



Linux Certification Boot Camp Study Flash Cards

Fold page over at vertical line or cover answer side while studying

v1.4

1	MBR	101.1	<p>MBR: Master Boot Record The MBR is located at the beginning of the hard drive, and usually contains the boot loader to boot the Operating System.</p>
2	Where does bios look to find a BOOT LOADER to boot the Operating System?	101.1	<p>BIOS looks to the beginning of the first hard drive in its MBR (Master Boot Record) to find a boot loader to boot the Operating System. <i>NOTE: In Linux, the boot loader can be placed in the MBR or the Root (/) Partition.</i></p>
3	LILLO Where is it loaded?	101.1	<p>LILLO is Linux's older boot loader. The newer boot loader is called GRUB (GRand Unified Boot Loader). LILLO is typically in the MBR, but can be installed in the ROOT (/) partition. LILLO cannot look beyond the first 1024 cylinders (8 GB) of the boot drive to find the boot loader <u>UNLESS</u> the LBA mode is enabled.</p>
4	Device, I/O Address, IRQ's?	101.1	<p>Device — I/O Address — IRQ LPT1 — 378-37f — 7 LPT2 — 278-27f — 5 COM1 — 3f8 — 4 COM2 — 2f8 — 3 Floppy — 3f0-3f7 — 6 Sound Card — 220 — 10 Remember, devices may share IRQ's but NOT I/O Addresses.</p>
5	modems	101.3	<p>COM1 /dev/ttys0 COM2 /dev/ttys1 COM3 /dev/ttys2 - An internal modem likely to appear as a third COM port.</p>
6	WinModems and Linux?	101.3	<p>WinModems depended on the CPU and special software drivers which are generally not compatible in a Linux system, unless a special driver is available for them. EXAM NOTE: When you are tested on this fact, WinModems are not totally incompatible; some of them can be made to work with special configuration. This fact is easy to remember, but is often drawn upon as a distracter on LPI exams.</p>

<p>7 Point-to-Point Protocol</p>	<p>101.3</p> <p>Point-to-Point Protocol (PPP) PPPD - Point to Point Daemon; Required to establish a dial-up connection. PPPD does NOT start up at boot time, and usually needs to be activated in the modem "chat" script. persist command - Reestablish a terminated PPP connection.</p>
<p>8 SCSI ID's</p>	<p>101.4</p> <p>All SCSI devices and SCSI controllers are required to have <u>unique</u> SCSI addresses (ID's). The older 8 bit 8-bit SCSI buses have three address lines and will only support 2³ (2x2x2) 8 devices (numbered 0 to 7 with 7 usually reserved for SCSI controllers). 16-bit buses have four address lines and can support 2⁴ (2x2x2x2) or 16 devices (numbered 0 to 15 with 0 expected to be a bootable device).</p>
<p>9 intentionally left blank</p>	
<p>10 How are SCSI devices named in Linux?</p>	<p>101.4</p> <p>Since each physical device is represented in the Linux filesystem by a directory under /dev, the first SCSI device is /dev/sda and its first partition is /dev/sda1 (not to be confused with the SCSI ID address). The second SCSI device would be /dev/sdb, etc. These drives are mounted from the /etc/fstab file.</p>
<p>11 SCSI Termination</p>	<p>101.4</p> <p>Both ends of a SCSI cable must be terminated with a terminator, but that most current SCSI drives have auto-termination. Improper termination may result in the drive originally seeming to function, and then failing completely.</p>
<p>12 /proc/scsi/scsi</p>	<p>101.4</p> <p>The /proc/scsi/scsi file will list all the SCSI devices currently attached and recognized devices by the SCSI subsystems.</p>
<p>13 /proc/dma /proc/interrupts /proc/ioports /proc/pci</p>	<p>101.5</p> <p>/proc/dma - This file contains a list of the registered ISA direct memory access (DMA) channels in use. /proc/interrupts - records the number of the interrupt's IRQ on the x86 architecture. /proc/ioports - provides a list of currently registered port regions used for input or output communication with a device.. /proc/pci - Lists which PCI devices are already installed and their I/O addresses and interrupts.</p>

14

pnpdump
Isapnp
/etc/isapnp.conf

101.5

pnpdump - The pnpdump utility was created to inspect the system for installed isa devices.
Isapnp - The isapnp utility is a part of the isapnptools package and checks resources that are available to configure ISA pnp cards. If there is a conflict, a diagnostic message is given, and the program can be made to abort if required by placing the appropriate settings in the configuration file. /etc/isapnp.conf is the standard place for the conf file, or configuration file, which isapnp reads.

15

setserial

101.6

Used to set baud rate on a serial device. setserial is a program designed to set and/or report the configuration information associated with a serial port. This information includes what I/O port and IRQ a particular serial port is using.
spd_vhi = 115KB/s
spd_hi = 56KB/s
spd_normal = 38.4KB/s

16

USB specification 1.0
USB specification 2.0

101.7

USB specification 1.0, FIRST SUPPORTED IN LINUX KERNEL **2.2.7** provided for low device speeds not to exceed about 8.5Mbit/s, with a more typical speed on 2.0 Mbit/s. Mice, keyboards, joysticks, and other peripherals devices used these speeds to communicate at 1.5 Mbit/s. **USB specification 2.0** made especially for higher speed **storage devices** allows 127 devices to be connected at the same time and the total communication bandwidth is limited to 12Mbit/s.

17

lspci

101.7

lspci - Lists PCI configuration information for currently installed PCI buses and devices connected to them. Also used as: **lspci |less**

18

OHCI and UHCI
usb-uhci.o
usb-ohci.o

101.7

USB host controllers are compatible with either the Open Host Controller Interface (OHCI, by Compaq) or the Universal Host Controller Interface (UHCI, by Intel) standard; hence, the Linux driver names following these standards respectively.
usb-uhci.o and **usb-ohci.o** are the Compaq and Intel USB driver modules, respectively.

<p>19</p> <p style="text-align: center;"><code>/etc/usbmgr</code> <code>/proc/bus/usb</code></p>	<p>101.7</p> <p>Linux uses a so-called "USB Manager", which automatically loads kernel modules (drivers) for USB devices. It loads only drivers for devices that are present on the USB bus to reduce memory usage. You may need to configure this file to load the <code>usb-uhci</code> or <code>usb-ohci.o</code>.</p> <p>Suppose, for instance you wanted to mount a new HP USB CD drive. You find that it runs the Intel specification of <code>usb-uhci</code>, and you add that or un-comment that from the <code>/etc/usbmgr</code> file.</p> <p>The USB device filesystem is a dynamically generated filesystem, similar to the <code>/proc</code> filesystem. This filesystem can be mounted just about anywhere, however it is customarily mounted on <code>/proc/bus/usb</code>, which is an entry node created by the USB code, intended to be used as a mount point for this system.</p>
<p>20</p> <p style="text-align: center;"><code>usbmodules</code> <code>/etc/hotplug</code></p>	<p>101.7</p> <p>usbmodules is a utility that lists driver modules that may be able to manage interfaces on currently plugged in USB devices. <code>usbmodules</code> may be used by <code>/sbin/hotplug</code> or one of its agents (normally <code>/etc/hotplug/usb.agent</code>) when USB devices are "hot plugged" into the system.</p> <p>/sbin/hotplug - This script is called by default by Linux kernel when something changes in its internal state (e.g., a new device is added or removed). Hotplugging means automatically detecting and mount new USB devices as they are plugged into the system.</p>
<p>21</p> <p style="text-align: center;">What are the two (2) required Linux partitions?</p>	<p>102.1</p> <p>Root partition (/) and the swap partition</p> <p>Note: Some exam questions may ask about only one required partition, in which case, that would have to be Root partition.</p>
<p>22</p> <p style="text-align: center;">What is the swap partition needed for? What is the recommended size of the swap partition?</p>	<p>102.1</p> <p>The swap partition is used to store portions of main memory which are not used as often. These portions are moved from hard disk space to live memory and back again, hence the concept of swapping. For the exam, the recommended size of the swap partition should be twice (2x) the size of the installed RAM on the machine. NOTE: This swap partition does NOT appear in any file manager, is NOT mounted in the <code>fstab</code> file, and is NOT accessed by anything other than the OS itself.</p>

<p>23</p> <p>When is it advantageous to mount the following in their own partitions?</p> <p style="text-align: center;">/var /home /usr /boot</p> <p style="text-align: right;">102.1</p>	<p>/var - When large log files are expected from many applications.</p> <p>/home - When there are many users so to allow more space for their home directories.</p> <p>/usr - when there are many applications installed and running from the server and/or when additional space is needed for frequent kernel rebuilding.</p> <p>/boot - Creating a separate /boot partition as small as one cylinder or 12MB or more typically 50MB to keep it under the 1024 cylinder limit.</p>
<p>24</p> <p style="text-align: center;">/etc/lilo.conf</p> <p style="text-align: right;">102.2</p>	<p>This file is read by LILO during the second stage of booting. This file tells LILO where to find the kernel image(s), video configuration to use, the default boot drive, etc.</p> <p>The image path in that lilo.conf file specifies one particular kernel to load. /etc/lilo.conf can contain multiple lines pointing to multiple kernels (all will be commented out except the one you wish to run at the next boot up).</p>
<p>25</p> <p style="text-align: center;">Kernel Numbering</p> <p style="text-align: right;">102.2 105.1</p>	<p>major.minor.patchlevel (an example is 2.6.5)</p> <p>Kernels are numbered as Major Release Number is first (2 in the example), Minor Release Number second (6 in the example), and Patch Level third (5 in the example).</p> <p>A "stable" kernel is one whose minor release number is an even number (e.g. 2.4 or 2.6).</p> <p>A "developmental" kernel is one whose minor release number is an odd number (e.g. 2.5).</p>
<p>26</p> <p style="text-align: center;">gzip</p> <p style="text-align: right;">102.3</p>	<p>gzip - Program used to compress a file. After it compresses the file, it gives the files the extension .gz to show that it has been compressed.</p>
<p>27</p> <p style="text-align: center;">tar tar -z tar -x tar -v tar -f</p> <p style="text-align: right;">102.3</p>	<p>tar - This program takes many files and groups them all into one file archive with the extension .tar .</p> <p>tar is often used with compressed files from gzip</p> <p>Common tar Commands:</p> <ul style="list-style-type: none"> -z - GZip; uncompress file using gzip while reading data. -x - Extract; Extract files from archive. -v - Verbose; Display the files being extracted. -f - File; Read the archive from the file given (if you don't specify a file, tar assumes a tape drive).

28 "tarball"
What will **tar -zxvf**
<application>.tar.gz typed into
the command line do?

102.3

A "**tarball**" is the name for a file with the extension .tar.gz or .tgz. This file is one in which its files have been compressed with **qzip** and all its files have been packaged into one file with **tar**.

tar -zxvf <application>.tar.gz - Type this on the command line to uncompress, extract the file application.tar.gz and show the results on the screen.

29 What are the steps involved in
installing an application from
source code?

102.3

If you had downloaded and uncompressed source code, it needs to be compiled before installation. Here are the steps:

1. **Configure** - Type **.configure** to analyze your system and prepare a **Makefile**.
2. **Make** - type **make** to compile the source code.
3. **Install** - Type **make install** to install the application on your system.

30 **ldd**
ldd <application>

102.4

An **ldd** is a Dynamically Linked System Library (much like a .dll file in MS Windows).

ldd <application> - Used to display the shared libraries required by each of the applications listed on it's command line.

31 Dynamically Linked System
Libraries (or dll)

102.4

When a program is compiled in Linux, many of its functions have dependencies on other functions. Several programmers write programs each requiring the same code or function to operate properly. We say that they may draw upon common code running from a shared library of code, just to keep from unnecessarily duplicating the same code all over the filesystem.

Called dynamically linked system libraries (or dll), this shared code runs from a link to a shared library.

ld.so
/etc/ld.so.cache
LD_LIBRARY_PATH

32

ldconfig -l
ldconfig -p
ldconfig -v

102.4

When a program requiring shared library dependencies is run, the shared object linker, ld.so attempts to locate them. It looks in the environment variable for the library path, LD_LIBRARY_PATH. If it fails to find them, the application won't run. Rather than continually adding new library paths to the LD_LIBRARY_PATH variable, the index /etc/ld.so.cache lists where these shared libraries are located.

ldconfig - used to update entries the cache in /etc/ld.so.cache.

ldconfig -l updates libraries in expert mode

ldconfig -p displays the current contents of ld.so.cache without updating it.

ldconfig -v gives verbose progress during its execution.

rpm -i
rpm -U
rpm -e
rpm -q; -qi; -qpil
rpm -V

33

102.6

Much of the major software available for Linux comes pre-compiled with an installation program, rpm (RedHat Package Manager). RPM automates the following life-cycle sequences of events, each with its own option after the RPM command:

rpm -i (or --install) installs a RPM package

rpm -U (or --upgrade) upgrades a RPM package

rpm -e (or --uninstall) uninstalls a RPM package

rpm -q (or --query) queries about the files that have been installed by a RPM package

rpm -V verifies what files were installed against a RPM package's list of what should have been installed.

rpm -qi In this case, the "i" does not mean "install" as it would have in its own mode but instead displays the package information you are querying

rpm -qpil - q to enter query mode, p to query a

"package" file listed, i for package information, and l for list all the files contained in the package.

rpm -q --changelog
<application>.rpm

34

102.6

The changelog feature of the query mode displays a listing of chronological changes that have been made to the components included in the application's RPM. To access this changelog from within the query mode type **rpm -q --changelog <application>.rpm**

/etc/rpmrc

35

102.6

/etc/rpmrc - RPM Configuration file. RPM provides a configuration file for specifying frequently used options. The system configuration file is usually /etc/rpmrc, and users can set up their own \$HOME/.rpmrc file. You can use the --showrc option to show the values RPM will use for all the options that may be set in an rpmrc file:

rpm --showrc

36 What are the three (3) main parts of Linux?

103.1

1. Hardware

2. Kernel - The kernel manages processes and is the interface to the Hardware.

3. Shell - The shell is a program that provides the user with an interface to the operating system.

37 Shell
What does the shell do?

103.1

Shell - The shell provides an interface for users to communicate with the Linux kernel.

What does the shell do?

- The shell is a command interpreter that waits for input and then executes the user or program commands.

- The shell is a powerful programming environment (scripts).

- The shell is customizable, in other words, it can be tailored to meet the needs of the users.

- The shell also creates and controls processes.

38 Kernel

103.1

Kernel - The kernel manages processes and is the interface to the hardware.

39 Command Line Prompts
\$ and #

103.1

\$ - The command line prompt for a non-root user.

- The command line prompt for a root superuser.

NOTE: If you see the # on your command line prompt, then you know that you have root privileges.

40 What is the command format?

103.1

The command format is:

command -options arguments

-Spaces are needed between the three different components.

- Options may be bunched together (-l ai), or separated (-l -a -l), but if they are separated, then there must be a space between each option AND each option should be preceded by it's own dash '-l'.

41 What shell variable declares the size of the command history?
What file holds the history?

103.1

The variable **\$HISTSIZE**

The history is stored in the **.bash_history** file located in the user's home directory.

42 /etc/shells

103.1

The **/etc/shells** file serves as the list of valid shells that may be loaded.

<p>43</p> <p>ls -a ls -l ls -R</p>	<p>103.1</p> <p>ls - The list command. ls -a - List all files, including any hidden files (files that begin with a period, as in .bash_history). ls -l - Long format listing (includes permissions, owner, size, modification time, etc.) ls -R - List directories and their contents recursively.</p>
<p>44</p> <p>In what file are color settings for the ls command located?</p>	<p>103.1</p> <p>The color settings for the ls command are located in the file /etc/DIR_COLORS</p>
<p>45</p> <p>How do you open a shell? How do you close a shell?</p>	<p>103.1</p> <p>Open a shell? Type in bash on the command line to open a new shell. Close a shell? - CTRL-D - Type exit on the command line - Type logout if it is the original shell.</p>
<p>46</p> <p>/etc/profile</p>	<p>103.1 109.1</p> <p>Sets up shell variables that are global for everyone. Loaded right after login. Executes once only at login. Contains: UMASK - for permissions masking PATH - path for search order HISTSIZE - size of command history MANPATH - path to the man pages Link to /etc/SuSEconfig/profile Link to /etc/bash.bashrc Link to \$HOME/bashrc Link to /etc/profile.local</p>
<p>47</p> <p>/etc/profile.local</p>	<p>103.1 109.1</p> <p>The file /etc/profile.local : - Shell variables; Global - Overrides /etc/profile settings - Executed once at login. NOTE: Any changes to the profile should be made in /etc/profile.local rather than /etc/profile.</p>

48

`/etc/bash.bashrc`103.1
109.1**/etc/bash.bashrc**

Shell variables

Executes right after `/etc/profile` and `/etc/profile.local` on login

Executes upon opening every new shell

/etc/bash.bashrc Contains:

- `$PS1` - User command prompt
- `ALIAS` - Aliases
- Link to **/etc/bash.bashrc.local**

NOTE: Any changes in this file should NOT be done in this file, but rather in the `/etc/bash.bashrc.local` file.

49

`/etc/bash.bashrc.local`103.1
109.1**/etc/bash.bashrc.local**Overrides `/etc/bash.bashrc`Changes should ONLY be made in THIS file instead of in the `/etc/bash.bashrc` file.

50

`set`
`unset`103.1
109.1**set** - With no arguments, **set** displays all the variables and all the functions for a current shell. With arguments (as in **set \$VARIABLE**), **set** is used to change or add variables to the current shell alphabetically.**NOTE:** This only shows variables for this shell only. Any child shells will need to have these variables exported to them by using the **export** command.**unset** - Used to remove a variable from the current shell.

51

`env`

103.1

env - Much like the **set** command in that without arguments it displays current environment variables for the current shell, except **env** DOES NOT display the functions and they are not displayed in alphabetical order. With arguments (as in **env \$VARIABLE**), **env** can be used to change or add environment variables to the current shell.

52

What is the command search order?

103.1

1. Qualified Search Path (such as `/bin`)**2. Reserved Words** (or keyword; such as `done`, `for`, `if`, `fi`, `else`, `select`, etc.)**3. Alias** (created by various environment configurations; such as `alias mud = ls -l`)**4. Built-In** (these are not modifiable; such as `ls`, `pwd`, `cat`, `exit`, etc.)**5. Functions** (as listed by the **set** command)**6. Path Variable or Hashed** (as set up in `/etc/profile` and/or `/etc/profile.local`)

<p>53</p> <p>Single Period '.' (as in ./command)</p> <p>Double Period '..'</p>	<p>103.1</p> <p>Single Period '.' - Current Directory Double Period '..' - Parent Directory NOTE: To use a file in the current directory if that directory is not within the command search order, you must use the single period, as in ./command.</p>
<p>54</p> <p>pwd</p>	<p>103.1</p> <p>pwd - "Print Working Directory"; Displays the current working directory's absolute path name.</p>
<p>55</p> <p>echo</p>	<p>103.1</p> <p>echo - The values that the variables represent can be displayed by entering the echo command. The trick is that the variable must be preceded by a \$ and entered in uppercase. All system variables are by default in uppercase. An example is echo \$PATH.</p>
<p>56</p> <p>alias</p>	<p>103.1</p> <p>alias - The format for alias is: alias name = cmd Assigns a shorthand name as a synonym for the cmd. If the =cmd is omitted, as in alias name, then the alias set up for name is displayed. And, if just alias is used, then a list of all the current aliases is displayed. - Aliases are set in /etc/bash.bashrc or /etc/bash.bashrc.local. - To export an alias to a child shell, use the export command (export alias).</p>
<p>57</p> <p>head -n file tail -n file</p>	<p>103.2</p> <p>head -n file - Displays the first n lines of a file. If a number of lines is not given, then the default is 10 lines. tail -n file - Displays the last n lines of a file. If a number of lines is not given, then the default is 10 lines.</p>
<p>58</p> <p>od -c file od -o file od -x file</p>	<p>103.2</p> <p>od - Octal Dump; Dumps the specified file to standard output. od -c file - ASCII Characters od -o file - Octal Characters (the default) od -x file - Hexadecimal Characters</p>
<p>59</p> <p>cat file cat > file tac file tac > file</p>	<p>103.2</p> <p>cat file - List the contents of a file. cat > file - Allows user to type input data and sends it to and saves it as that file. tac file - List the lines of a file in reverse order. tac > file - Allows user to type input data and sends it to and saves it as that file in reverse.</p>

60

sed

103.2

sed - The sed command is an easy one to remember, stream editor. The use of it is tricky however, and it is very powerful. For example, it can replace characters "x" and "y" with "a" and "b", remove certain characters, and so forth. It is designed to handle many repetitive steps to manipulate data as it filters it.

61

Files and Directories

103.3

Files - Collection of data or a non-organized stream of data.

3 types of files:

1. Ordinary data - text or code
2. Data organized as a directory
3. Special files which include blocks of data that represent a CD-ROM for example.

Directories - Only contains the:

1. Name of the directory
2. I-Node number ("Index Number")

62

mkdir
rm
rmdir
rmdir -p directory

103.3

mkdir - Use to create (make) a directory.

rm - Remove; Used to delete files

rmdir - Remove Directory; Used to remove an empty directory

rmdir -p directory - The **-p** option will remove all subdirectories too.

NOTE: You cannot delete the directory you are currently in.

63

touch

103.3

touch - The **touch** command updates the last accessed and modification dates on the files.

Touch can also make a zero length file by entering **touch filename**.

64

cp
cp -i
cp -r or cp -R

103.3

cp - The cp command simply copies a file or multiple files to a target consisting of a file name or directory.

cp -i - Will prompt if a file exists at destination with the same file name. The "i" stands for "interactive."

cp -r or cp -R - Copy a whole directory structure (recursively)

65

mv

103.3

mv - the mv command does two things:

1. **Moves** a file or directory to a new location in the filesystem.
2. **Renames** files or directories.

66	<p><u>File Globbing:</u></p> <p>?</p> <p>*</p> <p>[!ab]</p> <p>[az]</p> <p>[A-B]</p>	103.3	<p>File Globbing is using wild cards to define file names.</p> <p>? - Matches any single character</p> <p>* - Matches any number of characters</p> <p>[!ab] - Matches all characters except those following the !</p> <p>[az] - Matches any of the characters listed</p> <p>[A-B] - Matches any characters within the range listed</p> <p><u>NOTE:</u> File Globbing can be used with many commands such as ls, cat, rm, rmdir, etc.</p>
67	find	103.3	find - Used to find a file; Searches recursively automatically (Do not need to use -R).
68	<p>stdin</p> <p>stdout</p> <p>stderr</p>	103.4	<p>stdin - Standard Input; <0 (or 0<)</p> <p>stdout - Standard Output >1 (or 1>)</p> <p>stderr - Standard Error 2></p>
69	>>	103.4	<p>>> - Used to append data to another file.</p> <p><u>Example:</u> ls >> ls_weekly.out</p> <p>This example will append the results of ls to the end of the file <i>ls_weekly.out</i></p>
70	<p> (pipe)</p> <p>tee</p> <p>xargs</p>	103.4	<p> (pipe) - Takes the stdout and it becomes the stdin for the second command.</p> <p>tee - Take stdin and sends it to two different files</p> <p>xargs - Causes a command to take as stdin a file with many arguments and yet runs the command only once.</p>
71	What is a process?	103.5	<p><u>What is a Process?</u></p> <p>A process is actually a program or a command running on the Linux system. Many processes can be running at the same time. A process is spawned off from a parent process. Each of these have an ID the process has a PID. The parent process has a PPID.</p>
72	What is a job?	103.5	<p><u>What is a job?</u></p> <p>When a command is assigned to run in the background under your shell it becomes a job and is given a job number as well as a PID. The job number remains effective as long as the job is running.</p>

73	<p>& fg bg</p>	103.5	<p>& - Starts a process in the background fg - Bring current job to the foreground. bg - Put current job into the background.</p>
74	<p>jobs ps pstree top</p>	103.5	<p>jobs - Lists all running and stopped jobs. ps - Lists running processes. pstree - Displays running processes in a hierarchical order. top - Provides information about the "top" most CPU-intensive processes currently running.</p>
75	<p>kill kill HUP kill -signal pid</p>	103.5	<p>kill - sends signals to process ID's kill HUP - Used to restart a process so tht it will re-read its configuration file after a change has been made to it. Signals: 01 - Hangup ; If you logout before process is done, then process will hang. 09 - Kill signal; 15 - Terminate; NOTE: 15 is the default signal</p>
76	<p>nice renice</p>	103.6	<p>nice - Use to gives jobs a lower PS priority before that job has begun. Used to start the job at a different priority by putting the nice command in front of the command or script, as in nice -10 ls -lR / >bigfile &. renice - If the jobs needs to be adjusted after it begins, the renice command needs to be run. With the following typed on the command line renice -10 -u tux -p 476 all jobs owned by tux and job 476 are increased in priority.</p>
77	<p>grep grep -v grep -c grep -l grep -n grep -i grep -w grep -s</p>	103.7	<p>grep - used to find information in a file. Format: grep [options] pattern [files] grep -v - Print only the lines that do not match. grep -c - Print only the count (or number) of matching lines. grep -l - Print only names of files with matches. grep -n - Print matching lines with line numbers. grep -i - Ignore case in letters (uppercase and lowercase ignored). grep -w - Do whole word search grep -s - Suppress all error messages</p>
78	Regular Expressions	103.7	Regular Expressions are logical constructs that use patterns and wildcards to symbolize something special in that text or file.

<p>79</p> <p style="text-align: center;">vi i ESC (Esc key) : :! :w :x :q :q!</p>	<p>103.8</p> <p>vi - Text editor program; To begin the editor type in vi and the file name (new or old). vi commands: i - enters the insert mode ESC (ESC key) - quit the insert mode : - enter command mode :! - enter an external command line command :w - save (writes) changes without exiting :x - quit and saves :q - quit (if no changes were made) :q! - quit withOUT saving (! forces the quit)</p>
<p>80</p> <p style="text-align: center;">IDE Drive Names? SCSI Drive Names?</p>	<p>104.1</p> <p>IDE controllers have always had a master and slave drive. This is key because the naming of the drives is based on this standard. The default names for IDE drives are: hda - Primary Master Drive, hdb - Primary Slave Drive hdc - Secondary Master Drive, hdd - Secondary Slave Drive The default names for SCSI Drives are: sda - The first drive, sdb - The second drive.</p>
<p>81</p> <p style="text-align: center;">Partitions 3 types of partitions?</p>	<p>104.1</p> <p>The 3 types of partitions are: 1. Primary Partition 2. Extended Partition 3. Logical Partition NOTE: Disk drives may be broken up into 16 partitions each.</p>
<p>82</p> <p style="text-align: center;">Primary Partition</p>	<p>104.1</p> <p>Primary Partition - These partitions contain filesystems. At least 1 primary partition must exist on a disk, but no more than 4 may exist on a disk. They are named hda1, hda2, hda3, and hda4.</p>
<p>83</p> <p style="text-align: center;">Extended Partition</p>	<p>104.1</p> <p>Extended Partition - Extended partitions are like primary partitions and are treated as such. The exception is that extended partitions may not contain a filesystem. Instead they contain Logical Partitions. A disk may only have 1 extended partition, and it takes the place of 1 of the 4 primary partitions. Extended partitions are numbered just like primary ones (hda2 for example).</p>

84

Logical Partition

104.1

Logical Partition - Logical partitions exist in extended partitions.

There may be up to 12 logical partitions in an extended partition.

Logical partitions are number 5 to 16 (1 through 4 are reserved for primary and extended partitions). They are named hda5, hda6, to hda16.

Typical filesystems mounted on logical partitions are /boot,/opt, /tmp, /usr, /var, /home, and swap.

85

mount points

104.1

mount points - In order to access any filesystem in Linux, that filesystems must be attached to the directory structure by using the mount command.

The filesystem is attached as a directory to an existing directory tree. This directory is known as its mount point.

When Linux boots up, the first filesystem that mounts is the / (called root). In a very simple small system, everything could be in the / Filesystem with only a single mount. In larger systems with more Filesystems and partitions they are all mounted to the / Filesystem and they then makeup a unified directory of files and directories.

86

Which directories have a separate partition or become a Filesystem?

104.1

There is no hard fast rule on this. You need to know the purpose and your use of each directory to make this decision and the recommendations in the FHS.

The following are exceptions; they need to be part of the / (called root) Filesystem: /bin, /sbin, /dev, /etc, /lib. Note that /boot usually is part of / also.

Each of the following usually has their own partition: /home, /tmp, /var, and /usr.

87

/

104.1

What is this directory?

Root of Hierarchy

NOTES:

Root of the file system

If kernel is located here, name it vmlinuz or vmlinuz; otherwise in /boot

This is a STANDARD Directory.

Permissions = rwx r-x r-x

88	/bin	104.1	<p>What is this directory? Binaries Directory NOTES: Essential User (admin & user) command binaries. Executable programs. Can be NO sub-directories. (e.g. = cat, chgrp, chmod, chown, copy kill, ls, rm, mount) This is a STANDARD Directory. Permissions = rwx r-x r-x</p>
89	/boot	104.1	<p>What is this directory? Boot Files Directory NOTES: Static Files of the boot loader prior to kernel execution. No boot up configuration files are located here. OS Kernel must be in / or /boot. This is a STANDARD Directory. Permissions = rwx r-x r-x</p>
90	/dev	104.1	<p>What is this directory? Devices Directory NOTES: Special or device files that can be manually created. Makedev command must reside here. Also has /dev/null crw-rw-rw- This is a STANDARD Directory. Permissions = rwx r-x r-x</p>
91	/etc	104.1	<p>What is this directory? Configuration Files Directory NOTES: Configuration files and Directories for current system...like LILO.conf. NO binaries here. Much like Win Registry. Keep intruders out of here!!! Also can have sub-dir "opt" for add on apps. Contains: fstab, hosts_, inittab, passwd, profile, protocols, networks, + This is a STANDARD Directory. Permissions = rwx r-x r-x</p>
92	/lib	104.1	<p>What is this directory? Library Files Directory NOTES: Shared library images needed to boot and run commands in root filesystem (i.e. by binaries in /bin and /sbin) Can have sub-dir /lib/modules for other kernel loadable modules This is a STANDARD Directory. Permissions = rwx r-x r-x</p>

93	/mnt	104.1	<p>What is this directory? Mount Point Directory</p> <p>NOTES: Mount point for a temporarily mounted filesystem like floppy and CD-Rom Should not be used by any program being run...a local use only This is a STANDARD Directory. Permissions = rwx r-x r-x</p>
94	/opt	104.1	<p>What is this directory? Optional Directory</p> <p>NOTES: For Standard add-on apps. For Local Admin use It's static files MUST be located in /opt/ <package> directory This is a STANDARD Directory. Permissions = rwx r-x r-x</p>
95	/sbin	104.1	<p>What is this directory? System Binaries Directory</p> <p>NOTES: Utilities used by system Admin and other root-only commands. Especially binaries for booting, restoring, recovering and repairing. Required commands to be here: shutdown, fastboot, fasthalt, fdisk, fsck, ->halt, ifconfig, init, mkfs, mkswap, reboot, swapon, swapoff, update. This is a STANDARD Directory. Permissions = rwx r-x r-x</p>
96	/tmp	104.1	<p>What is this directory? Temporary Directory</p> <p>NOTES: Tmp files made available to programs. Recommended to delete them on boot up. This is a STANDARD Directory. Permissions = rwx rwx rwx</p>

<p>97</p> <p style="text-align: center;">/usr</p>	<p>104.1</p> <p>What is this directory? Unix Sharable Resources NOTES: 2nd major section of filesystem; Shareable, read-only data Shareable between various FHS-compliant hosts Any data host specific or varies with time is stored elsewhere Required sub-dir: bin(primary dir for executable commands), include(system's general-use include files for C prog lang), lib(object files, libraries, internal binaries that are NOT executed by users or shell scripts), local(must have bin, games, include, lib, man, sbin, share, src), sbin(non-essential binaries), share Optional sub-dir: X11RS(X windows sys), games, lib<equal>, src(source code) This is a STANDARD Directory. Permissions = rwx r-x r-x</p>
<p>98</p> <p style="text-align: center;">/var</p>	<p>104.1</p> <p>What is this directory? Variable Sized Files NOTES: Variable sized data like spools, logging, temporary files. A helper to /usr, so /usr can be made Read-Only. Should be in separate partition, move out of / partition and into the /usr partition. HOWEVER, do not link /var to /usr but link /var to /usr/var. REQUIRED SUBDIRECTORIES: cache, lib, local, lock, log, opt, run, spool, tmp RESERVED SUBDIRECTORIES: backups, cron, msgs, preserve APPS MUST SUBS: account(logs), crash(dumps), games, mail(user mailbox), yp(network database files). This is a STANDARD Directory. Permissions = rwx r-x r-x</p>
<p>99</p> <p style="text-align: center;">/home</p>	<p>104.1</p> <p>What is this directory? User's Home Directories NOTES: Systems users home directories. This is an OPTIONAL Directory. Permissions = rwx r-x r-x</p>
<p>100</p> <p style="text-align: center;">/root</p>	<p>104.1</p> <p>What is this directory? Root User's Home Directory NOTES: User ROOT's home directory This is an OPTIONAL Directory. Permissions rwx --- ---</p>

<p>101</p> <p style="text-align: center;">/proc</p> <p>104.1</p>	<p>What is this directory? Kernel Process Information Directory NOTES: Is De-facto standard virtual file system for the Linux internal use. A virtual file System for the Kernel, Process, Memory and System Information. This is an OPTIONAL Directory. Permissions = rwx r-x r-x</p>
<p>102</p> <p style="text-align: center;">/srv</p> <p>104.1</p>	<p>What is this directory? Services Directory NOTES: Site specific services for this host only. Examples: CVS, Apache.www This is an OPTIONAL Directory. Permissions = rwx r-x r-x</p>
<p>103</p> <p style="text-align: center;">File Types (4 types of files)?</p> <p>104.1</p>	<p>Static - Does not change; Read-Only Permission Variable - Changes are made as needed Shareable - Is shared with other users on this system or network; Read-Only permission; not backed up as often; separate filesystem. Unshareable - Considered local files for this host only; Config and boot files are a good example.</p>
<p>104</p> <p style="text-align: center;">What is a filesystem? What are the different types of filesystems?</p> <p>104.1</p>	<p>What is a filesystem? The Filesystem evolves from the following:</p> <ol style="list-style-type: none"> 1. The hard disks are labeled hda or sda by the partitioning. "hda" is for ide drives and "sda" is for SCSI drives. 2. Each partition is given a unique name by adding a number to it like sda1 or hda2. 3. Next each partition is formatted. This creates a Filesystem type as in ext2, ext3 or ReiserFS. 4. We need to have a place to mount these formatted Filesystems, and do this by giving them mounting points. <p>Filesystem types: ext2, ext3, ReiserFS, vfat, xfs</p>

<p>105</p> <p>What are the major criteria for deciding which directories become their own filesystems?</p>	<p>104.1</p> <p>The major criteria for deciding which directories become their own Filesystems are:</p> <ul style="list-style-type: none"> - The size they will become (/home). - Whether they are static (usually read only like /usr) or dynamic files (/tmp). - Backup considerations. - Performance of the disk/partition they reside on (/swap). - Amount of activity on them. - The critical nature of their content. - A pseudo or virtual Filesystem (really all in memory) like "/proc" that is a doorway to the kernel/ memory, where each "directory" represents a process and each "file" contains info on the condition of that process. - Built for reliability and redundancy like "Soft RAID".
<p>106</p> <p>du df</p>	<p>104.2</p> <p>du - Prints the disk usage for a directory and its subdirectories. df - Reports the amount of free disk space available on all mounted filesystems.</p>
<p>107</p> <p>fsck e2fsck</p>	<p>104.2</p> <p>fsck - Filesystem checker for the appropriate system type to check and repair unmounted filesystems. If a filesystem is consistent, the number of files, number of blocks used, and number of blocks free are reported. e2fsck - Similar to fsck, but specifically intended for Linux Second Extended Filesystems (ext2).</p>
<p>108</p> <p>mount umount</p>	<p>104.3</p> <p>mount - Mounts a file structure. umount - Unmounts a file structure.</p>
<p>109</p> <p>/etc/fstab</p>	<p>104.3</p> <p>/etc/fstab – is the configuration file or table used during boot time to verify and mount each Filesystem listed.</p>
<p>110</p> <p>usrquota grpquota</p>	<p>104.3</p> <p>Used to set quotas on the system for both users (usrquota) and groups (grpquota).</p>

<p>111</p> <p style="text-align: center;">edquota quota quotaon quotaoff quotacheck repquota</p>	<p>104.4</p> <p>edquota is used to set the quota limits and to modify the user and group quotas. quota is used to display quota limits set for both the user and group limits. quotaon is used to enable the correctly configured Filesystem disk quotas. quotaoff is used to disable the Filesystem disk quotas. quotacheck is used to examine and initialize or compile the quota database. Once the quota settings have been set in the /etc/fstab file they need to be accessed and create a data base with the information for the system to monitor the disk use by users and groups. repquota was a Unix command to summarize quotas on a Filesystem. It is no longer used.</p>
---	---

<p>112</p> <p style="text-align: center;">File Attributes chattr</p>	<p>104.4</p> <p>Here are all of the attributes listed in the Bonus flags box for the ext2 and ext3 Filesystems:</p> <ul style="list-style-type: none"> • A Don't update access time on Modify • a Append only, can't be cleared • c Compressed • d No dump (backup) • i Immutable, can only be cleared by the superuser. • j Journal file (like a transaction tracked file) • S Synchronous updates • s Secure deletion, can't be restored once deleted. • u Undeletable, can be restored after deleted. <p>chattr - changes the attributes for a file. (chattr +ds for example)</p>
---	---

<p>113</p> <p style="text-align: center;">umask</p>	<p>104.5</p> <p>umask - User file creation mask. Limits the default file attributes for a particular user. All files created by that user take on the attributes specified by umask. The umask value is usually read from the /etc/profile file. The octal value passed to umask defines the file permissions <i>disabled</i>. For example, umask 022 ensures that new files will have at most 755 permissions (777 NAND 022) umask command - The umask command displays the umask value for the current user.</p>
--	--

<p>114</p> <p style="text-align: center;">Initial Access Mode (IAM)</p>	<p>104.5</p> <p>Initial Access Mode (IAM) is created at the Partition creation time. A file is set to 666, Directory is 777. Umask is the filter that takes away from the IAM.</p>
--	--

<p>115</p> <p style="text-align: center;">File Permissions</p>	<p>104.5</p> <p>Permissions are broken down into three entities: User, Group, and Other. User is the owner of the file (usually the one who created it); Group is the default group of the owner; and Other is everyone else. There are three permissions for each of them, r = read, w=write and x=execute.</p>
<p>116</p> <p style="text-align: center;">What are the two ways to identify permissions?</p>	<p>104.5</p> <p>There are two ways to identify permissions: Symbolic (rwxrwxrwx) and Octal (777). Octal is represented by each of the 3 bit r(4)w(2)x(1) as 4+2+1=7.</p>
<p>117</p> <p style="text-align: center;">chmod chown chgrp</p>	<p>104.5 104.6</p> <p>chmod - used to make changes to permissions on files and directories. chown - used to change the owner of a file or directory. chgrp - used to change the group which owns a file or directory. The chgrp command is older and not used much seeing how the chmod can do all of the same tasks now.</p>
<p>118</p> <p style="text-align: center;">Hard and Symbolic Links</p> <p style="text-align: center;">ln</p>	<p>104.7</p> <p>hard link - Hard link is not really a “link” but a copy of a directory entry with a different name but pointing to the same inode number. They too, do not take up additional disk space by duplicating the data to another location. The major limitation here is that both hard links must point to the same filesystem. Another limitation is that they cannot point to a directory</p> <p>symbolic link - (some call it a soft link) is a very small file in a directory, and thus an entry in the directory table, that has a pointer that points to the real file that contains the actual data. There will be two entries in the directory table but they both will have the same inode number, thus pointing to the same file location on the disks.</p> <p>ln - command used to create links.</p>

<p>119</p> <p>find locate slocate whereis which</p>	<p>104.8</p> <p>find – will look for a command or file, starting at the location you tell it. It is powerful in that it will look Recursively from wherever you start it. It is weak in that it will only look in your current working directory if no start directory is provided.</p> <p>locate and slocate – both require a database to search in. slocate is now deprecated.</p> <p>whereis – finds the binary, source and manual pages of a command. Also needs a database to search in and does an exact match on keywords.</p> <p>which – its job is simple, find the executable that that the shell is asking for. It ONLY searches the user's \$PATH settings. This is very helpful is you are upgrading or trying a new kernel, module, package or whatever.</p>
<p>120</p> <p>modules /etc/conf.modules (Redhat) or /etc/modules.conf</p>	<p>105.1</p> <p>A kernel can be operated in two basic methods. First, by compiling every driver and feature into the core kernel. Second, the core kernel can be compiled as lean as possible and then “load” and “unload” compiled drivers as needed, even when it is up and running. These added capabilities are known as modules.</p> <p>Modules may also be loaded, not compiled to the kernel, during the boot process. The configuration file that is used to accomplish this will be one of these two, depending on the distro used: /etc/conf.modules or /etc/modules.conf</p>
<p>121</p> <p>lsmod insmod modprobe modinfo uname</p>	<p>105.1</p> <p>lsmod - prints a list of all loaded modules with its name, size, use count, and a list of referring modules.</p> <p>insmod - inserts a module into the running kernel.</p> <p>modprobe - is used to insert modules. It has the ability to load a single module, modules and their prerequisites, or all the modules stored in a specified directory.</p> <p>modinfo - displays information about a module.</p> <p>uname - prints out system information. The options –s is for name, Linux; -r is the release number and –a gives you the complete information.</p>

122

dependencies
modules.dep
depmod

105.1

Modules might have dependencies. In other words, they will not operate correctly unless some other module, shell, command or what ever, has been previously identified, set and loaded. All dependencies are entered into a file called modules.dep. All modules available to the kernel are listed in this file.

depmod - This command creates or overwrites the modules.dep file each time it runs. depmod is run each time the system boots and every time a change is made in the module dependency has changed.

123

Steps for Kernel Installation,
Update or Revision

105.2

1. Obtain a makefile from one of these sources: Delivered with the source code or Run ./configure to create a makefile file
2. Run make (with one of these targets: config, menuconfig, xconfig or oldconfig) to create .config file
3. Run the make command with each of these formats: Make dep, Make clean, Make bzImage, Make modules, Make modules_install

124

make dep
make clean
make bzImage
make modules
make modules_install

105.2

make dep - This command examines each source file (those with a .c extension) and looks in the header for dependencies. It then creates a file in each directory to enter all objects (.o extension) line by line that are dependencies.

make clean - Removes the old output files from the previous compile.

make bzImage - This creates the bootable kernel image in compressed format.

make modules - This command compiles the device drivers as modules.

make modules_install - This command installs all the compiled modules in the kernel.

125

dmesg
/var/log/dmesg
/var/log/messages
What are the differences in
these files?
When would you need to look at
each file?

106.1

During the boot process, many messages flash across the screen that are near impossible to read. /var/log/dmesg and /var/log/messages are files that save the messages displayed during the boot process.

dmesg - Displays basically messages about the hardware detected at boot-time and stores those in the /var/log/dmesg logfile.

/var/log/messages - Gives information about system processes that loaded or failed to load at boot-time.

- For hardware detection issues: use **dmesg**

- For process failure to load issues: use **/var/log/messages**

126

Linux Runlevels init

106.2

Linux uses several runlevels which are used not only for a “safe, single-user mode” but to shut down and reboot as well. The first daemon to load after the kernel is the init daemon. A runlevel establishes which sub-set of daemon processes will be loaded by the init daemon after the kernel loads. While logged in as the superuser, you may manually set the runlevel at the command prompt using the init command.

Runlevel - Purpose

- 0 - System Halt. Administrators use this for a quick system shut down.
- 1 - (also designated as runlevel s or S) Single-user or “safe mode” without loading network interfaces or file sharing. Used as a maintenance mode.
- 2 - Multiuser but without NFS file sharing
- 3 - Extended Multiuser Mode. The full multiuser level. The default runlevel for the init process. Does not load the X windows system.
- 4 - Unused
- 5 - Graphical Mode. Full multiuser with GUI login. Is runlevel 3 plus the launch of the X windows system. If the X windows system fails for lack of correct video driver, for instance, the system will revert to runlevel 3.
- 6 - Reboot. Would bring a system down and back to GUI login. If runlevel 6 were ever placed in the init script, it would cause a permanent loop of rebooting.

127

/etc/inittab

106.2

/etc/inittab - The file which sets the default runlevel used in the boot process is /etc/inittab (for “initialization table”). It contains a line specifying the default runlevel:
id:n:initdefault (where n is the runlevel, such as 5)

128

telinit

106.2

There is a command telinit, which is a link to init. It is not listed under the LPI objectives but may be used on the exams, so learn it as an alternate for init.

telinit

/sbin/telinit is linked to **/sbin/init**. It takes a one-character argument and signals init to perform the appropriate action. The following arguments serve as directives to telinit:

0,1,2,3,4,5 or 6 - tell init to switch to the specified run level.

a,b,c - tell init to process only those /etc/inittab file entries having runlevel a,b or c.

Q or q - tell init to re-examine the /etc/inittab file.

S or s - tell init to switch to single user mode.

U or u - tell init to re-execute itself (preserving the state). No re-examining of /etc/inittab file happens. Run level should be one of Ss12345, otherwise request would be silently ignored.

telinit can also tell init how long it should wait between sending processes the SIGTERM and SIGKILL signals. The default is 5 seconds, but this can be changed with the -t sec option.

129

shutdown

106.2

Shutdown - The shutdown command unloads processes and brings the system down in a safe manner. It defaults to putting the system into single user mode, but options can be used to cause a system halt or reboot:

Shutdown [options] time [hh:mm or "now"] [message]
The time argument is mandatory and cannot be omitted.

130

lpq
lpc
lprm

107.2

lpq - Issuing lpq will query all print queues and return a status report of active print jobs.

lpc - Used to control print jobs by changing their order, suspending or canceling them. lpc stands for line printer control.

lprm - To simply remove a print job from the queue, use the command lprm (line printer remove) with the ID of that print job.

<p>131</p> <p style="text-align: center;">lpr</p>	<p>107.3</p> <p>lpr - Sends files to the print spool to print. If no file is given, lpr sends stdin to print spool. The lpr command takes care of all the initial work needed to print the file, and then it hands control over to another program, lpd, the line printing daemon.</p> <p>On LPI exams, you will likely need to know how to send a job to print multiple copies. That is</p> <pre># lpr -# [number of copies] filename</pre> <p>The range of the of # parameter is 1-100. You may also need to know that you can redirect the stdout of a command to the printer. So, for example, if you were reading the lpr man pages and you wanted to print two copies of them, the command could be:</p> <pre># man lpr lpr -#2</pre>
<p>132</p> <p style="text-align: center;">lpd /var/spool/lpd</p>	<p>107.3</p> <p>lpd - The line printing daemon then tells the printer how to print the file. So when a job is sent to a print queue, the lpd, which was started at boot time, is already listening for requests</p> <p>/var/spool/lpd - The lpd's printer spool directory is by default located at /var/spool/lpd</p>
<p>133</p> <p style="text-align: center;">mpage</p>	<p>107.3</p> <p>mpage - mpage could be thought of as "many pages", Its purpose is to take longer documents and print multiple pages on a single sheet of paper.</p>
<p>134</p> <p style="text-align: center;">/etc/printcap</p>	<p>107.4</p> <p>The lpd daemon uses a printer definition file when it is spooling print jobs to the queue. That file is called the printcap or printer "capabilities" file. This file provides the system name for the printer, the physical location of its print spool file, and a filter, which defines how to format a print job using escape codes appropriate to that printer's unique command set.</p>
<p>135</p> <p style="text-align: center;">/etc/apsfilter/* /var/lib/apsfilter/*/ /etc/magicfilter/*/</p>	<p>107.4</p> <p>Two common filters, referenced as "magic filters," are the APSfilter and magicfilter. Each print queue listed in the /etc/printcap file can be associated with a different print filter using the if command. Study the following example of entries in a printcap file to see the association of the apsfilter with the laserjet printer:</p> <pre>lp ljet5/ :sd=/var/spool/lpd/lp:\br/>:lp:=/dev/lp0 :if=/usr/lib/apsfilter/filter^</pre>

136	<p><code>/etc/hosts.equiv</code> <code>/etc/hosts.lpd.</code></p>	107.4 Used to control remote access to the printers on the machine containing this file."
137	<p><code>/etc/ldap.perms</code></p>	107.4 Used for remote printing security, when the machine running lpd gets a request from a remote client program, it performs the checks specified by the rules in the lpd.perms to decide whether to accept or reject the request.
138	<p>Printing to Samba Printers</p>	107.4 To configure printing to a remote printer that is located on a Windows network require a Samba share (see Section 113.4). On the Windows side, they need to have that printer properly shared; on the Linux side we need to have Samba running and properly configured.
139	<p>man pages man [section#] command How many man sections are there?</p>	108.1 man pages - documentation in the form of electronic "manuals". A manpage may provide commands, dependency files, libraries, shells, languages, and/or devices. Manpages are divided into sections 1-9 and N: <u>Section - Description</u> 1 - User commands (executable programs or shell commands) 8 - System administration commands (usually only root can execute) 2- Linux System calls (functions provided by the kernel) 3 - Library calls (functions within program libraries) 4 - Special device files (usually in the /dev directory) 5 - File formats and conventions, e.g. /etc/passwd 6 - Games 7 - Miscellaneous (macro packages and conventions), e.g. man(7), groff(7) 9 - Kernel routines (Non standard) N - New commands not yet given a category These sections may be accessed directly by giving the section number after the man command; for instance, man 5 smb would take you into the file formats section for the Samba daemon.
140	<p>MANPATH</p>	108.1 MANPATH - environment variable used to construct a search path for all man files. Manpages are stored in /usr/doc. If a manpage is located in an alternate directory, the MANPATH variable needs to be updated to find it. The MANPATH is configured in the file /etc/man.config, but you can add a directory to the MANPATH by exporting a new one: # export MANPATH=/usr/man:/usr/local/man

141	apropos	108.1	apropos <name> Searches man page headings and returns all man pages who's headings have <name> in it.
142	info howto	108.1	info - Utility designed to replace man. Info uses a system named Textinfo, which utilizes a single source file to display information on the screen with a browser type of interface. Like the man command, info returns a description of a command and includes hyperlink references to other sources of information for that command. Example: # info smb howto - Procedures on to how to accomplish certain administrative procedures in Linux are written as HOWTO's and are stored in /usr/doc. They also include FAQ's
143	Where on the internet would you find Linux documentation? What is the USENET Newsgroup for Linux?	108.2	www.linuxdoc.org (also www.tldp.org - The Linux Documentation Project) Newsgroup is: comp.os.linux
144	getty process	108.5	getty process - The init daemon starts the getty process which displays the contents of the /etc/issue file and prompts the user for a login name.
145	/etc/issue /etc/motd /etc/issue.net	108.5	/etc/issue - a login welcome message that you wish to display to all users /etc/motd - "Message Of The Day"; works like the /etc/issue file. It is used to display current messages to all users such as "Please logout by 5 p.m. tonight." /etc/issue.net - a text file which contains a message or system identification to be printed before the login prompt of a telnet session.
146	export export -f export -n	109.1	export - Once variables are assigned to the login shell, variables can be exported to child sub-shells with the export command. export -f - Is used to export a function. export -n - Is used to stop the exporting of the variable.

<p>147</p> <p style="text-align: center;">exit code</p> <p style="text-align: right;">109.1</p>	<p>Exit Code - When a command is executed an exit code is sent to the parent shell and is stored in the shell variable <code> \$? </code>. If the command executed successfully, the exit code is zero. If not, the exit code will be a number from 1 through 255, meaning that an error occurred.</p>
<p>148</p> <p style="text-align: center;">Imported Shell Variables:</p> <p style="text-align: center;"> <code> \$\$ </code> <code> \$0 </code> <code> \$# </code> <code> \$* </code> <code> \$? </code> <code> \$! </code> </p> <p style="text-align: right;">109.1</p>	<p>Here are some common shell variables used in shell scripting:</p> <ul style="list-style-type: none"> \$\$ PID \$0 Shell script name \$# Number of arguments passed to the shell script \$* All command line arguments passed to the shell script \$? Exit values \$! PID of the last background process (echo <code> \$! & </code>)
<p>149</p> <p style="text-align: center;">Functions</p> <p style="text-align: right;">109.1</p>	<p>Functions - Functions are like aliases except that they contain a more complex set of instructions than just a command. They also can create values for variables and store them in the shell for later use and quick execution. Functions operate faster than commands or aliases. Their command structure is like "C" programming:</p> <pre>Function-name () { Shell commands}</pre>
<p>150</p> <p style="text-align: center;">SUID</p> <p style="text-align: right;">109.2</p>	<p>There are times when it is necessary for a user to have superuser permissions in order to execute a file properly. In most instances, however, you do not want to give out the root user's password. SUID provides this capability. Every user executing a program that is owned by root and has the SUID set will be run as if they are the owner, root.</p>
<p>151</p> <p style="text-align: center;">What is the name of the default X Server in Linux?</p> <p style="text-align: right;">110.1</p>	<p>XFree86 is the default X Server in Linux.</p> <p><u>NOTE:</u> The term "server" here does not need to refer to a separate, specialized piece of hardware or a dedicated server; it may refer to one of the many applications running on a single Linux server. In that sense, an X server and an X client could be one and the same machine (typical for us today), or could be physically separated from one another across a network as in the case of one Linux server and many X terminals running across a network.</p>

152	<p>What is the function of XFree86 X Server?</p> <p>What is the function of the X Client?</p>	<p>110.1</p> <p>The function of the XFree86 X server is to provide the proper graphical input for a given set of video and monitor hardware.</p> <p>The function of the X client is to format and render the display meaningful to the user.</p>
153	<p>What is a window manager?</p> <p>What does it do?</p>	<p>110.1</p> <p>window manager - provides the standard function and behavior of a window, such as size, buttons to open and close, copy and paste, etc.</p>
154	<p>desktop environments</p> <p>What are some examples of desktop environments?</p>	<p>110.1</p> <p>desktop environments - combine a windows manager with a set of utilities and applications in a unified environment.</p> <p>We know these as GNOME, KDE, and the Ximian desktop environment (which includes a window manager), but you will see them referred to simply as a window managers or on the exam as an X display manager.</p>
155	<p>What information is stored in the XF86Config file?</p>	<p>110.1</p> <p>XF86Config file- stores information about the chipset on your video card (note: you do not choose a video card; you choose a chipset), the keyboard specifications, the monitor's frequency and resolutions available, and the fonts to support.</p>
156	<p>What program is used to edit the XF86Config file? (/etc/X11/XF86Config)</p>	<p>110.1</p> <p>The program that writes the XF86Config file is XF86Setup. This is the program to run if your X windows session does not run or cannot run because you have changed video or monitor hardware, for instance. XF86Setup runs a graphical screen to make appropriate setup selections, and runs natively with most video devices. NOTE: xf86config (text based) and xf86cfg (graphical with block diagrams) also write the XF86Config file!</p>
157	<p>The default font path is located in what section of the XF86Config file?</p>	<p>110.1</p> <p>The font path is in the Files section of XF86Config, not a "Fonts" section, which does not exist.</p>
158	<p>What is the default font server?</p>	<p>110.1</p> <p>xfst is the default font server. This is a daemon process typically running on your Linux server to make fonts available to clients accessing that server either on the server itself or remotely.</p>

<p>159 What program would you use if the lines on your monitor are not straight or image is not centered and the knobs on your monitor don't help? 110.1</p>	<p>xvidtune</p>
<p>160 .Xresources and .Xdefaults 110.1</p>	<p>NOTE: Symbolically linked to point to same X application configuration. Located in a user's home directory; Various x applications are configured with a variety of resources, such as foreground and background colors, initial screen sizes, screen geometry, and initial font. Rather than configure each of them separately, applications look at this file.</p>
<p>161 xdm /etc/inittab 110.2</p>	<p>The default X display Manager, xdm, is usually loaded from the /etc/inittab, which either starts xdm in runlevel 5 or calls another script, rc5, to start runlevel 5 processes</p>
<p>162 How can you stop xdm?` 110.2</p>	<p>xdm may be stopped with CTRL-ALT-BKSP or by typing killall xdm on the command line.</p>
<p>163 .xinitrc 110.2</p>	<p>Running xdm is actually contained in the startx script, and startx looks to the user's home directory for the file .xinitrc to see which default windows display manager to start for that user (e.g., KDE, Gnome, etc.)</p>
<p>164 Where is the background color for your X display initialized? 110.2</p>	<p>Background color is specified in .Xresources, but the background itself as in a wallpaper or graphic and/or a screensaver is contained in the /etc/X11/xdm/Xsetup</p>
<p>165 What is the startup process for loading the X Windows Manager? 110.4</p>	<p>Startx -> xinit -> either the system-wide /etc/X11/xinit/Xinitrc script or a user's .xinitrc (the user's .xinitrc being read last takes precedence over the system Xinitrc)-> Xclients (to see which applications and window manager to run)</p>

166	DISPLAY environment variable xhost xauth	110.4	<p>DISPLAY environment variable - Allows a remote display of its X clients on another machine. To do this, you export your DISPLAY environment variable. The DISPLAY environment variable has the following syntax:</p> <pre>#export DISPLAY=hostname:displaynumber.screennumber</pre> <p>xhost - The xhost command allows another user to display to your machine.</p> <p>xauth - like xhost, but use xauth for more secure access, and/or to deny access to your machine.</p>
167	/etc/group /etc/gshadow	111.1	<p>/etc/group - Group information in plain text.</p> <p>/etc/gshadow - Encrypted group information file only readable by the root user.</p>
168	pwconv pwunconv	111.1	<p>pwconv - command converts the passwd file to a shadow file. Both files are maintained but the passwords are now encrypted in shadow and read only by root.</p> <p>pwunconv - reverses this process.</p>
169	useradd userdel userdel -f username usermod	111.1	<p>useradd - creates users</p> <p>userdel - removes users.</p> <p>userdel -f username - removes user and their home directory of files.</p> <p>usermod - modifies other user information, like home directory and locking their password.</p>
170	chage	111.1	<p>chage - User's password aging (I.e. when a password expires) can be changed using the chage command.</p> <p>NOTE: Breaking the name down to "ch-age" will help you remember this command.</p>
171	groups groupadd groupdel groupmod grpconv grpunconv gpasswd	111.1	<p>groups - command tells the user what groups they belong to.</p> <p>groupadd - Create new groups</p> <p>groupdel - Delete groups</p> <p>groupmod - Change group ID, name and some override settings</p> <p>gpasswd - Set the group password</p> <p>grpconv - Converts standard group passwords in /etc/passwd to encrypted passwords in the group shadow file /etc/gshadow.</p> <p>grpunconv - Converts encrypted group passwords in /etc/gshadow back to plain passwords in the /etc/passwd file.</p>

172	/etc/skel	<p>/etc/skel - Skeleton file; There are a number of files that are local to the user and usually in their home directories. Some of these files are: .Xdefaults, .bash_logout, .bash_profile, .bashrc, .kde, .kderc. These files are automatically transferred to a newly created users home directory from the /etc/skel file. The skeleton file contains many files that are basically templates with minimal commands and functions but provide a file for other programs to add other commands and information to as needed.</p>
173	/etc/syslog.conf	<p>/etc/syslog.conf - The /etc/syslog.conf configuration file has not only instructions and aides for helping configure the collection of data but also has the capability of blocking data from being recorded or being read. In one case you could block the login facility by isolating that system and giving it a simple login priority of NONE. Each entry in the syslog.conf file has three parts: facility.level action</p>
174	/var/log/wtmp /var/run/utmp /var/log/lastlog	<p>These files are used by such commands as who, w, and finger. /var/log/wtmp - a binary data log file that lists when users logged-in and their current status like if they are still logged-in or not. /var/run/utmp - is another binary data log file containing only logged-in users. /var/log/lastlog - also a binary data log file, like wtmp, containing information when a user was last logged in.</p>
175	faillog /var/log/faillog	<p>/var/log/faillog - one of those commands that has a name that is the same as the file it accesses for the information. The /var/log/faillog file contains both the login settings as well as a log of all system user login attempts. It is used to tell which users have had any amount of failed login attempts.</p>

176

Log File Rotation logrotate

111.3

There are a number of log files in the `/var/log` directory monitoring many procedures and programs that can grow to enormous sizes. Log rotation allows the log files to be cycled or copied out to a file by a new name like `/var/log/messages.1`. A facility is usually set up in all distros to rotate the log files so the working one does not become unmanageable in size. This command is called `logrotate` and is configured by `/etc/logrotate.conf`. The config file has specifications for determining when to rotate, what files to rotate, when to compress and so forth.

177

cron crond crontab

103.1
111.4

cron - A program that executes commands at scheduled times. The file that specifies what this schedule and commands are is `/var/spool/cron`.
crond - The `crond` daemon is the process that starts up when the system initializes and then wakes up every 60 seconds and examines the `crontab` files to execute any commands that are scheduled at that time.
crontab - `crontab` command is the cron table manipulation, view and edit program. It is confusing because of its name. It is the same name as the cron tables it reads, `/etc/crontab`. As it reads this file, every minute, it executes each scheduled item if the it's time is due.

178

`/var/spool/cron`

111.4

`/var/spool/cron` - You may be wondering how `cron` can manage to find all these `crontab` files for users that are scattered around the system. They aren't actually scattered; they are all in the same location, the `/var/spool/cron` directory.

179

crontab fields

111.4

`crontab` fields - There are 6 total fields, and of those, 5 that specify the time. They are:
 minute hour day month day of week
 command
 All fields need to be represented in each entry. If there are no entries for a field the `*` needs to be there as a place-holder. Spaces are required between the `*` or field entries.

180

`/etc/cron.allow` `/etc/cron.deny`

111.4

These two files allow or deny access to run or execute scheduled cron entries.

- if the `allow` file exists, only those listed may use the service;
- if the `allow` does not exist but the `deny` does exist only those not on the `deny` list may use it;
- if neither file exists all users can use the service.

181	at atq atrm	111.4	<p>at - a command that is used to just execute the commands or script at a later date, meaning only once. at is basically saying, "run it at...".</p> <p>atq - means check the jobs-to-run in the queue and display them.</p> <p>atrm - means delete a specific job to prevent it from running in the future.</p>
182	anacron	111.4	<p>anacron - The anacron utility utilizes the anacrontab config file and runs continuously. In other words, it does not go to sleep and wake up like cron so it does not miss any scheduled tasks if the time adjustment takes a jump to update.</p>
183	Using tar with backup files	111.5	<p>tar - The tar command is the standard services to not only backup systems but to take specific directories and files and put them in one file to transfer. Compression is an option but by default it does not compress files. Its main advantage is to have only one file to deal with rather than a whole filesystem of files. Tar can also be used to move directory trees by piping dump to restore.</p>
184	dump	111.5	<p>dump - The dump utility is used to backup Filesystems usually to tape. Dump will backup any Filesystem type, even though the man reference says ext2!</p>
185	restore	111.5	<p>restore - The restore utility is used to restore the filesystem that backed up using the dump command. One of the unique features of restore is that you can use it in an interactive mode (-i)and select the files to be restored one by one.</p>
186	cpio	111.5	<p>cpio - Is a very old utility that is similar to tar in backing up, moving complete directories or Filesystems. It also varies greatly among different distros. cpio can also be used to move directory trees. Here is an example of doing just that instead of backing up to tape: Find fromdir -depth -print cpio -pdm todir</p>
187	dd	111.5	<p>dd - Is a file copying and conversion command. Somewhat like the od command that converts data. dd will convert, add and delete data from almost any data type like ASCII to EBCDIC. If you had a tape that was written on some other system than Linux the dd command may be the only way to read it. It was also used to copy tapes.</p>

188	<p style="text-align: center;">System Time Server (NTP)</p> <p style="text-align: center;">ntpd ntpdate ntp.conf</p>	<p>NTP protocol - The NTP protocol is the standard for maintaining time synchronization among your network of servers.</p> <p>ntpd - the service that sets the time upon boot up and may continue to access the system time servers to constantly update the time clock.</p> <p>ntpdate - The purpose of the ntpdate command is to substitute for the ntpd daemon. If the daemon is not set to run on a continuous mode the ntpdate command can be entered into the cron config file (usually into the crontab configuration file) to continue the time sync process.</p> <p>/etc/ntp.conf - NTP configuration file used to identify the servers used to synchronize clocks on the system time servers from a standard accurate clock. Entry into this configuration file looks like: server ntp-s2.uom.edu</p>
189	<p style="text-align: center;">resolv.conf</p>	<p>resolv.conf - File that points to one of more DNS servers.</p> <p>NOTE: In Linux, configuration files typically have the .conf extension, whereas other operating systems use resolv.cfg. Remember that difference.</p>
190	<p style="text-align: center;">What are the Private IP addresses?</p>	<p>The IP standard defines specific address ranges within Class A, Class B, and Class C reserved for use by private networks (intranets). The table below lists these reserved ranges of the IP address space.</p> <p>Class A; Private Start Address - 10.0.0.0 Private Finish Address - 10.255.255.255</p> <p>Class B; Private Start Address - 172.16.0.0 Private Finish Address - 172.31.255.255</p> <p>Class C; Private Start Address - 192.168.0.0 Private Finish Address - 192.168.255.255</p>
191	<p style="text-align: center;">/etc/HOSTNAME</p>	<p>/etc/HOSTNAME - This file holds the name of the host and its domain on a single line: Example: sles tailwind.com</p> <p>NOTE: If networking had not been configured, the entry would read: noname nodomain nowhere</p> <p>On some implementations, this is lower-case /etc/hostname.</p> <p>To view the hostname in this file, use the command hostname.</p> <p>To permanently change the hostname, edit the file /etc/sysconfig/network.</p>

192	<code>/etc/hosts</code>	112.3	/etc/hosts - ASCII file is used to list IP addresses and text names. It can be used to convert local or private (host) names to the IP addresses on the internal network. This file is suitable for use only on small networks because it must exist on every host in the network and must constantly be updated when each host is added to the network. On larger networks or those where internet browsing of a large number of destinations, the hosts file is not used and the DNS (Domain Name System) service instead.
193	<code>/etc/networks</code>	112.3	/etc/networks - This file lists your own and the other known networks with which the host can communicate. It is used by the route command to specify a network by name rather than by IP address.
194	<code>/etc/host.conf</code>	112.3	/etc/host.conf - This file tells the order in which this system will attempt to resolve host names into IP addresses, such as using hosts file first and then a DNS name server.
195	<code>/etc/nsswitch.conf</code>	112.3	/etc/nsswitch.conf - Like hosts.conf, it provides the order of services to use to provide name resolution, and it can be used for authentication resolution also.
196	<code>arp</code>	112.3	arp - Can be used to see the entries in the Address Resolution table, which maps network card addresses (MAC addresses) to IP addresses. You can use arp to see if the IP addresses you know should be in the table are in fact there and if they are mapped to the computers you know they should be.
197	<code>hostname</code> <code>domainname</code>	112.3	<code>hostname</code> - A simple utility, it shows the name of the host. <code>domainname</code> - Similar to hostname, this utility shows the domain name used by the host.
198	<code>dig</code>	112.3	<code>dig</code> - A powerful Linux utility to perform DNS lookups on domain names. It requires DNS servers for information about domain names.

199	ftp telnet	112.3	<p>ftp - This file-transfer utility, operating on the File Transfer Protocol, is used to send files back and forth between a local and remote host.</p> <p>telnet - Allows you to establish a connection with a remote host. This is often used for administrative purposes.</p>
200	ifconfig	112.3	<p>ifconfig - This utility is used to configure the TCP/IP parameters on the command line. The syntax is <code>ifconfig interface options</code>. Thus, to configure the IP address of 192.168.1.1 for the first Ethernet card, the command would be <code>ifconfig eth0 192.168.1.1</code> (and this would use the class C default subnet mask of 255.255.255.0).</p> <p>To perform the same configuration but with a subnet mask of 192, the command becomes <code>ifconfig eth0 201.13.12.65 netmask 255.255.255.192</code>.</p>
201	netstat	112.3	<p>netstat - This command shows the statistics of the TCP and UDP protocols. Executing <code>netstat</code> without switches displays protocol statistics and current TCP/IP connections.</p> <p>netstat can also provide information about routing tables, and interfaces, similar to what can be obtained from <code>/sbin/route</code> and from ifconfig.</p>
202	ping	112.3	<p>ping - Packet Internet Groper. This utility sends echo messages to a host to see if it is reachable. If it is, the ICMP protocol on the remote host echoes the response. If it is not reachable, the resulting error message indicates such.</p>
203	pump	112.3	<p>pump - utility used to obtain a new IP address from a DHCP server. Pump is a combined BOOTP and DHCP client daemon, which allows your machine to retrieve configuration information from a DHCP server.</p>
204	route	112.3	<p>route - This utility is used to see and configure routing. If you give the <code>route</code> command with no parameters, it displays the current routing table. To add a route, use the <code>route add</code> command. This may be used to add a static route to a specific network, host, such as a default gateway.</p>

205	tracert	112.3	<p>tracert - This is a much-enhanced version of the ping utility, and is often used in place of it. Not only does it show that the remote host is reachable, it also reveals the path that was taken to reach the host.</p>
206	whois	112.3	<p>whois - Used for a domain name lookup</p>
207	ppp	112.4	<p>ppp - A PPP (Point-to-Point Protocol) connection would be used to establish a network connection between two hosts using a serial cable or phone line and a dial-up connection.</p> <p>The Sequence of Establishing a PPP Connection:</p> <ol style="list-style-type: none"> 1. A client running PPP creates a PPP connection with a remote host. 2. The local client sends user account and password information to the remote host PPP server. 3. At this point, PPP is started on the client. 4. The PPP server offers the client an IP address from a range of addresses available in its pool of addresses. 5. The client uses the IP information from the PPP server to setup its default route to the server on its routing table and then joins the data stream from the server.
208	pppd /etc/ppp/ip-up script persist	112.4	<p>pppd - The pppd daemon starts a PPP session on a device with a serial speed interface. The format for using this command is: pppd [device] [baud rate] [options]</p> <p>/etc/ppp/ip-up - Once the PPP link is up and established, the pppd looks for /etc/ppp/ip-up script. If this script exists and is executable, the PPP daemon executes the script ip-up which provides for the automation of any special routing commands that may be necessary and any other actions that you want to occur every time the PPP link is activated.</p> <p>persist - used to reestablish a terminated PPP connection when one has been lost.</p>
209	"Chat Script" chat	112.4	<p>"Chat Script" - Automates the sequence of establishing and negotiating modem communications. The Linux utility chat executes a script that follows the instructions to be sent to the remote modem, as well as the responses expected from it.</p> <p>EXAM NOTE: It is possible that the exam asks you about either the meaning of "chat script," or about typical commands found within it.</p>

210	<p>PPP used for ISDN connections HiSax</p>	<p>112.4</p> <p>When PPP is used over an ISDN connection, Password Authentication Protocol (PAP) may be used and the <code>/etc/pap-secrets</code> file may be used to provide your name and password for automated authentication.</p> <p>HiSax - Linux driver for ISDN card; HiSax is a Linux hardware-level driver for passive ISDN cards with Siemens chipset. Most all ISDN card manufacturers now include support for the HiSax driver.</p>
211	<p>WvDial wvdailconf <code>/etc/wvdial.conf</code></p>	<p>112.4</p> <p>WvDial - A utility used for a system to dial up an ISP to establish a PPP connection. (NOTE: notice its case SeNsiTivty.)</p> <p>wvdailconf - Configuration utility for WvDial.</p> <p>wvdialconf also is used first to probe and discover a modem.</p> <p><code>/etc/wvdial.conf</code> - wvdial configuration file. WvDial will invoke wvdialconf (a process) which then creates a configuration file, <code>/etc/wvdial.conf</code>.</p>
212	<p><code>/etc/ppp/peers</code> directory <code>/etc/ppp/peers/provider</code> file</p>	<p>112.4</p> <p>This file was often provided by the ISP to tell your system how to connect uniquely with them. Hence, you would have one peers file for every commercial dial-up connection you used. The default is often called <code>/etc/ppp/peers/provider</code>.</p> <p><u>NOTE:</u> Almost no reference books discuss it this file, but it did play a prominent role in dial-up connections with ISP's when those were popular.</p> <p><u>EXAM NOTE:</u> This file has NOTHING TO DO WITH MACHINES THAT ARE PEERS TO EACH OTHER ON A LOCAL NETWORK –do not be misled by such a distracter!</p>
213	<p>inetd superdaemon</p>	<p>113.1</p> <p>inetd - To reduce the number of daemons required to service requests for various internet services, Linux runs a superdaemon named inetd. This one process "listens" on all the ports used by the various services and channels requests to the appropriate process.</p> <p>inetd is usually started during the boot process by the init daemon and runs continuously.</p>
214	<p><code>/etc/inetd.conf</code></p>	<p>113.1</p> <p><code>/etc/inetd.conf</code> - This configuration file is read by the superdaemon inetd and is a plain text file used to determine which services inetd will be managing.</p> <p><u>NOTE:</u> This file exists on every Linux system, but its lines may be commented out if it is not used to manage anything.</p>

215 What are the fields within the /etc/inetd.conf file and what are their meanings?

113.1

<service name> <service name> <proto> <flags> <user>
<server_path> <argument>

<service name> - name of networking service which is listed in the /etc/services file.

<service name> - Defines one of the communications this service will use, which is typically either stream or dgram.

<proto> - Protocol, which is either tcp or udp, corresponding with stream or dgram socket types respectively.

<flags> - If the socket type is dgram for datagram services, this flag controls the handling of inbound requests. The flag is set to either wait or nowait (default is wait). If the socket-type is stream, this setting is nowait. The wait flag can be used with a dot and a maximum number of server instances spawned by inetd within a sixty second interval. If no maximum number is given, the default is 40 instances within one minute.

<user> - Designates the user account the service should run under. Typically, this is root, but if fewer permissions are needed, the user can be user "nobody."

<server_path> - Path to the executable daemon on the server. When TCP wrappers are used, this path just shows "tcpd."

<argument> - Lists all arguments that are to be passed on to the daemon.

216 How do you put the changes to the /etc/inetd.conf file into effect?
SIGHUP

113.1

Any changes to the /etc/inetd.conf will not be put into effect until and unless the **inetd** is signaled to reread its control file (**SIGHUP**) or restarted. The **SIGHUP** command often is given with a **kill** command like this:
#killall -HUP inetd
(or **#kill -1 inetd** would also accomplish the same thing).

217 TCP Wrappers
/etc/hosts.allow
/etc/hosts.deny
/usr/sbin/tcpd

113.1

To add more security to the inetd's controls of the services it manages, all requests can be verified against /etc/hosts.allow and /etc/hosts.deny access control files, IF TCP wrappers are enabled. It is very important to note that the contents of the hosts.allow file is read first and then the hosts.deny file.

The hosts.allow file specifically lists other hosts that are allowed access to the service(s) listed and running from /usr/sbin/tcpd services on that machine. Likewise, specific hosts listed in hosts.deny cannot access services running from /usr/sbin/tcpd services.

218	Email System	112.3	An email system, in very summary form, has two back-end components and one front-end component. The end user uses a mail client on the front-end to send mail either within his or her mail system or externally to other mail systems. The user's client typically uses an IP connection to deliver mail to or from an intermediary known as the Message Transfer Agent or MTA.
219	Message Transfer Agent (MTA) sendmail	113.2	The user's client typically uses an IP connection to deliver mail to or from an intermediary known as the Message Transfer Agent or MTA. The MTA transfers the mail either for internal delivery within their mail system or transfers it to an SMTP (Simple Mail Transfer Protocol) gateway to the internet. Sendmail is an MTA.
220	intentionally left blank		
221	intentionally left blank		
222	Starting and stoping the sendmail process	113.2	The Sendmail process is started with the command: # /etc/init.d/sendmail start <or stop or restart> or /usr/sbin/sendmail start Once started, it runs continuously, always checking its queues for inbound and outbound mail to process.
223	/etc/sendmail.cf	113.2	/etc/sendmail.cf - This file is the sendmail configuration. This file controls the processing of mail and is only read upon startup of sendmail; hence, changes to sendmail.cf require a restart of sendmail.
224	/etc/aliases /etc/aliases.db newalias	113.2	/etc/aliases - For raw aliases of local users/accounts. The syntax is alias: user. This is a simple mapping of one or more user names to a recipient. For example: postmaster: root /etc/aliases.db - The database for aliases. Sendmail does not actually read the aliases file. It reads instead the compiled version of that file stored as aliases.db newalias - The aliases.db is updated by running the process newalias, which must be run as root.
225	/var/state/sendmail.st	113.2	/var/state/sendmail.st - Statistics about operations

226	<code>/var/spool/mqueue</code>	113.2	<code>/var/spool/mqueue</code> —The location of the mail queue
227	How do you avoid relaying and spam from your mail system?	113.2	The “no relaying” parameter, which prevents outside users from “co-opting” your SMTP gateway and using it to forward or “relay” their mail as if it were coming from your system), is also contained within the <code>sendmail.cf</code> file.
228	<code>.forward</code> file	113.2	As an alternative to the alias database, any user may put a file with the name <code>.forward</code> in his or her home directory. If this file exists, sendmail redirects mail for that user to the list of addresses listed in the <code>.forward</code> file.
229	The Mail Queue	113.2	Sometimes a host cannot handle a message immediately. For example, it may be down or overloaded, causing it to refuse connections. The sending host is then expected to save this message in its mail queue and attempt to deliver it later.
230	How do you print the list of the mail in the queue?	113.2	The contents of the queue can be printed using the <code>mailq</code> command (or by specifying the <code>-bp</code> flag to <code>sendmail</code>). This will produce a listing of the queue id's, the size of the message, the date the message entered the queue, and the sender and recipients.
231	<code>/var/spool/mqueue</code> <code>/var/spool/mail</code>	113.2	<code>/var/spool/mqueue</code> - The mail queues are typically located at <code>/var/spool/mqueue</code> <code>/var/spool/mail</code> - Once sendmail processes mail in its queue, delivered mail is held for users at <code>/var/spool/mail</code> for RedHat and SuSE.
232	<code>httpd</code>	113.3	<code>httpd</code> - Apache web server daemon; . Apache is the pre-eminent web server across many platforms (Microsoft, Novell, and Linux), and is preferred for its performance, security, and stability. Running on Linux, Apache is known as the HTTP daemon or <code>httpd</code> . You start and stop it with: <code>#/etc/init.d/httpd <start, stop, or restart></code>

233	apachectl	113.3	<p>apachectl is an alternate startup script for Apache, which may be located in /usr/local/sbin/ where its commands are:</p> <p>apachectl stop</p> <p>apachectl startssl (to start Apache with SSL for secure https support)</p> <p>apachectl restart</p>
234	/etc/httpd.conf	113.3	<p>/etc/httpd.conf - This file contains general information about the name of the Apache administrator, the user name it runs under, how it handles its log files, its default port, etc.</p>
235	srm.conf	113.3	<p>srm.conf – This file includes local parameters about your web site, such as the location of the /docs directory which is the root of the html tree, the path to its cgi extensions, the languages it supports, etc.</p>
236	access.conf	113.3	<p>access.conf – This is the main security file that can control access to selected pages on this server from client browsers</p>
237	Network File System (NFS)	113.4	<p>Network File System (NFS) is the protocol through which partitions may be mounted and shared with other Linux hosts. Linux uses three different daemons – primarily, nfsd, but portmap, and also rpc.mountd. nfsd will load partitions that are specified in the /etc/exports file.</p> <p>The nfsd is started and stopped with: nfsd <start, stop, and restart></p>
238	/etc/exports	113.4	<p>/etc/exports file - Specifies which partitions to be loaded by NFS. This file exists on each Linux host, but will be empty until it is used to share a partition and the rights to it.</p>
239	SMB (Server Message Block) smbd /etc/smb.conf	113.4	<p>SMB (Server Message Block), or Samba, is the protocol that is used for sharing file resources on a Windows-based network.</p> <p>smbd is the Samba daemon running on a Linux machine to support file and print sharing with authentication to SMB clients.</p> <p>smbd <start, stop, and restart> are the usual command to start and stop the Samba process.</p> <p>/etc/smb.conf - The smbd relies upon the /etc/smb.conf file for its configuration, to determine what, where, and how to share.</p>

240 NetBIOS Message Block (nmb)

113.4

NMB - Samba allows file sharing, but if the NetBIOS names are used to identify Windows-based computers, Samba by itself cannot resolve those names. Samba uses the NetBIOS Message Block (nmb) as a WINS proxy to contact a WINS server to identify and resolve Windows-based names. So, there is the nmbd daemon that is loaded and used to support both NetBIOS Name Service and WINS, the Microsoft implementation of NetBIOS Name Server. It is important to know that in this sense, nmb and WINS share the smb.conf file with Samba.

241 Domain Name Server (DNS)
named
named.conf

113.5

DNS (Domain Name Services) - a distributed database of domain names-to- IP addresses, used to resolve domain names for such purposes of servicing requests from a web browser or an email server to find the IP address for a given domain name so that web contents may be displayed or email delivered across the internet.
named - The DNS daemon that runs locally to provide DNS services. The name daemon (hence, named and pronounced name-dee) reads its /etc/named.conf file for its proper configuration (i.e., whether it is a local caching-only, or an authoritative primary or secondary DNS server with its own database).

242 /etc/nsswitch.conf
/etc/resolv.conf

113.5

When a “resolver” (services requiring name-to-IP translation) deems that a DNS query is needed, it first consults the /etc/nsswitch.conf file to determine the order of the methods of name server to be used (e.g., use the hosts file first, then use a dns server), then it reads the /etc/resolv.conf file to determine the IP address of the first dns nameserver it should query. Then, if that DNS server cannot resolve the name, it may “escalate” the query to another DNS server in its hierarchy of name servers.

243 SSH (Secure Shell) Client
sshd

113.6

SSH (Secure Shell) - This is a program for logging into a remote machine and for executing commands on a remote machine. Ssh replaces the older rlogin and rsh, and is used to provide secure encrypted communications between two untrusted hosts over an insecure network. .
sshd - This is the Secure Shell Daemon.

244 **/etc/ssh/ssh_known_hosts**
 What permissions should this file be given? 113.6

/etc/ssh/ssh_known_hosts - A system wide list of known host keys. This file should be prepared by the system administrator to contain the public host keys of all machines in the organization. This file should be world-readable, but writable only by root, the owner. This file contains public keys, one per line

245 **/etc/passwd** 114.1

/etc/passwd - When a user is created they automatically get entered into a file called passwd. Each entry in /etc/passwd consists of:
 username : password : UID : GID : user's name : home directory : default shell
 Most of these entries are self-explanatory except the last one, the default shell. This entry tells Linux to start the user in a specific type of a shell, like bash. One way to temporarily disable a user from logging in is to point to a file called /bin/false.

246 **/etc/shadow** 114.1

/etc/shadow - To have user passwords encrypted, rather than plain text in the /etc/passwd file, the password in the passwd file will be replaced with an "x" and the password will be encrypted and moved to the /etc/shadow file. The shadow password file is only readable by the root user.

247 **passwd command**
/etc/passwd file 114.1

passwd - A user's password may be created and or changed with the passwd command. Again, here is a command with the same name as its configuration file, /etc/passwd. Some options are:

- passwd -l tux can lock the user account
- passwd -u tux unlock the account
- passwd -d tux delete the password
- passwd -S tux Shows status of the users password (upper case S)
- passwd -e tux force the password to be changed the next login
- passwd -f tux change the full name, phone, and other info

248 **shutting down services not in use** 114.2

One of the first and most effective steps to Linux security is shutdown the network services that are not needed. This can be accomplished by using the netstat -an command. It will show what protocol is used, what activity has been seen in both send and receive, the local address being used (port address), and the state it is in, like LISTEN. You also can view the current services by using the chkconfig command.