. IPL (boot) your new system:

    ```
    $ ipl 2b0                              (this command on the base console)
    ```

    The example hung with the message `unlabeled traffic allowed by default`. For unclear reasons, the base Linux needed to be rebooted to resolve the problem. If necessary, start zPDT again:

    ```
    $ awsstart devSLES12
    $ ipl 2b0
    ```
14. This example skips **grub2**, as offered in the SLES 12 startup options. At this point you could start a vncviewer again. If, like the example, you are unable to start an **ssh** session, open another window on the base Linux and enter the command **vncviewer 10.1.1.2:1**.

The vncviewer provides a fully graphic screen, but operation was very slow in the test environment. This slowness was probably due to the default use of gnome 3, which uses complex graphics. The test zPDT setup (emulation, VNC, and so forth) did not provide the hardware acceleration needed for gnome 3. Nevertheless, this VNC window was used to complete additional installation steps.

[4] Creating the volumes with the zPDT **alcckd** command does not format them for Linux use. You need to format new volumes manually.

## Additional installation steps

You can use YaST (in the VNC window) to complete the following additional steps:

1. Select the DASD option and verified that your third volume (dasdc) is present. Partition this volume and assign a mount point.
2. Enable vsftpd.
3. Configure the second LAN interface using static IP address `192.168.1.112`. You can use any address that is appropriate for your configuration.
4. Disable the firewall to allow ssh to connect. You might handle this step differently. In the test environment, the firewall disable option that was selected during installation did not carry over to the final system.

In addition to the VNC interface, you can log in by using ssh and run all the normal Linux commands.

4

# Linux for z Systems (Red Hat)

This chapter covers installing both RHEL 6.7 and RHEL 7.1. At the time of writing, RHEL 6.7 was more widely used, while RHEL 7.1 was being applied to newer installations.

For more information, see the *Red Hat Enterprise Linux 7 Installation Guide*, at:

https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/pdf/Installation_Guide/Red_Hat_Enterprise_Linux-7-Installation_Guide-en-US.pdf

The *Red Hat Enterprise Linux 7 Installation Guide* contains a large section on IBM z Systems installation with specific information for configuring additional LANs and DASD volumes. Some of this information is included in this book, but see the Guide for more information.

This chapter includes the following sections:

- Installing RHEL 6.7
- Installing RHEL 7.1

 

## 4.1 Installing RHEL 6.7

You can use these steps to install REHL 6.7:

1. Download the file `RHEL-6.7-20150702.0-Server-s390x-dvd1.iso` from an approved site. The file is almost 3 GB. You did not need to explode this single .iso file. The user `ibmsys1` was used on the base Linux system in the example.
2. Create two subdirectories in your home directory:

   ```
   $ cd ~                    (go to your home directory)
   $ mkdir ISO               (use this as a mount point)
   $ mkdir DVD               (copy the .iso file into this directory)
   ```

3. Copy the single .iso file into the DVD directory.
4. Change to `root` and mount the .iso file using the **`loop`** option:

   ```
   $ su                      (change to root)
   # mount -o loop DVD/RHEL-6.7-20150702.0-Server-s390x-dvd1.iso ISO
   # exit                    (leave root)
   ```

   After performing the **`mount`**, you should see the required RHEL installation files in the ISO directory. For reasons described later, it is critical that the mount point (ISO in this example) be in the home directory (or a subdirectory) of the base Linux user ID that is installing Linux for z Systems.

   The example used a devmap (in the base Linux home directory) named `devRHEL6`:

   ```
   [system]
   memory 3000m
   processors 3                         #hash symbols indicate comments

   [manager]
   name awsckd 22
   device 02C0 3390 3990 /z/RHEL6A         #this will be the boot drive
   device 02C1 3390 3990 /z/RHEL6B
   device 02C2 3390 3990 /z/RHEL6C

   [manager]                            #this creates your first LAN
   name awsosa 33 --path=a0 --pathtype=OSD --tunnel_intf=y
   device 0400 osa osa
   device 0401 osa osa
   device 0402 osa osa

   name awsosa 44 --path=F1             #a second ethernet
   device 0404 osa osa
   device 0405 osa osa
   device 0406 osa osa
   ```

5. The target volume is named `RHEL6A`, and was earlier created using the zPDT **`alcckd`** command:

   ```
   $ alcckd /z/RHEL6A -d3390-10         (this has about 9 GB usable space)
   $ alcckd /z/RHEL6B -d3390-3          (a much smaller volume; optional)
   $ alcckd /z/RHEL6C -d3390-3          (a much smaller volume; optional)
   ```

   The volume file names (RHEL6A, and so on) are arbitrary, as is their location in the `/z` file system.

6. You can create a Linux alias to avoid entering `oprmsg` many times as shown in the example. Then start zPDT, perform the IPL (“boot”) from the `.iso` file, and work through

the initial installation script. The summary omits many displayed lines and concentrates on the options selected.

```
$ alias +=oprmsg
$ awsstart devRHEL6
   (some output from zPDT startup)
$ ipl_dvd ISO/generic.ins
AWSMID021I Enter 'Y' to continue or 'N' to cancel the installation
Y
   (output from the initial Linux image)
Blacklist active......
c) clear blacklist m) manual configuration r) rescan s) shell
+ c                          (select the "clear blacklist option)
Autodetection found two devices
   NUM CARD CU      CHPID TYPE DRIVER DEVICES
    1  OSA  1731/01  A0   OSD  qeth   0.0.0400,0.0.0401,0.0.0402
    2  OSA  1731/01  F1   OSD  qeth   0.0.0404,0.0.0405,0.0.0406
+ 1                          (select your tunnel ethernet connection)
Portname ....
+                            (make a null entry)
Port number......
+                            (make a null entry)
Layer mode.......
+                            (default to layer 3)
IPv4 address.......
+ 10.1.1.2                   (IP address for new Linux for z)
IPv4 netmask.......
+                            (default to 255.0.0.0)
IPv4 gateway........
+ 10.1.1.1                   (address of base Linux; the installation
source)
IPv4 DNS server
+                            (no DNS)
You may encounter problems.......
+ 1                          (to continue)
DASD range.......
+                            (default to autoprobe)
Initial configuration complete
c) continue p ) parameter file n ) network status .....
+ c                          (to continue)
Connect to 10.1.1.2 as user install
```

7. All the above text and parameters were entered in the base Linux command window where zPDT was started. *Keep this window open!* In the rest of this chapter it is referred to as the *base command window*.
8. Open another window in your base Linux and enter this command:

   ```
   $ ssh -x install@10.1.1.2
   ```

9. This command starts a "text graphics" program in the window. Select a language and select **URL** as the installation source.
10. The following window contains a field to enter the URL. The following URL was used in the example:

    ```
    ftp://ibmsys1:xxxxxxxx@10.1.1.1/ISO
    ```

In the example, *ibmsys1* is the user ID used on the base Linux and *xxxxxxx* is the password for this user ID. The base Linux is at IP address `10.1.1.1` and ISO is the directory that contains the RHEL 6.7 files.[1]

11. You are next invited to start a VNC session. Open a third command window in your base Linux and enter this command:

```
$ vncviewer 10.1.1.2:1
```

12. This commands opens a graphics window where you can select options for the next part of the installation. The following parameters and choices are involved:

    a. Select **basic storage devices**.

    b. Verify the host name that you selected in the first part of the installation process.

    c. Set the *root* password.

    d. Select the type of installation for disk management. We used the Create Custom Layout option to manually configure the disks. The `RHEL6A` disk at address `02C0` had over 7800 MB available. The following configuration was specified:

    ```
    dasda1   6800 MB  ext4  mount at /
    dasda2   1000 MB  swap
    ```

    e. Select **Development Workstation** as the installation type and select **customize now** for additional software options. In the menus provided by this option, add **FTP** to the servers and delete all virtualization options (optional).

13. After exiting from the software selection options, the Linux for z installation started. It took about 30 minutes on the notebook computer and resulted in 3.4 GB in the new root directory.

14. Select the shutdown function and wait until the process seems to complete. There is no definitive message that indicates that shutdown has completed. Then use the base command window (on the base Linux) to IPL the new Linux for z Systems:

```
$ ipl 2C0                      (the RHEL6A volumes is at address 2C0)
```

15. The new system restarts and offers a login prompt on the base command window. Normally, do not log in by using the base command window because it is necessary to enter a zPDT **`oprmsg`** command (or use an alias shortcut) for every command entered. Instead, use a different base Linux command window to start an SSH session:

```
$ ssh root@10.1.1.2
```

This command is likely to cause a conflict with the previous SSH session and you might need to remove the previous cryptographic keys for this IP address:[2]

```
$ ssh-keygen -R 10.1.1.2 -f /home.ibmsys1/.ssh/known_hosts
```

## 4.1.1 Installation status

At this point, you have a working RHEL for z Systems, with command line input. In the test environment, it was difficult to start a graphics connection. For example, the following command did not work:

```
$ vncviewer 10.1.1.2:1          (from the base Linux)
```

[1] The path that you enter here (ISO in the example) is always relative to the user ID home directory. Do not attempt to enter a path that is not based on the user ID home directory. It will not work and your installation activity will fail for no obvious reason.

[2] Or you can simple delete the `.ssh/known_hosts` file.

The first SSH login to root produced a “text graphics” window that offered the following options:

- Authentication configuration (leave as local authentication)
- Firewall configuration (disable the firewall for initial setup)
- Keyboard configuration (ignore)
- Network configuration (attempted to use, but hung the SSH session)
- RHN configuration (ignore)
- System services

### 4.1.2 Additional steps

An excellent guide to additional steps, such as establishing a graphic interface, can be found in *The Virtualization Cookbook for IBM z/VM 6.3, RHEL 6.4, and SLES 11 SP3*, SG24-8147.

### 4.1.3 Shutdown

Complete these steps to shut down:

1. Switch to root and enter a **shutdown** command to bring down the system. This process generates several messages in the base Linux command window.

   ```
   $ su                              (change to root, if not already root)
   # shutdown -h now
   ```

   In the example, the following was the last message in the base Linux command window:

   ```
   init: Re-executing /sbin/init
   ```

2. Use the **awsstop** command to terminate zPDT.

### 4.1.4 Adding additional disk volumes

Complete these steps to add additional disk volumes:

1. Create and define additional disk volumes in the devmap by using the zPDT **alcckd** command.
2. Make these volumes usable by RHEL 6.7 with the sequence of commands shown (working as *root*). In this example, the additional drives are defined as addresses `02B1` and `02B2` in the devmap.

   ```
   # cio_ignore -R                          (do not hide any devices)
   # chccwdev --online 0.0.02b0-0.0.02bf    (make range of devices online)
   # lsdasd                                 (observe names, such as dasdb)
   # dasdfmt -b 4096 -p /dev/disk/by-path/ccw-0.0.02b1  (basic DASD format)
   # fdasd -a /dev/disk/by-path/ccw-0.0.02b1     (create single partition)
   # fdisk /dev/dasdb                       (“dasdb” name from lsdasd output)
     (create partitions with the p, n, and w subcommands)
   # mke2fs -j -t ext4 /dev/dasdb1          (create Linux file system)
   # mount /dev/dasdb1 /mnt                 (or use some other mount point)
   ```

3. This example uses `/mnt` as a mount point. You would probably create something more meaningful, such as these directories:

   ```
   # mkdir /myfiles1                        (no need to be in root directory)
   # mount /dev/dsadb1 /myfiles1
   ```

4. List the `/etc/dasd.conf` file. Your new drive (`0.0.02b1` in this example) should be included in the list so that it will be recognized when Linux for z Systems is started again.

The steps described here are effective only for the running Linux session. They are not persistent over a restart.

5. You can see partition names (such as `dasda1` and `dasdb1`) with the command **`ls /dev`**. You can see device names (such as `dasda` and `dasdb`) with **`ls /dev`** or **`lsdasd`**. You can cause the drive (actually the partition) to be mounted automatically by adding the following line to `/etc/fstab`:

```
/dev/dasdb1 /mnt defaults 0 0
```

## 4.2 Installing RHEL 7.1

You can use the following steps to install RHEL 7.1:

1. Create a devmap in your base Linux home directory named `devRHEL7`:

```
[system]
memory 3000m
processors 3                        #hash symbols indicate comments

[manager]
name awsckd 22
device 02d0 3390 3990 /z/RHEL6A         #this will be the boot drive
device 02d1 3390 3990 /z/RHEL6B
device 02d2 3390 3990 /z/RHEL6C

[manager]                           #this creates your first LAN
name awsosa 33 --path=a0 --pathtype=OSD --tunnel_intf=y
device 0400 osa osa
device 0401 osa osa
device 0402 osa osa

name awsosa 44 --path=F1            #a second ethernet
device 0404 osa osa                 #use the find_io command to get path
name
device 0405 osa osa                 #you could omit this second LAN
initially
device 0406 osa osa
```

2. The target volume is named RHEL6A, and was earlier created by using these zPDT commands:

```
$ alcckd /z/RHEL7A -f4 -d3390-10     (this has about 8 GB usable space)
$ alcckd /z/RHEL7B -f4 -d3390-3      (a much smaller volume; optional)
$ alcckd /z/RHEL7C -f4 -d3390-3      (a much smaller volume; optional)
```

The volume names (`RHEL7A` and so on) are arbitrary, as is their location in the `/z` file system.

> **Tip:** RHEL7.1 is missing modules that are needed to format z Systems count key data (CKD) volumes for Linux use. There are two ways to bypass this problem. You can format the volumes (named `RHEL7A`, and so on, in the example here) by using an earlier Linux for z Systems version, or use a new function of the **`alcckd`** command shown in above. The -f4 option (new in a zPDT fix pack released in December 2015) formats the volumes for Linux for z Systems use.

3. Place the `RHEL-7.1-20150219.1-Server-s390x-dvd1.iso` file in a subdirectory named DVD in your base Linux home directory.

4. Working as user ID `ibmsys1` on your base Linux system, create a subdirectory named ISO2[3] in your base Linux directory:

```
$ cd ~                          (move to home directory /home/ibmsys1)
$ mkdir ISO2
```

5. A mount point named ISO had already been created in the same home directory. Installing RHEL7.1 involves editing one or two files in the distribution materials. However, you cannot edit a file within an `.iso` file. Copy all the files within the .iso file to normal Linux files that can be easily edited:

```
$ su                            (change to root to handle the mount)
# mount -O loop DVD/RHEL-7.1-20150219.1-Server-s390x-dvd1.iso ISO
# exit
$ cd ISO                        (location of the loop mount)
$ cp -r * ISO2                  (copy all to /home/ibmsys1/ISO2)
$ cp .discinfo ISO2             (the cp * did not get the "dot" files)
$ cp .treeinfo ISO2
```

6. Change two files to make them writable:

```
$ cd ~/ISO2
$ chmod 644 generic.ins
$ chmod 644 images/generic.prm
```

7. Using your favorite editor, change `generic.ins` as shown:

```
* minimal LPAR ins file
images/kernel.img 0x00000000
images/initrd.img 0x02000000
images/generic.prm 0x00010480             (change this line)
images/initrd.addrsize 0x00010408
```

   This change can be avoided by using the `images/genericdvd.prm` file instead of the `/images/generic.prm` file. However, the example used the `generic.prm` file for consistency with other distributions.

8. Using your favorite editor, create a file named `generic.prm` in your home directory (that is, in `/home/ibmsys1/generic.prm`):

```
ro ramdisk_size=40000
rd.znet=qeth,0.0.0400,0.0.0401,0.0.0402,layer2=1,portname=awsosa,portno=0
ip=10.1.1.2::10.1.1.1:255.255.255.0:RHEL7:enccw0.0.0400:none
inst.repo=ftp://ibmsys1:xxxxxxxx@10.1.1.1/ISO2 selinux=0
rd.dasd=0.0.02d0
```

9. In this file, the `0.0.0400`, `0.0.0401`, and `0.0.0402` parameters are the addresses of the tunnel network adapter, as defined in the devmap. `10.1.1.2` is the address for the new Linux for z Systems installation that you are building, and `10.1.1.1` is the address of the underlying PC Linux. `0.0.2d0` is the address of the DASD volume to contain the new Linux. RHEL7 is the host name for the new Linux. Your parameters can be different. You can have multiple `rd.dasd` operands to make more volumes initially visible in your new Linux. The example used a minimal set of definitions. The `selinux=0` parameter can probably be omitted.

   You might need to change or add to these parameters as you experiment with the installation process. Keep a copy of the `generic.prm` file in your home directory for convenient editing.

[3] ISO2 is an arbitrary name.

10. Copy this prm file to `ISO2/images/generic.prm`. You might want to rename the original file with this name.

11. Edit the `ISO2/images/generic.prm` file to combine all the lines into a single line with space delimiters between the parameters. Only the first line in the `generic.prm` file is processed during installation:

```
$ cd ~                                          (home directory)
$ cp generic.prm ISO2/images/generic.prm        (copy the file)
$ cd ISO2/images
$ leafpad generic.prm                           (edit the operational prm)
```

This example used an editor named **leafpad**, but any editor can be used.

12. Start zPDT with the appropriate devmap and start the installation process:

```
$ awsstart devRHEL7
   (normal zPDT startup messages)
$ ipl_dvd ISO2/generic.ins               (the DVD format is important)
 Enter Y to continue or N to cancel the installation
Y                                        (no "+" or "oprmsg" for this one)
  (Several screens of output)
Reached Target System Initialization
```

13. A long pause here followed by failure messages that indicate something is wrong with your connection to the base Linux FTP function. Is it a firewall problem? The base Linux firewall was usually disabled during these installations. Is the base Linux FTP accepting connections? Are the generic.prm parameters correct and all in a single long line?

```
Mounted Configuration File System
   (Long pause. Be patient)
...orderd data mode,,,
   (Longer pause)
...Received SIGTERM...                  (ignore this)
   (Pause)
Please ssh install@RHEL7 (10.1.1.2) to begin install.
```

14. Open another command window on the base Linux and enter the following commands:

```
$ ssh install@10.1.1.2
  (Reply to normal ssh message)
  (Some text is presented, and then:)
Select:
  1) Start VNC
  2) Use text mode
1                                  (this selects VNC)
   (You are asked for a VNC password. The example used a null.)
```

15. Open a third command window on the base Linux and start a vncviewer:

```
$ vncviewer 10.1.1.2:1                (you might use a different viewer)
```

16. This command should open a new GUI window on the base Linux. Some selections in the GUI windows have slow responses and result in warning/error messages in the second base Linux command window. Ignore those messages. In the GUI windows, work with the **`System: Installation Destination`** and **`Software Selection`** choices. The Installation Destination option manages partitioning and file system formatting. You might need a little practice to master the GUI techniques of this option. You might find it difficult to use existing partitions or to delete old partitions. We used *standard partitions* and allocated a simple system with 6800 MB with mount point root (/) and 800 MB for swap.

17. In the **Software Selection**, select **Server with GUI and FTP**.

18. Click **Start Installation** and complete the root password and user definitions on the next window. Define the user ID `ibmsys1`.[4]

19. The installation took about 30 minutes. A `Complete` message is displayed when the installation is complete. The **Reboot** button on the window does not work. You must boot your new Linux with a zPDT **`ipl`** command.

## 4.2.1 System status

You can use these steps to check the system status:

1. Switch to the first base Linux command window and boot the new Linux using this command:

   ```
   $ ipl 2d0
   ```

   This command generates startup messages, ending with an RHEL7 login prompt. In the example, this prompt was followed with multiple instances of these messages:

   ```
   A start job is running for Crash recovery kernel arming
   A start job is running for dev-ttyS0.device
   ```

2. These messages stopped after about 100 repetitions and seemed to have no side effects. Open another command window on the base Linux and use **`ssh`** to log in to Linux for z Systems:

   ```
   $ ssh ibmsys1@10.1.1.2
   ```

   The example was installed in a special test environment without access to the Internet and without proper registration with Red Hat. In this case postinstallation tasks were not completed. You (as a proper RHEL user) can discuss registration procedures and additional installation steps with your Red Hat representation. From this point onward, there should be little that is unique related to the zPDT environment.

This book is intended to address only issues related to use with zPDT. For more information about additional installation steps for RHEL 7, see *The Virtualization Cookbook for IBM z Systems Volume 2: Red Hat Enterprise Linux 7.1 Servers*, SG24-8303. The registration process is described in section 3.1 of this book. Additional material can be found in *The Virtualization Cookbook for IBM z/VM 6.3, RHEL 6.4, and SLES 11 SP3*, SG24-8147.

[4] The example also uses user ID `ibmsys1` on the base Linux system. There is no need to continue using this user ID, but it is simpler to do so.

5

# General topics

Linux for z Systems, whether based on Red Hat or Novell, is a normal Linux system and all general Linux administrative procedures apply. However, a few considerations that are unique to the zPDT environment are described in this chapter.

This chapter includes the following sections:

- Memory usage
- Disk backups
- zPDT updates
- Base Linux updates
- Disk addresses

 

## 5.1 Memory usage

A key factor in Linux performance is efficient operation of its disk cache. This book assumes that you are familiar with this topic. As a general statement, Linux uses much of the system real memory as a disk cache.[1] zPDT, running on the *base Linux system*, presents a unique situation because the base system also has a disk cache.

You are likely to have a situation where the same data is being cached in both the Linux for z Systems disk cache and the base Linux disk cache. This situation is wasteful of real memory that is better used to cache more data, although it does not introduce errors.

As part of using zPDT, you must create a device map (devmap). A statement in the devmap specifies the size of the virtual z Systems you want to use:

```
[system]
memory 3000m
processors 3                              #hash symbols indicate comments
```

In this example, the virtual z Systems machine where Linux for z Systems will run has 3000 MB, or about 3 GB memory. This 3 GB exists, through the zPDT implementation, as shared virtual memory in the base Linux. The Linux for z Systems kernel, and various malloc and stack data, consume some of this memory, and much of the rest is used as a disk cache. Assume the base PC that you are using has 16 GB memory. The base Linux kernel, stacks, and the shared virtual memory used to create the virtual z Systems consume some it, for example, 5 GB at any time. This configuration leaves about 11 GB that the base Linux can use as a disk cache, which is sufficient for excellent performance in most cases.

If you then increase the devmap specification (for Linux for z Systems) to 15 GB, and if the processing of whatever is running in Linux for z Systems causes all its memory to be active (that is, in real memory), then only a small mount of real memory is left for the base Linux disk cache. This situation can cause performance problems, depending on what else is running in the base Linux system.

It is difficult to make specific recommendations for setting the memory size for zPDT operation. In lightly loaded systems, make the virtual z Systems memory relatively modest (for example 3 GB) and leave most of the PC real memory for use by the base Linux. Depending on the nature of the workload intended for Linux for z, this plan might or might not be reasonable.

## 5.2 Disk backups

Everyone should back up their systems. On a real z Systems, this can involve tape drives or special programs to completely save (“dump”) disk volumes. This topic is much simpler on a zPDT system. All the Linux for z volumes (“drives”) exist as simple files on the base Linux system. An emulated 3390-3 drive, for example, is a simple 2.8 GB file on the base Linux.

You can back up these files (when your Linux for z and zPDT are not running) by copying the files on the base Linux system. Or you can first create a compressed copy and back that up.

Remember that emulated 3390 volumes are created with the **alcckd** command:

```
$ alcckd /z/SLES23 -d3390-12
```

[1] There are complexities behind this general statement that can be largely ignored in a general description.

This example creates a file 12*1113 cylinders large, or roughly 10 GB. You can place the file anywhere in your base Linux file systems. The example used a file system mounted at /z, but there is nothing special about this configuration.

You would include the file (/z/SLES23) in your devmaps for use by your Linux for z Systems installation. You would format it, partition it, create file systems in it, and probably place data in it. After stopping Linux for z Systems and stopping zPDT, back it up using one of these commands:

```
$ cp /z/SLES23 /run/media/ibmsys1/USBdrive/SLES23      (simple copy)
```

or

```
$ gzip -c /z/SLES23 > /abc/backups/S23.gz              (compressed copy)
```

You can store the backup copies on your hard disks, on a USB flash drive, or burn them on DVDs. If the backup is on a hard disk, you can use it directly by changing your devmap to point to the backup. Or you can copy (and decompress if needed) the backup copy over the original copy.

Always make backup copies when zPDT is not active to avoid any chance of a corrupted backup due to changed data not being completely flushed to disk.

## 5.3 zPDT updates

There is typically a new zPDT release once each year, although they are not released on a specific schedule. Typically each zPDT release will have two or three fix packs. The releases of fix packs are installed as described in 2.3, “zPDT installation” on page 10. Installing these updates involves little risk, so install the new releases and updates as they become available. In the unlikely event of a problem, an older release can be easily installed.

The installer program for zPDT automatically removes any existing release and replaces it with the one being applied.

## 5.4 Base Linux updates

The base Linux is a normal PC Linux, either SUSE or Red Hat based. Updates are often released for these Linux distributions (usually “online updates”). Whether you install these updates is up to you. In the early days of zPDT development, online updates were seldom installed. However, in the last few years they have been routinely installed, which has not caused any serious problems.

However, confine your base Linux distribution levels to those mentioned in the most recent editions of the main zPDT documentation (*IBM zPDT Guide and Reference: System z Personal Development Tool*, SG24-8205). The zPDT developers do not “chase” the latest PC Linux distributions and you can sometimes encounter problems when using the latest distributions.

## 5.5 Disk addresses

Every z Systems disk drive that is defined to the system has a device number, commonly known as an address, for example `2A0`. Every additional drive (or "volume") has a different address, for example `2A1`, `2A2`, `1000`, and `12FA`. Simple sequential addresses such as `2A0`, `2A1`, `2A2` are used in this book to avoid confusion.)

Linux has disk addresses and partition addresses. These addresses can be in several different formats. The examples use the `dasda`, `dasda1` format. In this case, `dasda` is the disk drive, and `dasda1` is a partition on the drive. A second partition, if it exists, would be `dasda2`, and so on. Another disk drive might be `dasdb`, and third drive might be `dasdc`, and so on. Generally speaking, the IPL (boot) volume is `dasda`.

There is a mapping issue involved in these two sets of names, as shown in Figure 5-1.

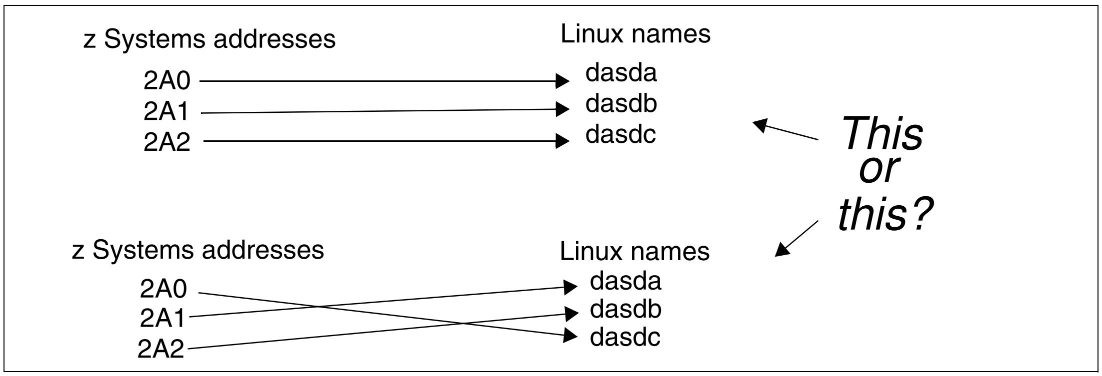


*Figure 5-1 Address and name mapping*

This mapping is not normally an issue on large z Systems mainframes because the disks (in the z Systems view) generally have fixed addresses that seldom change. zPDT presents more of a challenge because disk addresses can be easily changed by altering a devmap.[2] This limitation could create great confusion when your Linux for z Systems `/etc/fstab` attempts to mount file systems in partitions on the disks.

An authoritative description of how this mapping is handled was not found. By informal experimentation, the file `/etc/dasd.conf` was found to be an important control point. Apparently the IPL device is always `dasda`. The remaining volumes are mounted as `dasdb`, `dasdc`, and so on, in the order they are listed in `/etc/dasd.conf`.

[2] You can have multiple devmaps, although only one can be active with a zPDT instance at any given time. The multiple devmaps might refer to the same emulated volumes with different addresses in different orders.

# Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

## IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.


- *IBM zPDT Guide and Reference: System z Personal Development Tool*, SG24-8205
- *Sysplex Add-On for zPDT,* SG24-8315
- *The Virtualization Cookbook for IBM z Systems Volume 1: IBM z/VM 6.3*, SG24-8147
- *The Virtualization Cookbook for IBM z Systems Volume 2: Red Hat Enterprise Linux 7.1 Servers*, SG24-8303


## Online resources

These websites are also relevant as further information sources:


- *Red Hat Enterprise Linux 7 Installation Guide*

  `https://access.redhat.com/documentation/en/red-hat-enterprise-linux/7/`


 

(0.1”spine)
0.1”<->0.169”
53<->89 pages

SG24-8330-00

ISBN 0738441546

Printed in U.S.A.