

Part I

# *Linux+ Study Guide*

*A guide and template to study for the Linux+  
LXO-103 exam.*

Produced by

**PHOENIX**TS

# *Table of Contents*

- Introduction
- System Architecture
  - Determining and Configuring Hardware Settings
  - Booting the System
  - Runlevels/Boot Targets and Shutting Down the System
- Linux Installation and Package Management (18%)
  - Design and Configure the Hard Disk Layout
  - Install the Boot Manager
  - Manage Shared Libraries
  - Debian Package Management
  - RPM and YUM Package Management
- GNU and UNIX Commands (43%)
  - The Command-Line
  - Processing Text Streams and Filters
  - File Management
  - Streams, Pipes, and Redirects
  - Create, Monitor and Kill Processes

# *Table of Contents...*

## *(Continued)*

- Modify Process Priority
- Search Text Files with Regular Expressions
- Edit Files with Vi
- Devices, Linux Filesystem, and Filesystem Hierarchy
  - Create Partitions and Filesystem
  - Maintain Filesystem Integrity
  - Mount and Unmount Filesystems
  - Manage Disk Quotas
  - File Permission and Ownership
  - Hard and Symbolic Links
  - Finding and Moving Files
- Additional Resources
  - Man, Help, and Info Pages
  - Open Source Online Community
  - Anki Decks
  - Linux Commands Cheat Sheet

# Introduction

## *How to Interpret and Use the Guide*

This guide provides a template for a personalized study guide for Linux+ certification candidates. The guide follows the published CompTIA LX0-103 exam objectives and subsections. Each objective includes subsections with important commands, files and directories, concepts, terms, challenge questions, and additional learning resources.

### ***How to Engage ~***

Use the document to guide your studies. Read, practice, and make notes within the document (adjacent space provided). Nothing replaces the learning experience of creating your own notes for studying. Research even goes far enough to suggest that [handwritten notes](#) are better for learning than typing.

Also, if you know a command, but don't know how to use it, refer to the manual (man page) entry by entering "man <command>". If there's no man page, try the help command.

### ***Caveat ~***

This document does not stand as an all-inclusive resource for preparing for the exam and does not replace and/or compensate for training or experience.



**Exam Objective**

# *System Architecture*



# Determine and Configure Hardware Settings

You should be capable of accomplishing the tasks of:

- partitioning disks
- enabling and disabling peripheral devices
- configuring systems with or without external peripherals

You should have an understanding of the differences between internal and external mass storage devices. You should have familiarity with the location of configuration files for different devices. You should understand how the Linux system treats different hotplug and coldplug devices.

## **Commands:**

modprobe

lsmod

modinfo

lspci

lsusb

setpci

insmod

rmmod

usbmgr

hotplug

## **Files and Directories:**

/dev  
/etc/udev  
/proc  
/proc/interrupts  
/proc/dma  
/proc/bus/usb  
/sys  
/etc/modprobe.conf  
/etc/hotplug  
/etc/hotplug/usb.usermap

## **Concepts and Terms:**

Coldplug and hotplug devices, Central Processing Unit (CPU), Random Access Memory (RAM), Peripheral Component Interconnect (PCI), I/O addresses, Direct Memory Addressing (DMA), Cylinder/Head/Sector (CHS) Geometry, SCSI disks, PATA disks, ATA disks, CHS Geometry translation, logical block addressing, sysfs, HAL daemon, dbus, udev, kernel modules

## **Challenge Questions:**

1. Why would you remove kernel modules?
2. What are the differences between the USB 1.0, 1.1, 2.0, and 3.0 device types?
3. What external devices does Linux treat as SCSI devices?

# Booting the System

You should understand the entire boot process, the hardware and software involved, how to discover and interpret boot messages in case of booting issues. You should also know the differences, pros, and cons between initialization systems of Unix System V (SysV), Systemd, and Upstart.

## **Commands:**

dmesg  
systemctl

## **Files and Directories:**

/var/log/dmesg  
/var/log/messages  
/var/log/syslog  
/var/log  
/var/log/boot  
/dev/console  
/sbin/init



## **Concepts and Terms:**

Systemd, Upstart, SysV, System logger (syslog), BIOS, UEFI, kernel, initramfs, Sysvinit, init

## **Challenge Questions:**

1. Explain the boot process and sequences starting from powering on to boot completion.
2. How would you go about finding and reading boot messages?

# Runlevels/Boot Targets and Shutting Down the System

Understand the purposes for the default runlevels 0 – 6 for SysV initialization. Know if you should and how to change the default runlevels. Understand how to change runlevels or boot targets in single user mode.

## **Commands:**

chkconfig  
update-rc.d  
rc-update  
runlevel  
shutdown  
halt  
wall  
reboot  
poweroff  
systemctl

## **Files and Directories:**

/etc/inittab

/etc/rc.d

/etc/init.d

/etc/rc.d/init.d

/etc/systemd

/usr/lib/system

/lib/systemd/system

## **Concepts and Terms:**

Default runlevels, SysV, systemd, Upstart init, telinit

## **Challenge Questions:**

1. What are the default runlevels used for (runlevel 1-6)?
2. What are the default runlevels of 0 and 6 normally reconfigured for?
3. How does the systemd initialization process differ from SysV or Upstart?
4. How would you alert system users before switching runlevels/boot targets?



**Exam Objective**

*Linux Installation  
and Package  
Management*



# Design and Configure the Hard Disk Layout

Know how to make and assign filesystems and swap space to partitions or disks. Make sure the /boot partition follows the hardware architecture.

## **Concepts and Terms:**

/root, /var, /boot, LVM, swap space, mount points, partitions, multiple OS support

# Install the Boot Manager

You should be capable of demonstrating proficiency in installing and configuring boot loaders, such as GRUB Legacy or GRUB 2, configuring alternate boot locations and backup boot options, explaining the pros and cons of UEFI booting (secure boot as well).

## **Commands:**

```
grub-install  
grub-mkconfig (grub2-mkconfig)  
update-grub  
Debugfs  
dump2fs
```

## **Files and Directories:**

```
EFI/boot/bootx64.efi  
/boot/efi  
/boot/grub/menu.list  
/grub.conf  
/boot/grub/device.map  
/boot/grub/grub.cfg  
/boot/grub2, /etc/grub.d, or /etc/default/grub
```

## **Concepts and Terms:**

GRUB Legacy, GRUB 2, Master Boot Record (MBR), BIOS boot loader, GUID Partition Table (GPT), LILO, Syslinux, EFI boot loader, EFI system partition (ESP), Superblock, initial RAM disk (initrd), rootnoverify, chainloading, efibootmgr, secure boot, System logger (syslog)

## **Challenge Questions:**

1. What the secure boot options for the UEFI boot loader?
2. What are the differences between GRUB legacy and GRUB 2 (configuration, scripts, tools, loadable modules)?

## **Additional Resources:**

- “UEFI vs BIOS Legacy Booting: What’s the Difference?” - <http://phoenixts.com/blog/uefi-vs-legacy-bios/>
- “Bootting into UEFI Mode: The Good, the Bad, and the Ugly” - <http://phoenixts.com/blog/bootting-uefi-mode/>

# Manage Shared Libraries

Linux relies on shared libraries, especially for reducing file size and the amount of shared memory consumed on the system. Understand the difference between shared libraries (.so file extension) and static libraries (.a extension). Know why identifying libraries plays a role in package management. You should demonstrate proficiency in:

1. Identifying and loading shared libraries.
2. Practicing with the library management commands, ldd and ldconfig. Although ldconfig run automatically when installing or updating RPM and Debian packages (even when compiling from source).
3. Identifying library chains with the ldd command. See what libraries depend on other libraries. Practice employing this command enables you the ability to understand how to identify missing libraries.

## **Commands:**

Ldd

ldconfig



## **Files and Directories:**

/etc/ld.so.conf

/lib

/usr/lib

/etc/ld.so.cache

## **Concepts and Term:**

C library (libc), GIMP Tool Kit (GTK+), Qt, GNU C Library (glibc),  
LD\_LIBRARY\_PATH

## **Challenge Questions:**

1. How do you temporarily set the library path to test a new shared library before using it for all programs?
2. Why may programs fail to locate libraries? What tools and methods may resolve this issue?

# Debian Package Management

You should understand how to obtain, build, install, and manage Debian packages from source. Also, know the meta-packagers that simplify package management, such as Aptitude, synaptic, and dselect. Know the relevant configuration files and other important files for updating and maintaining package repositories. You should be capable of demonstrating proficiency in:

1. Downloading, installing, upgrading, checking, updating software packages and dependencies for Debian (.deb) from source code.

## **Commands:**

dpkg, apt-cache  
apt-get  
Alien  
dpkg-reconfigure

## **Files and Directories:**

/etc/apt/sources.list  
/etc/dpkg/dpkg.cfg  
/var/lib/dpkg

## **Concepts and Terms:**

Synaptic, Dselect, Aptitude

## **Challenge Questions:**

1. How do you convert an RPM package to Debian or tarballs? What issues may arise in the conversion process?
2. If an attempt at a custom configuration for a Debian package goes awry, what dpkg command option will reset the package back to the initial installation?
3. What online resources or repositories are available for locating Debian source packages?
4. What common package dependency and conflicts may occur? How may you resolve these conflicts? Hint – missing libraries or support programs.

## **Additional Resources:**

- <http://www.debian.org/distrib/packages>

# RPM and YUM Package Management

Although the meta-packagers (YUM, Aptitude) simplify the tasks for managing software packages for most major Linux distributions, it is critical to know how to build and maintain packages from source code. RPM packages (.rpm) work best on Red Hat based distros. Therefore gaining proficiency in using the RPM and YUM command sets involves working either on Red Hat, Fedora, or CentOS systems. You should be capable of demonstrating proficiency in:

1. Understanding the elements of the RPM package names, such as the CPU architecture, build and version numbers in a package such as this – `packagename-a.b.b-x.architecture.rpm`
2. Using the RPM and YUM command sets and critical options available to view info about packages, install, re-install, update, and remove packages.

## **Commands:**

`rpm`

`rpm2cpio`

`yum`

`Alien`

`rpmbuild-rebuild`

## **Files and Directories:**

/usr/lib/rpm/rpmrc

/etc/rpmrc

/etc/yum.conf

/etc/yum.repos.d

## **Concepts and Terms:**

Yumdownloader, YUM, RPM

## **Challenge Questions:**

1. What RPM compatibility issues may arise and what solutions exist? For example, a RPM package designed for one Linux distribution may require different dependencies for another distribution.
2. How do RPMs compare to tarballs for Linux and other operating systems?
3. List several reasons for building and maintaining software packages from source code.
4. What are the pros and cons of using a meta-packager, such as YUM, over RPM?

## **Additional Resources:**

- <http://freshrpms.net/>
- <http://www.rpm.org/>
- <http://yum.baseurl.org/>



**Exam Objective**

*GNU and UNIX  
Commands*



# The Command Line

Although the meta-packagers (YUM, Aptitude) simplify the tasks for managing software packages for most major Linux distributions, it is critical to know how to build and maintain packages from source code. RPM packages (.rpm) work best on Red Hat based distros. Therefore gaining proficiency in using the RPM and YUM command sets involves working either on Red Hat, Fedora, or CentOS systems. You should be capable of demonstrating proficiency in:

1. Understanding the elements of the RPM package names, such as the CPU architecture, build and version numbers in a package such as this – `packagename-a.b.b-x.architecture.rpm`
2. Using the RPM and YUM command sets and critical options available to view info about packages, install, re-install, update, and remove packages.

## **Commands:**

`rpm`

`rpm2cpio`

`yum`

`Alien`

`rpmbuild-rebuild`

## **Files and Directories:**

`/.bash_history`  
`/etc/environment`  
`/etc/profile.d, /etc/profile`  
`/etc/default/locale`  
`/etc/bash.bashrc`

## **Concepts and Terms:**

bash, sh, tcsh, csh, ksh, zsh shells, default interactive shell , default system shell, man, info, and help pages, environment variables, \$PATH, \$EDITOR

## **Challenge Questions:**

1. How do you permanently change an environment variable?
2. How do you change the default text editor your system?



# Process Text Streams with Filters

Effective sysadmins come to understand the importance of commands for combining files, transforming data, formatting text, displaying text, and summarizing text with the commands listed below. You should be able to demonstrate proficiency in:

1. Sending text and output streams through filter tools to modify output with standard UNIX commands found in the GNU textutils package (listed below).

## **Commands:**

cat

cut

expand

join

tac

paste

sed

unexpand

od

sort

split

tr  
uniq  
mft  
nl  
pr  
head  
tail  
less  
more  
wc

### **Challenge Questions:**

1. Why would you use the `expand` or `unexpand` commands to convert tabs to spaces (vice versa) for text files?
2. What file viewing command is helpful for monitoring log files as they actively grow?

# File Management

You should understand how to create, move, copy, and remove files and directories. Know file naming best practices, dot files, and special file names. You should be able to demonstrate proficiency with the main file management commands (including their common options), such as `ls`, `rm`, `mkdir`, `rmdir`, `touch`, `cp`, and `mv`. You have familiarity with the different archiving and compression tools.

## **Commands:**

`tar`

`cpio`

`xz`

`gzip`

`gunzip2`

`bzip2`

`mv`

`ls`

`rm`

`mkdir`

`rmdir`

`touch`

`cp`

`dd`

## **Concepts and Terms:**

file globbing (also called filename expansion), wildcard expansion rules, Linux file naming rules and conventions, dot files, special filenames,

## **Challenge Questions:**

1. How do you recursively copy and delete entire directories?
2. What three types of timestamps to Linux files have?

# Streams, Pipes, and Redirects

Students should gain proficiency redirecting Standard Input (STDIN), Standard Output (STDOUT), and Standard Error (STDERR). Know the role of file descriptors in redirection. Also, understand why streams, redirects, and pipes are vital to system administration in Linux. These tools and methods adhere to the UNIX philosophy of performing complex actions with simple tools. You should have proficiency in:

1. Redirecting standard input, standard output, and standard error.
2. Using the redirection operators.
3. Piping output of one command as input to another command

## **Commands:**

tee  
xargs  
grep  
find  
locate

## **Files and Directories:**

/dev/null

## **Concepts and Terms:**

Standard Output, Standard Input, Standard Error, File Descriptors, Redirection Operators, Data Pipelines

## **Challenge Questions:**

1. What are the differences between Standard Input (STDIN), Standard Output (STDOUT), and Standard Error (STDERR)?
2. Why would you redirect standard output or error to /dev/null?
3. Why is grep used often with pipes?
4. Even though you can use backticks ( ` ) to enclose separate commands, why is this not a good practice?

## **Additional Resources:**

1. Grep Examples - [http://tldp.org/LDP/Bash-Beginners-Guide/html/sect\\_04\\_02.html](http://tldp.org/LDP/Bash-Beginners-Guide/html/sect_04_02.html)
2. Piping Fundamentals - <http://linuxcommand.org/lts0060.php>

# Create, Monitor, and Kill Processes

Process and job management plays a vital role in the system administrator's job. Understanding how to manage and analyze active system processes provides critical for system performance. You should have proficiency in:

1. Launching and moving jobs in the foreground and background from the terminal with the `fg` and `bg` commands.
2. Monitoring active system processes with the `top` command.
3. Gathering and identifying relevant process information (normal vs abnormal) with relevant commands.

## Commands:

`uname`

`ps`

`top`

`fg`

`bg`

`jobs`

`pgrep`

`screen`

`nohup`

`free`

`uptime`

`kill, killall, pkill`

`ampersand (&)`

## **Concepts and Terms:**

PS\_PERSONALITY

## **Challenge Questions:**

1. What information does the ps and top commands provide? How do these commands differ?
2. Why would you pipe output from the ps command as input to the grep command?



# Modify Processes

You should have proficiency in:

1. Identifying processes and relevant information (usernames/owners, PID, group ID, and priority value).
2. Reassigning process priority values.
3. Send signals to processes

## Commands:

`nice`

`renice`

`ps`

`top`

## Challenge Questions:

4. What is the numeric range of priority values?
5. Are ordinary users permitted to edit processes with negative priority values?
6. What is the default value for a program run with `nice`?

# Search Text Files with Regular Expressions

You should become familiar and proficient with employing regular expressions in combination with the `grep` and `sed` commands. Understand the various notational elements in regular expressions, such as repetition operators, range and bracket expressions, carats, dots, parentheses, and escape characters. Also, know the various options for the `sed` and `grep` commands, such as `rgrep`, `fgrep`, `pgrep`, and `egrep`.

## **Commands:**

`grep`

`rgrep`

`fgrep`

`rgrep`

`pgrep`

`sed`

## **Concepts and Terms:**

Bracket Expressions [], Basic and Extended Regular Expressions, Range Expressions, dot (.), caret (^), Repetition operators (\*, +, ?), Multiple possible strings (|), Parenthesis for subexpressions, grep syntax and options, sed syntax and options

# Search Text Files with Regular Expressions

Gain familiarity and proficiency with the core Vi editing commands, switching between different modes, and navigating text documents. Although you can change the default editor from Vi to an editor that is easier to use, it is still beneficial to understand how to use Vi, especially for editing specific configuration files in emergency boot situations. Your best bet is to practice using Vi for the exam and to save a cheat sheet for those emergency boot and recovery situations.

## Concepts and Terms:

Command mode, insert mode, ex mode. Vim

## Challenge Questions:

1. How do you change the system default text editor?

## Additional Resources:

- Vi Cheat Sheet - [http://www.atmos.albany.edu/daes/atmclasses/atm350/vi\\_cheat\\_sheet.pdf](http://www.atmos.albany.edu/daes/atmclasses/atm350/vi_cheat_sheet.pdf)

# 4

**Exam Objective**

## *Devices, Linux Filesystem, and Filesystem Hierarchy*



# Create Partitions and Filesystem

You should understand how to manage the MBR and GPT partitioning systems. You should understand how to create filesystems, such as ext2, ext3, ext4, jfs, XFS, NTFS, syslinux, btrfs, Reiserfs, UDF, and VFAT. Know the differences between the common filesystems and the benefits of journaling. Understand the process and tools for formatting partitions.

## **Commands:**

mkfs  
mkswap  
swapon  
fdisk  
gdisk  
parted  
pvcreate  
lvscan  
mkdosfs

## **Files and Directories:**

/dev/mapper  
/etc/fstab

## **Concepts and Terms:**

MBR, GPT, LVM, swap space, extended partitions, logical partitions, boot partition, primary partitions, Reiserfs, btrfs, gdisk, fdisk GNU parted, Gparted

## **Challenge Questions:**

1. What are the pros and cons of using Logical Volume Management (LVM)?
2. Why does Linux use swap space?

# Maintain Filesystem Integrity

Understand the best practices, how to use specific tools, and monitor critical information for tuning and maintaining filesystem health. Tasks include the ability to monitor free space and nodes, verify filesystem integrity, and repair common filesystem issues such as data congestion, buggy utilities and drivers, and hardware errors.

## **Commands:**

du  
df  
fsck  
e2fsck  
mke2fs  
debugfs  
dumpe2fs  
tune2fs  
xfs\_metadump  
xfs\_info



xfs\_admin  
xfs\_db  
xfs\_check  
xfs\_repair

### **Challenge Questions:**

1. What is the first critical step to often take before tuning, debugging, or editing the filesystem? (hint – to mount or not to mount)
2. Explain the purpose and benefits of journaling, especially for filesystem maintenance.
3. What percentage of used filesystem space should you consider cleaning the filesystem? This answer does depend on administrator preference, but we are looking for the “best practice” answer.

# Mount and Unmount Filesystems

As an admin, you should know how to configure filesystem to mount automatically at boot up. For many reasons, specifically filesystem maintenance and tuning, you should know how to manually mount and unmount the filesystem.

## **Commands:**

mount  
umount  
fsck  
umask  
UID  
dump

## **Files:**

/etc/fstab

/mnt

/media

/usr/src/linux/documentation/filesystems (or /usr/src/linux/fs/*fsname*)

## **Challenge Questions:**

1. What is the process for configuring the filesystem to mount when the system boots?
2. What mount point do you mount and unmount removable media? Is this process always necessary?

# Mount and Unmount Filesystems

Admins should understand the benefits of setting disk quotas for users and groups. They should know how to create, edit, and check quotas for users and groups with relevant commands and through editing specific configuration files, such as `/etc/fstab` and `/etc/quotatab`.

## **Commands:**

`edquota`, `repquota`, `quotaon`, `quotacheck`

## **Files and Directories:**

`/etc/fstab`, `/etc/quotatab`

## **Concepts and Terms:**

Quota v1 support system, Quota v2 support system, `usrquota` and `grpquota` options

## **Challenge Questions:**

1. Explain the process of enabling and setting disk quota support for users and groups in the `/etc/fstab`.

# File Permissions and Ownership

Admins must be capable of having a strong grasp on the best practices for setting and modifying file and directory permissions for users and groups. Without a comprehensive understanding of the tools, methods, and possibilities for permission management, various security risks increase.

## **Commands:**

umask  
chown  
chgrp  
chmod  
newgrp

## **Files and Directories:**

/etc/profile

## **Concepts and Terms:**

permission bits, sticky bits, special permission bits, SUID, SGID, Discretionary Access Control Model (DAC), Access Control List (ACL), Mandatory Access Control (MAC), Role-Based Access Control (RBAC), default permissions, user mask

## **Challenge Questions:**

1. How do SUID and SGID create potential security risks?
2. Does read/write access to a directory give the user the right to change inode structures?
3. What is the difference between symbolic and octal permissions? When is the best time to use each?

# Hard and Symbolic Links

You should know the differences, advantages, and disadvantages of hard and symbolic links, including the common uses of each type of link. You should know how to create and remove links. Understand how to copy and link files.

## **Commands:**

ln

ls

## **Concepts and Terms:**

Hard links, symbolic links

# Find and Move Files

You should be able to demonstrate proficiency in using the find and locate tools. Also, know the advantages and disadvantages of using both search tools. You should have a strong grasp on the layout of important directories according to the Filesystem Hierarchy Standard.

## **Commands:**

find

locate

whereis

which

updatedb

type

## **Files:**

/etc/updatedb.conf



## **Concepts and Terms:**

Filesystem Hierarchy Standard (FHS), shareable and unshareable files, static and variable files, /root, /boot, /etc, /bin, /sbin, /lib, /usr, /usr/local, /usr/share/man, /usr/X11R6, /opt, /home, /var, /mnt, /media, /dev, /proc

## **Challenge Questions:**

1. What directories would you consider placing on a separate partition?
2. What directories should never be placed on a separate partition?
3. How does the FHS extend and fix the limitations of the original Filesystem standard (FSSTND)?

## **Additional Resources:**

4. The Filesystem Hierarchy System (FHS) - <http://www.pathname.com/fhs/pub/fhs-2.3.pdf>

# Additional Resources

*Read, Practice, Repeat.*

**Putting it all together** ~ Install an Arch Linux system to apply what you learned about Linux installation and filesystems with this helpful guide – [Full Arch Linux Installation in 29 Tweets](#)

**YouTube** ~ The article, "[9 Youtube Channels to Learn Linux Online](#)" outlines great YouTube channels for learning about Linux administration.

**Live on the Command Line** ~ The most effective way to gaining proficiency with the topics, concepts, commands, and filesystem involves practicing on a Linux command line. Navigate files, change directories, launch and close GUI applications, even use [Linux text-based web browsers](#).

# *Linux+ Training Prep*

Reinforce the learning process by training with an experienced instructor in a CompTIA Linux+ certification class.

[LEARN MORE](#)