



LPIC-1 Exam 101

Study Sheet

Table of Contents

Introduction	3
Study Sheet Conventions	3
Topic 101: System Architecture	4
101.1 Determine and Configure Hardware Settings	4
101.2 Boot the System	4
101.3 Change Runlevels and Shutdown or Reboot the System	5
Topic 102: Linux Installation and Package Management	7
102.1 Design Hard Disk Layout.....	7
102.2 Install a Boot Manager	7
102.3 Manage Shared Libraries.....	7
102.4 Use Debian Package Management.....	8
102.5 Use RPM and YUM Package Management.....	8
Topic 103: GNU and Unix Commands	10
103.1 Work on the Command Line	10
103.2 Process Text Streams Using Filters.....	11
103.3 Perform Basic File Management	13
103.4 Use Streams, Pipes and Redirects	14
103.5 Create, Monitor and Kill Processes	15
103.6 Modify Process Execution Priorities	16
103.7 Search Text Files Using Regular Expressions	16
103.8 Perform Basic File Editing Operations Using Vi	17
Topic 104: Devices, Linux Filesystems, Filesystem Hierarchy Standard	18
104.1 Create Partitions and Filesystems	18
104.2 Maintain the Integrity of Filesystems.....	18
104.3 Control Mounting and Unmounting of Filesystems	19
104.4 Manage Disk Quotas	20
104.5 Manage File Permissions and Ownership	20
104.6 Create and Change Hard and Symbolic Links	21
104.7 Find System Files and Place Files in the Correct Location	21
2015 Objective Updates	25
Upstart	25

Systemd.....	25
Systemd Commands And Directories.....	26
Managing MBR and GPT Partitions	27
Commands and Compression	27
GRUB2	28



Introduction

This study sheet is intended to help you do a targeted review of key topics that are very likely to be covered during your LX0-101 LPIC exam. This is not intended to replace any quizzed, labs or lectures here at Linux Academy, but can be used as a final review just prior to your exam.

Study Sheet Conventions

This follows the LPIC syllabus for Exam 101 exactly. Each section contains the full LPIC designation, section title and description and the weighting you can expect for these topics during the exam.

There are more topics that could potentially be covered than are listed in this review material. However, the items on this study sheet are the most common topics tested in each section based on community feedback and experience. Please do take the time to review all the topics from the formal syllabus available at <http://www.lpi.org>.



Topic 101: System Architecture

101.1 Determine and Configure Hardware Settings

Weight: 2

Description: Candidate should be able to determine and configure fundamental system hardware.

- modprobe
 - automatically loads (or unloads) any dependent modules and is usually the preferred way of loading kernel modules (replacing the deprecated ‘insmod’ and ‘rmmod’ utilities)
- lsusb
 - a utility for displaying the information about USB buses and associated connected devices in the system
 - by default, reads the /dev/bus/usb directory for contents and configurations
- /proc
 - Contains virtual system information
 - /proc/interrupts shows the IRQs in use on the system
 - /proc/dma displays DMA addresses in use on the system
 - DMAs are an alternate method of communicating with I/O ports
- lsmod
 - list modules currently loaded by the running Linux kernel
- insmod
 - older method used to insert modules into the running Linux kernel
- rmmod
 - older method used to remove modules from the running Linux kernel

101.2 Boot the System

Weight: 3

Description: Candidates should be able to guide the system through the booting process.

- dmesg
 - extracts information about the boot process and can be viewed with the command directly
- kernel
 - this is the main part of the Linux operating system, it is responsible for the entire OS, threads, devices, filesystems and video
- BIOS
 - Basic Input Output System
 - responsible for preparing the system to boot, finding the CPU, memory and disk in order to begin the boot process

- **Boot Process**
 1. CPU starts and then runs the BIOS.
 2. BIOS checks for hardware, configuring hardware and looking for a boot sector.
 - Newer systems can use EFI/UEFI instead of BIOS
 3. Boot loader (grub) takes over for the bios.
 4. Find a kernel, load it into memory and execute it.
 5. Kernel takes over, performs systems tasks, and loads the final process of executing the initial program for your system. By default, this is the program /sbin/init.
 6. The initial program gets the process ID (pid) of 1.
- **/var/log/messages**
 - Contains detailed post boot messages for your system as well as messages/errors/information from applications that communicate with your kernel, kernel modules (drivers, etc) – often referred to as the contained of “global system messages”
- **dmesg**
 - /var/log/dmesg: on some Linux distributions
 - Contains “kernel ring buffer information”
 - prints number of messages on the screen that display information about the hardware devices that the kernel detected during the boot process

101.3 Change Runlevels and Shutdown or Reboot the System

Weight: 3

Description: Candidates should be able to manage the runlevel of the system. This objective includes changing to single user mode, shutdown or rebooting the system. Candidates should be able to alert users before switching run level and properly terminate processes. This objective also includes setting the default run level. It also includes basic feature knowledge of potential replacements to init.

- **/etc/inittab**
 - Run level and default run level configuration is set by the contents of this file on sysvinit systems
 - Pointer to systemd start configuration in /lib/systemd/system/<target name>.target
- **shutdown**
 - Can be used to power off the machine, but can notify users before doing so for a set period of time
 - -k: send warning of impending shutdown but do not shutdown
 - -r: reboot after shutdown
 - -h: halt or power off the system
 - -P: half action to turn off power

- -f: skip fsck on next boot
- -F: force fsck on next boot
- -c: cancel a waiting shutdown
- -t: tell init to wait XX seconds before shutting down
- init
 - can be used to change run level and shutdown a machine
 - the last process run in the boot process
- run levels
 - 0 - Used to shift the system from one state to another, used to shut down and will power off the system
 - 1, s or S - Single-user mode. Typically used for low-level system maintenance that may be impaired by normal system operation such as resizing partitions.
 - 2 - On Debian and its derivatives, it's a full multi-user mode with X running and a graphical login. On all other distributions, the run level is undefined.
 - 3 - On Fedora, Mandriva, Red Hat and most others, it's a full multi-user mode with console (non-graphical) login
 - 4 - Usually undefined by default and available for customization
 - 5 – On Fedora, Mandriva, Red Hat and most others it's the x system (graphical login)
 - 6 - Used to reboot the system. Also a transitional run level. Your system is completely shut down, and then the computer reboots automatically.
 - Change run level with init
 - init 1 - Changes to run level 1
 - id:2:initdefault: - located in /etc/inittab - Says the default run levels 2 runlevel
 - N 2 - n is the previous run level and 2 is the current run level. N if there is no previous run level
 - id:runlevels:action:process
 - id - Consists of a sequence of 1-4 characters that identifies its function
 - runlevels - consist a list of run levels for which this entry applies.
 - action - Wait, tells init to start the process once when entering a run level and to wait for the processes termination, respawn tells init to restart the process whenever it terminates.

Topic 102: Linux Installation and Package Management

102.1 Design Hard Disk Layout

Weight: 2

Description: Candidates should be able to design a disk partitioning scheme for a Linux system.

- /var
 - Used to contain files that will change frequently
- /home
 - User directories

102.2 Install a Boot Manager

Weight: 2

Description: Candidates should be able to select, install and configure a boot manager.

- Common Type Codes For Disks
 - 0x0c - FAT
 - 0x05 (old type of extended partition)
 - 0x07 ntfs
 - 0x0f newer type of extended partition
 - 0x82 - Linux swap
 - 0x83 - Linux file system
- Grub changes from grub 1 to grub 2
 - GRUB1: /boot/grub/menu.lst
 - GRUB2: /boot/grub/grub.cfg

102.3 Manage Shared Libraries

Weight: 1

Description: Candidates should be able to determine the shared libraries that executable programs depend on and install them when necessary.

- ldd
 - print shared library dependencies for the indicated program or file
- LD_LIBRARY_PATH
 - Bash environment variable that determines additional library locations to be searched when looking for shared libraries
- ldconfig
 - program run to clear the library cache after adding library locations to the /etc/ld.so.conf file
- /etc/ld.so.conf
 - Stores shared library paths for caching with ldconfig

102.4 Use Debian Package Management

Weight: 3

Description: Candidates should be able to perform package management using the Debian package tools.

- `/etc/apt/sources.list`
 - Stores the repository locations that apt uses to search for packages specific to your system
- `apt-get`
 - package handling and installation utility for Debian based distributions
 - will install packages by name and include dependent packages during install
- `dpkg`
 - installs .deb package files on Debian based systems
 - `-i`: install as well as configure package
 - `-r`: remove package
 - `-configure`: configures a package
 - `-c`: list contents of a package
 - `-s`: list status of package (installed or not)
- `apt-cache`
 - allows searching of named package or show installed packages
 - `apt-cache pkgnames` ← show installed
 - `apt-cache search` ← search for named package
- `dpkg-reconfigure`
 - reconfigure an already installed package
- `aptitude`
 - high level package management interface for Debian based distributions

102.5 Use RPM and YUM Package Management

Weight: 3

Description: Candidates should be able to perform package management using RPM and YUM tools.

- `rpm`
 - package installation utility for Red Hat based distributions
 - `-nodeps`: install the package without worrying about installed dependencies
 - `-i`: install
 - `-K`: check package signature
 - `-V`: verify
 - `-a`: all packages
 - `-Va`: verify all packages (for example)
 - `-U`: upgrades or installs a new package

- -F: upgrade a package already installed (if exists only)
- -q: query a package to determine if already installed
- -e: erase or uninstall
- -f query package owning file
- -p package
- -l (lowercase L) List files in a package
- -rebuild: rebuilds a source package
- -rebuildddb: rebuilds the rpm database
- -qa: print all install packages
- -ql: list files in an installed package
- -qf: determine which installed packaged a file belongs to
- -qpl: list all files in an RPM package
- -checksig: same as -K
- /etc/yum.repos.d
 - Directory containing yum source repository files
- cpio – create cpio archive
 - -d create leading directories where needed
 - -i extract
 - -u replace all files without asking
 - -m Retain previous modification times when creating files
- rpm2cpio
 - convert RPM packages to CPIO compressed files
 - Used primarily to extract files from a RPM package without installing the rpm package
 - rpm2cpio file.rpm | cpio -dium

Topic 103: GNU and Unix Commands

103.1 Work on the Command Line

Weight: 4

Description: Candidates should be able to interact with shells and commands using the command line. The objective assumes the bash shell.

- make
 - utility to read a source configuration file and maintain a group of files or programs (often used to compile source code and install the binaries once complete)
- env
 - shows your current session environment variables
- pwd
 - print the path of the current working directory
- bash
 - an 'sh compatible' command language shell interpreter that executes commands read from the standard input or from a file
- uname
 - prints system information
 - -n: node name
 - -s: kernel name
 - -v: kernel version
 - -r: kernel release
 - -m: machine/cpu information
 - -p: processor information
 - -i: hardware information

- -o: operating system name
- -a: print all information
- export
 - Used to set an environment variable
 - export HOME=/home/user
- unset
 - used to remove an environment variable
 - unset \$HOME
- man
 - The table below shows the section numbers of the manual followed by the types of pages they contain.
 - 1 Executable programs or shell commands
 - 2 System calls (functions provided by the kernel)
 - 3 Library calls (functions within program libraries)
 - 4 Special files (usually found in /dev)
 - 5 File formats and conventions eg /etc/passwd
 - 6 Games
 - 7 Miscellaneous (including macro packages and conventions), e.g. man(7), groff(7)
 - 8 System administration commands (usually only for root)
 - 9 Kernel routines [Non standard]

103.2 Process Text Streams Using Filters

Weight: 3

Description: Candidates should be able to apply filters to text streams.

- cat
 - a tool used to combine files (concatenate)
 - commonly used to display a single file as well
- fmt
 - reformats streams or files for display as indicated (WIDTH of columns)
- join
 - join lines of two files based on a common field or delimintor
 - -t: delimiter
 - -1 FIELD: join on this field of file 1
 - -2 FIELD: join on this field of file 2
- pr
 - covert files for printing (adds header and footer and sets breaks)
- split
 - split a file into pieces
 - split -l 2 file.txt name ← split file.txt every two lines, output namea nameb

- unexpand
 - converts spaces to tabs
- expand
 - converts tabs to spaces
- cut
 - used to remove sections from each line of files
- head
 - output the first part of a file
 - -c: XX bytes
 - -n: XX lines
- nl
 - add line numbers to a file for display or redirect to another file
- sed
 - stream editor for filtering and formatting text
 - sed 's/ugly/beautiful/g' myfile.txt ← replace all instances of 'ugly' with 'beautiful' in myfile.txt
- regex
 - short for regular expressions
 - referred to as the GNU tool implementation of the regular expression POSIX standard
 - encompasses tools like sed, awk, grep, egrep, ed, etc
 - examples:
 - [\t] would match white space tabs
 - [\t\r\n] would match line breaks
 - ([A-Z]) would match the alpha characters in caps
- tail
 - output the last part of a file
 - -c: XX bytes
 - -n: XX lines
- uniq
 - report or omit repeated lines
 - -d: only print duplicate lines
 - -u: only print unique lines
 - -r: recursive
- od
 - dump files in octal format
- paste
 - merge lines of files
- sort

- `tr`
 - sort forward/reverse contents by alpha/numeric characters
 - translate/squeeze/delete characters from standard input writing to output
 - `tr test blah < test`
 - sends contents of test file to standard input and trims out the blah contents
- `wc`
 - word count

103.3 Perform Basic File Management

Weight: 4

Description: Candidates should be able to use the basic Linux commands to manage files and directories.

- `cp`
 - copy files and directories
 - `-a`: archive
 - `-backup`: back up each destination file
 - `-d`: same as preserve links
- `mv`
 - move files and directories
 - `-f`: do not prompt to overwrite
 - `-i`: interactive, prompt for overwrite
 - `-n`: no clobber (do not overwrite existing file)
- `touch`
 - change file date/time attributes
 - `-a`: access time only
 - `-c`: do not create if not exist
 - `-d`: parse string and use it instead of current time
 - `-m`: Change only the modification time
 - `-r`: Use this file's times instead of current time
 - `-t`: Use specified timestamp
- `dd`
 - copy file, converting and formatting according to operands
- `gzip/gunzip`
 - compression utility, used in conjunction with tar for archiving
 - `-d`: decompress
 - `-f`: force
 - `-h`: help
 - `-l`: list

- -q: quiet
 - -t: test
 - -v: verbose
- mkdir
 - make directories
 - -p: make all directories in parent chain
- tar
 - tape archive utility, used for backups
 - -d Find differences between archive and file system
 - --delete Delete from archive
 - -c Create a new archive
 - -A Append tar files to an archive
 - -r Append files to the end of an archive
 - -u Only append files newer than copy in archive
 - -t List contents of an archive
 - -x Extract files from an archive
 - -z filter through gzip
 - -j filter through bzip2
 - -J filter through xz compression
- file
 - Determine file type
- bzip2
 - compression utility, used in conjunction with tar for archiving

103.4 Use Streams, Pipes and Redirects

Weight: 4

Description: Candidates should be able to redirect streams and connect them in order to efficiently process textual data. Tasks include redirecting standard input, standard output and standard error, piping the output of one command to the input of another command, using the output of one command as arguments to another command and sending output to both stdout and a file.

- &&
 - command1 && command2
 - In this example, command2 will only run if command1 is successful (i.e. return code of '0' from the process)
- tee

- The tee command will save standard input to a file as well as sending it to standard output. t command can be used to save standard input into a file as well as send it to standard output
- xargs
 - Build and execute command
 - -a Reads items from file instead of standard input
 - -0 Input items are terminated by a null character instead of by white spaces.
 - -d Input items are terminated by the specified character

103.5 Create, Monitor and Kill Processes

Weight: 4

Description: Candidates should be able to perform basic process management.

- jobs
 - Jobs command displays minimal information about processes associated with the current session
- ps
- By default ps only displays process that were run from its own terminal. -Ae will display all processes on a system. -u displays processes given by a specified user, H -F group processes and use indentation to show the hierarchy of relationships between processes. ps-w >ps.txt tells ps not to truncate to system
- and output to file.
- uptime
 - find uptime and display load average
- bg
 - restores a job to the running status but in the background
- fg
 - use CTRL+Z to pause a program and then fg to send the program to foreground
- kill
 - can be used to stop executing processes, uses PID
- nohup
 - run a command immune to hangups, with output to console or non-tty
- killall
 - can be used to kill all processes of a certain name
- free
 - show free memory and swap
- Common Kill Signals
 - SIGHUP 1 HANGUP
 - SIGINT 2 INTERRUPT FROM KEYBOARD
 - SIGKILL 9 KILL SIGNAL

- This signal is unblock able and causes the program to terminate abruptly. Only use if you can't terminate with 15.
- SIGTERM 15 TERMINATION SIGNAL
 - It asks the program to finish what it is doing then exit. Clean exists and is the preferred way of killing processes.
- SIGSTOP 17,19,23 STOP THE PROCESS
 - When a child process exits from a parent process it sends signal 1.
 - Signals in the man page **man -k signal**

103.6 Modify Process Execution Priorities

Weight: 2

Description: Candidates should be able to manage process execution priorities.

- nice
 - run a program with modified scheduling priority
- renice
 - alter priority of running processes
- top
 - display Linux processes
 - while running –
 - k Kills processes
 - q Quits processes
 - r Change process priority
 - s Change update rate
 - P Sort by CPU usage
 - m Sort by memory usage. Can also show uptime, memory info. and load
 - average (all the same as w).
 - ps - By default ps only displays
 - launch parameters –
 - -d Specifies delay between updates
 - -p Lists of to 20 specific PIDs
 - -n Display certain number of updates then quit
 - -b Batch mode.
 - Commands while

103.7 Search Text Files Using Regular Expressions

Weight: 2

Description: Candidates should be able to manipulate files and text data using regular expressions. This objective includes creating simple regular expressions containing several notational elements. It also includes using regular expression tools to perform searches through a filesystem or file content.

- grep
 - -v - Selected lines are those not matching any of the specified patterns.
 - egrep is the same as grep -E: interpret pattern as an extended regular expression.
 - fgrep is the same as grep -F: interpret pattern as a list of fixed strings, separated by new lines any of which is to be matched.
 - -i Ignores case and Matches any single character
 - [^] Matches any character not contained in brackets [^abc] matches any character other than a, b, or c.
 - ^ Matches the starting position of a line.
 - \$ Matches the ending position of a string or the position just before a string ending new line it matches the ending position of any line
 - * Matches the preceding element 0 or more times ab*c matches b zero or more times. IE ac, abc, abbbbc, abbbbbbbbbbbbbc.
 - + Matches the preceding element 1 or more times

103.8 Perform Basic File Editing Operations Using Vi

Weight: 3

Description: Candidates should be able to edit text files using vi. This objective includes vi navigation, basic vi modes, inserting, editing, deleting, copying and finding text.

- vi
 - text editor
 - i,o,a
 - /,?
 - c,d,p,y,dd,yy
 - h - Move left
 - j - Move down
 - k - Move up
 - l - Move right
 - ZZ - Save changes and quit
 - wq! - save and quit immediatly
 - q! - Quit without saving
 - e! -
 - lh - Move left one character
 - P - Paste above current line
 - p - Paste below current line
 - O - Open line above cursor
 - o - Open line below cursor
 - a - Append text after cursor
 - A - Append text at the end of the line

- I Insert text at beginning of line

Topic 104: Devices, Linux Filesystems, Filesystem Hierarchy Standard

104.1 Create Partitions and Filesystems

Weight: 2

Description: Candidates should be able to configure disk partitions and then create filesystems on media such as hard disks. This includes the handling of swap partitions.

- fdisk
 - partition a disk in preparation for install the Linux operating system or data storage
 - -n: interactive mode
- mkfs
 - Build a Linux file system (format)
 - -t Specifies the type of file system to build
 - -c Check device for bad blocks before building the file system
- mkswap
 - Turns a file system into swap. This is done after creating a new swap partition. swapoff and swap on commands cannot be used until a signature is created with mkswap.

104.2 Maintain the Integrity of Filesystems

Weight: 2

Description: Candidates should be able to maintain a standard filesystem, as well as the extra data associated with a journaling filesystem.

- du
 - estimate file space usage
 - -c Produce a grand total
 - -h Human readable
 - --max-depth - Print the total for a directory only if it is N or fewer levels below the command line argument.
 - -b – Bytes
- dumpe2fs
 - Obtains file system information. xfs_info does the same thing but for ifx file system
- df
 - Displays the filesystem usage, but not a breakdown within the file system
 - df . - Will show where your working directory is mounted
- mke2fs
 - create an EXT2/3/4 filesystem
 - equivalent of “mkfs -t ext2/3/4”
- tune2fs
 - Allows you to change the file system parameters such as: maximum mount count, time between checks, add a journal, set desire reserve blocks, obtain file system level. FILE SYSTEM CANNOT BE MOUNTED
- fsck
 - checks and repairs filesystems
 - -A Walk through /etc/fstab file and try to check all file systems
 - -a Attempt to automatically repair all errors
 - -C Display completion/progress
 - -N Don't execute; just show what would be done
- debugfs
 - Interactively modify a file system. Features: program provides abilities of tune2fs and dump2fs use debug fs to undelete a file if needed. Do not use on a mounted file system.

104.3 Control Mounting and Unmounting of Filesystems

Weight: 3

Description: Candidates should be able to configure the mounting of a filesystem.

- mount
 - attempts to mount filesystems to directories
 - -a Causes all files system mentioned in fstab to be mounted as indicated

- unmounts
 - -o Override mount options in fstab when manually mounting partitions.
 - Attempts to unmounts mounted filesystems
 - -f attempt to forcibly unmounts a filesystem in use or busy
- /etc/fstab
 - Configuration file containing persistent filesystem mounts (for mounting on boot)

104.4 Manage Disk Quotas

Weight: 1

Description: Candidates should be able to manage disk quotas for users.

- quotaon
 - Turns quotas on
- edquota
 - Edit a quota
 - -p - Duplicate the quotas of the prototypical user specified for each user specified.
- repquota
 - Summarize quotas for a file system
 - -a Print quotas of all the file systems configured with a quota mount option file at its root
 - -g Print only group quotas
 - -u Print only user quotas
 - -v Print a header line before printing each filesystem quota

104.5 Manage File Permissions and Ownership

Weight: 3

Description: Candidates should be able to control file access through the proper use of permissions and ownerships.

- setuid/setgid
 - normally set with the command 'chmod' by setting the high-order octal bit to 4 (for setuid) or 2 (for setgid).
 - chmod 6711 – will set the setuid and setgid bits (6)
 - will make the file read/write/executable for the owner and (7), executable by the group (the first "1") and others (the second "1")
 - all 'chmod' flags are octal
- text shortcuts
 - 4 – suid u+s
 - 2 – guid g+s
 - 1 – +t sticky bit

- **chmod**
 - change file/directory access attributes
 - 1 - execute
 - 2 - write
 - 4 - read
- **chgrp**
 - change group ownership of files/directories
- **umask**
 - change default user level permissions for files and directories when created
- **chown**
 - change ownership of files and directory
 - -R: all “ch” commands use this switch for recursion

104.6 Create and Change Hard and Symbolic Links

Weight: 2

Description: Candidates should be able to create and manage hard and symbolic links to a file.

- **ln**
 - create links between files or directories
 - no parameter means hard link
 - -s: softlink
 - If you remove the source file on a hard link, then the link file will still exist

104.7 Find System Files and Place Files in the Correct Location

Weight: 2

Description: Candidates should be thoroughly familiar with the Filesystem Hierarchy Standard (FHS), including typical file locations and directory classifications.

- **find**
 - find files on the local or any mounted file system
- **which**
 - display path the indicated command is in (if in PATH variable)
- **whereis**
 - locates the sources/binary and manuals section for specific files
- **/etc/updatedb.conf**
 - Location database configuration file
- **locate**
 - find files by name, uses the location database
- **FHS**

“Where things go in Linux”?

Linux uses **unified directory trees** - therefore every partition, removable disk, network file share, and other disk or disk-like storage device is accessible from a single directory tree (filesystem)

- **user files vs system files**
 - **system files - Files that control how the computer operates**
 - System startup scripts, and daemons
 - Program files, both binary and scripts
 - Program support files, such as fonts and icons
 - Configuration files that define how the system works (ex: network config settings, disk layout information etc)
 - Configuration files that most servers and other daemons
 - Data storage for system programs, such as the database that describes what programs are installed
 - System Log files, which record normal system activity
 - You would not want regular users to be able to access either read or write for example on your system configs - other example
 - /etc/shadow - this file holds encrypted passwords for instance
 - Such systems files are usually owned by root or by system accounts that have more limited rights
 - Many server programs rely on there own specific system users accounts.
 - Because ROOT user has access to read and write any files you protect access to config files of server apps and system settings by having the owned by root.
 - **User files** typically live under the /home/user directories
 - Even on personal laptop Linux installs systems files are separated by rights from root to your user to protect form accidental deletion or modification.

The Filesystem Hierarchy Standard (FHS)

- Linux distributions should have standardization in where the file system stores key system configuration files. This is so applications can access these
 - **FHS** is here to address the need for this standardization
 - **Shareable Files**
 - **Shareable files** can be shared between computers, like user data files and program binary files
 - You many not need to share these files but you certainly can
 - Normally when shared it would be on an **NFS** Network File System
 - **Un-shareable Files**
 - **Un-sharable** files contain system-specific info, like configuration files We wouldn't want to share these server files between computers.

- **Static Files**
- Don't normally change except through direct editing by the system administrator
- **These would be files like the program executables**
- **Variable files**
- **Can be changed/edited by users, automated scripts, servers etc**
- **FHS Common Linux Directories**
- **/** is the root directory - all files appear in the directory and the subdirectories built off of it
- **/etc** Holds systems Configuration files
- **/boot** Holds important boot files, like the Linux kernel, initial RAM disk, and usually boot loader config files
- **/bin** Holds program files that are critical for normal operation and that ordinary users may run
- **/sbin** Holds program files that are critical for normal operation and that ordinary users seldom run.
- **/lib** Holds libraries - code used by many other programs - these are critical for basic system operation
- **/usr** Holds programs and data used in normal system operation but that aren't critical for a bare-bones boot of the system
- **/home** Users' home directories. By separating this directory into its own low-level file system you effectively isolate most user data from the operating system - which is useful if you want to re install the OS without losing user data
- **/root** is the root user's home directory this is different than the **/** folder that is also pronounced root
- **/var** Holds miscellaneous transient files. these are things like files and print spool files. **/var/tmp** deserves special mention much like the **/mnt** (see below) **/var/tmp** holds temporary files. These files should not be deleted when the computer reboots
- **/tmp** Holds temp files - often including temporary files created by users programs. While many distributions don't delete these temp files - they should be deleted when the computer reboots
- **/mnt** The traditional mount point for removable media; this is usually split into subdirectories for each of the then mounted file systems
- **/media** The new mount point for removable media, again this is split into subdirectories for each of the mounted file system
- **/dev** This holds device files, which in turn provides low-level access to the system's hardware

- ***Remember** normal users should not be able to access most system directories like the /usr directory so the system can not be damaged
- Every day normal users create their user files in the /home directory that is broken up typically in the user name such as /home/stephen /home/Anthony
- Users can then also access where your removable media is like /media or /mnt
- You can use /tmp /var folders for users access as well - most users will not need to know about them however applications TYPICALLY use these folders for application temp files such as incoming emails files
- For System Admins the /etc folder is very important because this is where MOST of the system configuration files reside
- Example: /etc/fstab where you define where partitions are mounted
- Example: /etc/passwd primary account definition file
- Example: /etc/X11 (X windows system)
- Example /etc/samba Samba file server settings

Executable Directories

- **Program files** live usually in the /sbin, /bin, /usr/bin, and /usr/sbin. Such directories further will house programs/files in such directories as /usr/local/sbin and /usr/local/bin as the locally compiled programs
- **Library Directories** * Libraries are collections of programming functions that can be useful to programs. They are stored in separate files to save disk space and RAM when programs are ran. Most library directories live in /lib and /usr/lib however some can also reside in /usr/local/lib
- Windows programs typically all live within a single directory (its config files, binary applications etc) However on LINUX most of the programs key files are likely to reside in the above standard locations that are shared with other programs and are scattered out.
- **EXAMPLE:** Programs executable lives in /usr/bin but its libraries are in the /usr/lib and even furthermore its configuration files are in the /etc or even in the users' home directories.

2015 Objective Updates

Upstart

Upstart is a replacement daemon for the old SysVinit initialization program. A few key differences include

- The ability to start jobs/services asynchronously during the startup process which allows for a faster bootup period.
- Upstart “listens” for events on a system and executes jobs based off of events that jobs are set to start/stop on.
- Job configuration files are located in the /etc/init directory
- Upstart replaces the SysVinit but replicates backwards compatibility with the boot process as well as runlevels, leaving that architecture the same on Upstart systems (although in some cases, older configuration files simply being a text file pointing to the new location of method)
- Jobs can be managed with the “service” command

New commands with upstart

- `initctl list` (list all jobs located in /etc/init this does not list service files in /etc/init.d)
- `init-chefconfig /etc/file.conf` -> Checks the syntax of the job configuration file
- `start jobname` -> starts a job
- `stop jobname` -> stops a job
- `restart jobname` -> restarts a job
- Job management is also backwards compatible with the service command.

Systemd

- Systemd is a daemon that manages all other system daemons.
- Systemd is the first daemon to start during the boot process and is the last daemon to stop during shutdown.
- Systemd allows processes, daemons, and services to be started parallel to each other creating a

faster boot process

- Systemd enables on-demand starting of daemons which doesn't require an additional service to be running
- The SysVinit daemon started daemons/services one by one causing the system to have to wait
- Systemd has more of a central control for many of the regular Linux management tasks.
 - Systemd now controls items such as shutdown, cron, journald, and other low level components
 - Systemd is an ever evolving program and will incorporate more Linux utilities as it goes along
- Systemd is a system and service manager for the Linux operating system, not all distributions utilize Systemd, most commonly we see RedHat based distributions and soon a version of Debian 7. Ubuntu 14 utilizes the Upstart init daemon
- Systemd comes with built in support for a new daemon called journald.
 - Journald is responsible for event logging in the system
 - By default journald does not keep persistent data after a reboot.
 - Persistence can be applied to journald through additional configuration
- Service command has been replaced with the systemctl command for managing system services
- Systemctl also replaces the chkconfig command
- On systems that use the Systemd daemon the concept of runlevels has been completely replaced with "targets"
- Like SysVinit, it is still the first process to run at boot time with a PID of 1

Display different unit types
systemctl -t help

Journalctl

- Used to query the contents of the systemd journal

Systemd Commands And Directories

- Restart a service
 - systemctl restart httpd.service
- Enable a service to start at boot time
 - systemctl enable httpd.service
- Stop a service
 - systemctl stop httpd.service
- Move into the graphical.target
 - systemctl isolate graphical.target
- Set the default target for system boot
 - systemctl set-default graphical.target
- Get current default target
 - systemctl get-default

- System directory that contains the core system unit configuration files
 - `/usr/lib/systemd/system`
- Custom configuration directory which is override by any matching configurations in the system unit configuration library
 - `/etc/systemd/system`
- Display different unit types on the system
 - `systemctl -t help`
- Overview of available targets
 - `systemctl list-units --type=target | grep target`
- `systemctl list-unit-files --type=target -all` (all will show all enabled and disabled)
- What is the default target?
 - The default target is a sym link from `/etc/systemd/system/default.target` to the target configuration file
- Wall command is used to broadcast messages to all logged in users on the system
- `wall -n` will suppress the “from user” information and wall will open the wall console, just typing the message and ending with `ctrl+d` will send the message

Managing MBR and GPT Partitions

GPT Based partitions can have at 128 Primary partitions

GPT based partitions can have up to 9.4ZB or 8ZiB in disk size

MBR partitions are older and can have only 4 primary partitions

MBR partitions can have only 2TiB in size for each partition

`fdisk` - Used to managed MBR based partition tables

`gdisk` - Used to managed GPT based partition tables

`parted` - Used to manage GPT based partition tables

Commands and Compression

- Screen
 - Screen is a full-screen window manager that multiplexes a physical terminal between several processes
 - `-d` starts in detached mode and forks a new process
 - `-D` detach mode but doesn't fork a new process and command exists if the session terminates
 - `-S` session name
 - `-x` attach to a session that is not detached
 - `-X` send a specified command to a running screen session
 - `screen -list` will list all your current screens
 - `screen -r name` will reattach given a given screen
- XZ compression
 - Has a better compression ration than gzip (compresses data more)

- Requires more memory during compression but has lower file size footprint when completed
- Flags usage: `xz -flag filename`
 - `-z filename` will compress filename
 - `-d` decompresses also known as uncompressed
 - `-l` list information about compressed files and view compression ratios
- To compress a directory you must first tar the directory then use xz to compress it
- `pgrep` & `kill`
 - Used to identify processes based off of a pattern or regular expression
 - Examples
 - List all process associated with a user
 - `pgrep -u username`
 - List all processes associated with a user and display process name
 - `pgrep -l -u username`
 - Kill all processes started from a specific terminal
 - `kill -t ttyid`
 - Show all processes that DO NOT belong to a user (inverse)
 - `pgrep -v -u username`
 - Show the process id of the most recent process started for a user
 - `pgrep -n`
 - Show all SSHD processes associated with a user
 - `pgrep -l -u username sshd`
 - Kill all httpd processes
 - `kill httpd`
 - Kill all sshd processes for a specific user
 - `kill -u username sshd`
 - `dmesg`
 - Used to view the kernel messages in the kernel ring buffer
 - Most commonly used for troubleshooting kernel, memory, or any issue related to boot up and the kernel activity
 - examples
 - `dmesg > boot_messages`
 - `dmesg | grep -l tty`
 - `dmesg | grep memory`
 - `dmesg` will just dump the entire ring buffer to standard output.

GRUB2

`grub2-mkconfig > /boot/grub2/grub.cfg`

`/boot/grub2/grub.cfg` is made by the `grub2-mkconfig` command and is made of up files in the `/etc/default/grub` and the `/etc/grub.d` configuration locations.

`grub2-install /dev/device` -> will install grub on the specified device. If the GRUB bootloader is having issues then re-installing is the same process.

