Chapter 6

Administrator Applications

This chapter introduces the user to some of the commands that they may utilize in the normal administration of a Unix / Linux host. The normal user does not necessarily use these commands, as they deal with the general administration. The command requirements necessary for setting up a server will be covered in Chapter 10.

These utilities, in combination with those of Chapter 9, Data Manipulation, provides a strong foundation for commands that an administrator may utilize to control a system. A basic knowledge of scripting (covered minimally in Chapter 19 – Programming and Scripting) should be studied by the student.

Concepts Learned in this Chapter

- Administrator Utilities to maintain system operation

Table of Contents

6.1 System Processes

Whenever a Unix or Linux system starts an application, we say that it initiates a **process**. We then keep track of each process by a unique number, called the **process id**. Many processes are started when we boot Linux, so we normally observe a number of processes that are operating in the background.

Some applications may be operating multiple times, each with its own process id, commonly referred to as its **pid**.

6.1.1 ps Utility

To display the active processes, we use the command:

ps

```
# ps
  PID   TTY        TIME      CMD
10431   pts/0      00:00:00  bash
10449   pts/0      00:00:00  ps
```

This will give us a short list. Most of the process will be running in the background, performing various functions for the general operation of the system – but there will be a set of which apply to specific active applications. Background processes are often referred to as a **daemon**.

pid	The identifier assigned to the process.
tty	The terminal it is running on. Up to eight different terminals may be in use (multi-user). Processes which are required to operate Linux are on terminal "**?**".
stat	Process State Code D Uninterruptible sleep R Runnable (on run queue) S Sleeping T Traced or Stopped Z A defunct ("zombie") process
time	Time the process has been running. If the process is sleeping, then it may not indicate any time accrued.
command	Process name.

The command **ps aux** provides a far more complete listing of processes. Using only the **ps** command gives a very short list of only those processes that are current. The "aux" gives us all processes that are present, and details that are far more detailed. An application whose name ends in a "d" is a daemon, running in the background. If we wish to see a complete list of all processes that are running, we issue the command: **ps aux**

```
# ps aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.1    500   244 ?        S    Oct18   0:05 init
root         2  0.0  0.0      0     0 ?        SW   Oct18   0:00 [keventd]
root         3  0.0  0.0      0     0 ?        SW   Oct18   0:00 [kapmd]
root         4  0.0  0.0      0     0 ?        SWN  Oct18   0:00 [ksoftirqd_CPU0]
root         5  0.0  0.0      0     0 ?        SW   Oct18   0:00 [kswapd]
.....s
```

This gives us a listing of all processes in a user-friendly format that are not controlling the tty's. The display includes the PID, %CPU usage, and %memory.

A few of the many options for the ps utility include:

- t Select all process associated with this terminal.
- a Select all process except session leaders and processes not associated with a terminal.
- r Restrict the session to only running processes.
- x List all processes owned by the user.
- u Display user oriented format.
- e Display the environment after the command.

Many lines have been deleted in order to shorten the listing, but one can gain the basic understanding of the listing that considerable information is provided.

6.1.2 top Utility

The **top** command provides the same information as "ps", but adds an additional excellent feature, a continuously updated status of the memory status SWAP usage, which is continuously updated every 10 seconds.

```
# top
top - 18:46:25 up 5 days,  1:16,  4 users,  load average: 0.00, 0.00, 0.00
Tasks: 107 total,   1 running, 106 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.0% us,  0.0% sy,  0.0% ni, 99.7% id,  0.0% wa,  0.3% hi,  0.0% si
Mem:    515388k total,   502516k used,    12872k free,    86388k buffers
Swap:  1020116k total,      192k used,  1019924k free,    93376k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
    1 root      16   0  1692  552  476 S  0.0  0.1   0:00.82 init
    2 root      34  19     0    0    0 S  0.0  0.0   0:00.07 ksoftirqd/0
    3 root      10  -5     0    0    0 S  0.0  0.0   0:00.01 events/0
    4 root      13  -5     0    0    0 S  0.0  0.0   0:00.00 khelper
    5 root      10  -5     0    0    0 S  0.0  0.0   0:00.00 kthread
    7 root      20  -5     0    0    0 S  0.0  0.0   0:00.00 kacpid
  133 root      10  -5     0    0    0 S  0.0  0.0   0:00.00 kblockd/0
  136 root      15   0     0    0    0 S  0.0  0.0   0:00.00 khubd
  182 root      20   0     0    0    0 S  0.0  0.0   0:00.00 pdflush
  183 root      15   0     0    0    0 S  0.0  0.0   0:00.22 pdflush
  185 root      11  -5     0    0    0 S  0.0  0.0   0:00.00 aio/0
  184 root      15   0     0    0    0 S  0.0  0.0   0:03.92 kswapd0
  272 root      15   0     0    0    0 S  0.0  0.0   0:00.00 kseriod
  427 root      15   0     0    0    0 S  0.0  0.0   0:00.47 kjournald
 1392 root      10  -5  1592  468  384 S  0.0  0.1   0:00.03 udevd
 2091 root      15   0     0    0    0 S  0.0  0.0   0:00.00 kjournald
 2752 root      16   0  1600  608  512 S  0.0  0.1   0:00.57 syslogd
. . . . .
```

Many lines have been deleted.

6.1.3 vmstat Utility

The **vmstat** command provides a report of the virtual memory statistics. The information provided includes processes, memory, paging, block IO, taps, and cpu activity.

```
# vmstat
procs -----------memory---------- ---swap-- -----io---- --system-- ----cpu----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in    cs us sy id wa
 0  0    192  13920  86160  93384    0    0     4     5    5    89  0  0 99  0
```

Processes (Procs)
- r Number of processes waiting for run time.
- b The number of processes in uninterruptible sleep.
- w The number of processes swapped out.

Memory (in KiloBytes)
- swpd Amount of virtual memory used.
- free Amount of idle memory.
- buff Amount of memory used as buffers.
- cache Amount of memory that has been reserved for cache.

Swap (in KiloBytes per second)
- si Amount of memory swapped in from disk.
- so Amount of memory swapped to disk.

IO (in blocks per second)
- bi Blocks sent to a block device.
- bo Blocks received from a block device.

System
- in Number of Interrupts per second, including the clock.
- cs Number of context switches per second.

CPU (Percentages of total CPU time)
- us User Time.
- sy System Time.
- id Idle Time.

6.1.4 pstree Utility

The command **pstree** is an interesting take-off from the **ps** listing. Instead of showing all of the statistics, it provides a tree display of which process were initiated by what.

```
# pstree
initâ─┬─_plutorun─┬─_plutoload
      │           └─plutorun─pluto─┬─_pluto_adns
      │                            └─pluto
      ├─acpid
      ├─artsd
      ├─atd
      ├─bonobo-activati
      ├─clock-applet
      ├─crond
      ├─cups-config-dae
      ├─cupsd
      ├─2*[dbus-daemon-1]
      ├─dbus-launch
      ├─eggcups
      ├─events/0
```

```
├─gam_server
├─gconfd-2
├─gnome-keyring-d
├─gnome-panel
├─gnome-settings-
├─gnome-terminal─┬─bash
│                └─gnome-pty-helpe
├─gnome-vfs-daemo
├─gnome-volume-ma
├─gpm
├─hald
├─httpd───8*[httpd]
├─khelper
├─khubd
├─2*[kjournald]
├─klogd
├─kseriod
├─ksoftirqd/0
├─kswapd0
├─kthread─┬─aio/0
│         ├─kacpid
│         ├─kblockd/0
│         └─2*[pdflush]
. . .
```

6.2 Killing a Process

Unfortunately there are times (even with Unix and Linux) that a process will lock up. At this time we need to terminate the process. This is accomplished through the **kill** command. There is a range of levels that kill can take as an option, typically 1 through 9 (others are available). For options 1 - 8, an application may be able to exempt itself from being killed, depending upon its level of priority. All applications or utilizes run at a level of 8 or below, thus killing a process at a level 9 will absolutely terminate the process. Recall that each application has a **processor ID (pid)** associated with it. Previously, the PID was found by issuing the command:

ps aux

This returns the pid of the process / utility.

Alternatively, another command, **pidof** can be used to return the pid if the utility name is known.

The syntax of the command is:

kill [options] pid

where pid is the Process ID and the one option that is commonly used insure termination:

– 9

Prior to issuing the kill command, the administrator must determine what the process id is for the desired locked process. This is done using the **ps aux** command. Once the pid is known, the kill command may be issued to terminate

the process. Of course, the administrator may have to change to another terminal (ALT-FX) to obtain the pid.

Of course, there are times when the process just does not want to die, no matter how many times you issue the kill command. As a last resort (and maybe even the first), you could alternatively issue the command:

kill –9 pid

One might view the "9" as a person standing up to be executed – well that is one way to remember the format! For sure, the process is now terminated.

There is a situation where a process may appear to not be killed, they are still active upon re-running the ps command. In fact, the application has been killed, but has been restarted, the process has been call re-spawned. This is required to insure that some processes are maintained.

6.3 Modifying File Permissions

When using DOS, if we type **dir,** we see a list of files with the filename, last opened date, and size of the file. To learn the attributes, we need to use the **attrib** command, which displays whether the file is general, read-only, or hidden.

Recall that we previously learned that Unix and Linux provide enhanced attributes that may be configured by either the owner or the **root** administrator. If you run the command **ls –l**, you will observe something like:

drwx ------	**2**	**root**	**root**	1024	**Jan 26**	**00:32**	**mail**
-rw-r--r--	**1**	**root**	**root**	345	**Apr 5**	**01:07**	**Xrootenv.0**
\----------/							
permissions	links	user	group	size	date	time	filename

This displays all non-hidden files in full-length form.

6.3.1 File Permissions

The first group of letters and dashes provides the file type and the operational attributes. Look at the sequence as one character plus three sets of 3 characters:

f r w x r w x r w x
\ user / \ group / \ other / field grouping

f **File type**
– normal file that may be read, written to, or executed
d directory file, a file that specifies additional files within a directory
l link to another file

The three groups of letters have like meaning, each with a different function. The **r w x** provides permission for:

r read - the file may be opened for reading
w write - the file may be written to
x execute - the file is executable

As appropriate, the **r w x** permissions are set with either a **0** or a **1**, where a 0 means the action is not allowed, and a 1 means it is allowed. A 0 will be displayed as a "**–**", a 1 is displayed as an **r, w,** or **x** as appropriate.

The first **r w x** field specifies the rights of the **file's owner**, or the **username** of the person that created the file. If you log on using your username and create a file, you will observe your username as the owner and group name.

The second **r w x** field specifies the rights of users belonging to the specified **group name.**

The third **r w x** field specifies the rights of everyone else, commonly referred to as the world or **other**.

6.3.2 Modifying Permissions

To change the user attributes, we issue the command:

chmod xyz {filename}

Where **x**, **y** and **z** each represent the octal equivalent of the 1 / 0 binary combination of the rwx number set. This is a simple binary – octal conversion.

r	w	x	octal	Operational Status
0	0	1	1	file is executable only
0	1	0	2	file is writeable only
0	1	1	3	file is writeable and executable
1	0	0	4	file is read only
1	0	1	5	file is readable and executable
1	1	0	6	file is readable and writeable
1	1	1	7	file is readable, writeable, and executable

For example if we wish to set the following to **afile**:

	Permissions	Binary	Octal
User/Owner	r w x	1 1 1	7
Group	r w -	1 1 0	6
Other/World	r - -	1 0 0	4

Then use the command **chmod 764 afile**.

Are there alternative methods of specifying the attributes of file rather than using the octal value? Of course, Unix and Linux always try to provide multiple methods. One still needs to use the **chmod** command, but the format may be altered. If one specifies the group that is to be modified, then a given permission may be modified. The format is:

chmod u+rwx g+rw o+r afile

This is the same as was shown just above. One or more groups may be specified. The meaning for the above is:

- **u** User or Owner of the file or directory
- **g** Group users
- **o** Other or World users
- **+** Adds a persmission
- – Deletes a permission

Examples might be:

chmod g +rx	Adds read and executable to the group
chmod g –x	Removes executable from the group
chmod o +r	Adds read to the others / world
chmod u +x	Adds executable from the user / owner

6.3.2.1 File Links (1)

The number following the rwx attributes specifies the number of links that point to the file. An alternative name (alias) may be created that points to the real file.

6.3.2.2 File Owner Name

The name that follows is the owner of the file, given as the login name.

6.3.2.3 Group Name

The second name is the group name of users that share access to the file. Alternative to a name, it may be a Group ID (gid) numeric value.

6.3.2.4 File Size

The following number is the size of the file in bytes.

6.3.2.5 File Date

The next set of information is the date and time that the file was last modified. If the file was last modified some time ago, then the year will be displayed rather than the time. Two of the file dates may be modified using the **touch** command.

6.3.2.6 File Name

Finally, the name of the file is displayed.

6.3.2.7 Displaying Hidden Files

If you want to display all files, including those that are hidden, use the command:

ls –la.

6.3.2.8 Displaying Child Directories Only

If you wish to only list directories, use the command:

ls –F * | grep /.

6.3.3 Changing the File's Owner

The owner of a file may be changed to another user by issuing the command:

chown new-owner filename

If we look at a file's full attributes with ls –l myfile, we might observe:

-rwxrw-r- - 1 root root 50 Apr 10 8:08 myfile

If we issue the command chown jdoe myfile, then ls –l again, we will have:

-rwxrw-r- - 1 jdoe root 50 Apr 10 8:08 myfile

This assumes that the user jdoe exists.

6.3.4 Changing the File's Group

The group name of a file may be changed to another group name by issuing the command:

chgrp new-group-name filename

Following from the above, issuing the command chgrp public myfile, we would will have:

–rwxrw-r- - 1 jdoe public 50 Apr 10 8:08 myfile

Again, the group public must exist.

6.3.5 Special Attributes

There are three additional attributes that a file can take on. When we issue the chmod command, you often see is issued with four numbers rather than the previous three that we have discussed so far. The fourth, or left hand digit, provides additional features. Each permission entity, owner, group, and world, have a fourth bit that can be set. There are several ways that these can be viewed. The following discussion provides a means to understand how the bits relate to the change modification (chmod) process.

Previously, it was specified that each permission entity was made up of three bits. In fact, it is made up of four, but the specification of the fourth bit is treated differently. View the permission bits in the following way:

This is the relationship for each entity.

This is the way the fourth bit is designated.

When the special attribute is displayed in the set of permissions, it appears as:

r w $\frac{a}{x}$ **r w** $\frac{b}{x}$ **r w** $\frac{c}{x}$

where	if **x = 0:**	if **x = 1:**
a	**displayed as s**	**displayed as S**
b	**displayed as s**	**displayed as S**
c	**displayed as t**	**displayed as T**

6.3.5.1 User ID

The Set User ID, often specified as **Set UID** or **SUID**, applies to files that another user may utilize. It allows a user (who is not the owner) to inherit the rights of the owner. The file's attributes will be denoted as either of the following:

4 7 5 5 or

r w s r – x r – x

It may be set by issuing either one of the following commands:

chmod u+s filename or
chmod 4755 filename

If the execute bit has been set for the user / owner, then the “x” character is replaced with a lower case “s”. If the execute bit has not been set, then the “x” is an “S”.

6.3.5.2 Group ID

In like manner, a user may inherit the executable rights of the file’s group. Adding this to file does not really mean much, but applying it to a directory is more useful. Any file created in a directory created with the Group ID (SGID) set, will automatically inherit the directory’s group name. A file’s (directory’s) is denoted as:

2 7 5 5 or
r w x r – s r - x

It may be set by issuing either one of the following commands:

chmod g+s filename or
chmod 2755 filename

If the execute bit has been set for the group, then the “x” character is replaced with a lower case “s”. If the execute bit has not been set, then the “x” is an “S”.

This option is best described as making the directory shareable.

6.3.5.3 Sticky Bit

The last option is to have a program remain in memory after the program has finished. This is also known as "Terminate Stay Resident". This provides faster execution if it is to be executed again. Additionally, by setting the Sticky bit, a file may be set such that any user may not delete it; if a directory is set to allow one to write to a file, then a user may also delete it. By setting the Sticky Bit, the file may be written to but not deleted. A file is denoted to remain in memory as:

1 7 5 5 or
r w x r – x r - t

The sticky bit is set by issuing either one of the following commands:

chmod o+t filename or
chmod 1755 filename

If the execute bit has been set for the world, the “x” character is replaced with a lower case “t”. If the execute bit has not been set, then the “x” is a “T”.

6.3.6 Default Permissions

When a file is first created, it has default permissions. These permissions are set by a factor called the **umask**. Issuing the command provides the default response:

$ umask
0022

For a starter, do not take the value as shown, because it is changed for several reasons. The rules are:

1. The normal attributes are 0644 by system default at installation.
2. By default, no file is executable. It must be changed manually.
3. Special Attributes are always set to zero.

To initially understand the creation of the umask, ask the question, when a file is created, what are the default permissions that one wants? Typically, as an owner, one wants to have read / write, groups typically has read only, and the world also has read only. To put this in binary format, including the special attributes, it would be:

0 0 0 1 1 0 1 0 0 1 0 0 or **0 6 4 4**

In binary, this value is inverted. Thus the value is:

1 1 1 0 0 1 0 1 1 0 1 1 or **7 0 3 3**

Now accepting the rules specified previously, where the special attributes are set to zero and the permission to execute is not allowed, the above is modified to:

0 0 0 0 0 0 0 1 0 0 1 0 or **0 0 2 2**

Note that the special attributes were modified back to all zeros, the owner's permissions were not modified, but that the group and world permissions were reduced by one. This is probably the hardest part to understand. What else can be said, but that is the way it is.

So when a file is created, the umask is again inverted to create the default values of the file permissions. This results in:

1 1 1 1 1 1 1 0 1 1 0 1

Again, we use the rules that the special attributes are not set, and that no file is executable, so the final result is:

0 0 0 1 1 0 1 0 0 1 0 0 or **0 6 4 4**

6.3.7 Fixed Attributes

In addition to the general visible attributes specified above, there is one set of attributes that are not directly visible. These may be set only by the host administrator, and once set, may not be over-written unless changed back to the normal state. Using this feature, a file or directory may be set to allow only information being appended, not be modified at all, dates may not be modified, and prevent the file from being deleted. These features are unique to Linux only, not supported by Unix.

6.3.7.1 chattr[1] Utility

A file has additional "back-end" attributes that can be used to prevent it from being overwritten. This command is Options include:

- –a Append only, file cannot be overwritten
- –i File Immunity, file cannot be changed in any way. The file cannot be deleted.
- –A Do not change the file's last access time
- –S Update the file synchronously
- –c Kernel automatically compresses / decompresses file
- –d File can not be dumped with the **dump** command
- –s File will be deleted securely using a special secure deletion algorithm
- –u File can not be deleted
- –j s – Available only on ext3 type file system

Other attributes exist, some of which are not yet supported. For a full list, one needs to check the man or info pages.

To change these attributes, you issue the command:

chattr +/-option filename

To set an attribute, you place a "+" in front of the option. To remove the option, precede it with a "-".

6.3.7.2 lsattr

To view the attributes that have been set, issue the command:

lsattr [option] filename

This will display the file's attributes in the form of:

– – – – – – – – – – – – – – filename
s u S i a d A c – – – – j

The four remaining dashes are reserved for future use.

The assignment of the values are defined from the chattr command:

- –a Append only, file cannot be overwritten
- –i File Immunity, file cannot be changed in any way. The file cannot be deleted.
- –A Do not change the file's last access time
- –S Update the file synchronously
- –c Kernel automatically compresses / decompresses file
- –d File can not be dumped with the **dump** command
- –s File will be deleted securely using a special secure deletion algorithm
- –u File can not be deleted
- –j s – Available only on ext3 type file system

Options for the **lsattr** utility include:

- –R Recursively list attributes of directories and their contents.

[1] Red Hat Linux Networking and System Administration, Red Hat Press (Wiley Publishing)

–V Display the program version.
–a List all files in directories, including files that start with `.'.
–d List directories like other files, rather than listing their contents.
–v List the file's version/generation number.

6.3.8 Listing Locked Files

There are some applications that only allow one instance to be run at a time. Most commonly, the **yum** application will only allow itself to be run once for either updating or installing applications. Other instances occur for a serial interface, such as the com (serial) port or Ethernet port. If one has ether of these in service, then another instance is not allowed to run as a conflict would exist.

When a given application is started, it creates an entry in the **/var/lock** directory. An entry here will prevent another instance of the application from starting. Unfortunately, if an application is improperly terminated, the lock may not be removed. This will require that the lock be manually terminated. The simple method is to delete the entry.

To list locked files, issue the command:

```
$ lslk
SRC              PID DEV    INUM       SZ TY M ST WH END LEN NAME
sendmail        1925 3,3  992041       33  w 0  0  0   0   0
/var/run/sendmail.pid
sendmail        1933 3,3  992042       49  w 0  0  0   0   0
/var/run/sm-client.pid
atd             2018 3,3  992048        5  w 0  0  0   0   0
/var/run/atd.pid
system-config-n 5445 3,3 1120006 62328832  r 0  0  0   0   0
/var/lib/rpm/Packages
```

As noted, one can learn what files are locked and thus know what needs to be removed.

If the **lslk** application is not installed, then use the **yum** utility to install it.

6.3.9 Managing Filesystems [2]

Every file system needs to be managed for optimum performance. The following additional commands are available.

badblocks	Searches the specified storage device for bad blocks.
chattr	Sets special attributes of a file that are viewable only with the lsattr command.
dosfsck	Checks a DOS filesystem.
dumpe2fs	Lists the superblocks and blocks of the specified device. Functions only on ext2 or ext3 Linux filesystem.
fdformat	Performs a low level format to a specified floppy disk. Various densities may be specified.
fdisk	A very jpowerful disk partition manipulation program. Linux version supports nearly all other partition types, including vendor proprietary partitions.
fsck	An application (similar to MS Scandisk) used to check or repair a file system. Typically used as a periodic basis for an ext2fs or after a power interruption.

[2] Mark Allen, www.comptechdoc.org/os/linux

hdparm	Displays or sets various parameters of a hard drive.
lsattr	Displays special attributes of a file that are not viewable with the stat command.
mkdosfs	Creates a MSDOS file system on a specified partition.
mke2fs	Creates a Linux ext2 file system on a specified partition.
mkswap	Creates a Linux swap file system on a specified partition.
mount	Mounts a partition file system to the designated mount point.
smartctl	Displays the status of the specified hard drive.
stat	Displays all attributes of a file except those set by the chattr command.
swapoff	Used to de-activate a swap partition.
swapon	Used to activate a swap partition.
tune2fs	Used to adjust filesystem parameters on a Linux filesystem.
umount	Unmounts a removable disk storage filesystem.

The user needs to refer to the man pages for details regarding these commands if not previously covered.

6.4 File System Utilities

In order to keep track of a hard drive's utilization, two commands are available. They are:

1. df
2. du

6.4.1 df – Filesystem Usage

It is necessary to keep track of a hard drive's utilization. As a hard drive is used, more and more file and directories are created. At some point we may need to install an additional drive to add capacity. Tracking of the partition utilization is provided with the command **df**.

The output of the df command lists each partition on the system. From the display, one observes:

Filesystem	Specifies the partition name and path.
1K-blocks	Specifies the number of 1024 blocks on the filesystem.
Used	Specifies the number of 1024 blocks that are used with directories or files.
Available	Specifies the total remaining blocks that are unused.
Used%	Specifies the percentage of blocks that are used.
Mounted on	Specifies the mount name of the partition.

To view the existing file system, issue the command df:

```
$ df
Filesystem   1K-blocks      Used  Available  Use%  Mounted on
/dev/hda3     37357128  11136196   24323256   32%  /
/dev/hda1       101086     30692      65175   33%  /boot
none            257692         0     257692    0%  /dev/shm
```

6.4.2 du – Filesystem Structure

It is very often important to learn the size of a directory of file. An easy way to display this information is with the command **du**. This outputs a two-column report of the present directory's contents and all sub-directories.

It is recommended that you not perform this from the root directory – you will list the whole system. If the listing is extensive, you may wish to pipe the output to the **less** command to provide a display that you may scroll through.

One of the most important uses is to determine is how much hard drive space is used by each of the users on your system. Here you would change to the /home directory that contains the home directories for each of your users, and specifies how much space each is using. Although you will still be required to do some scanning through the listing, you will be able to derive the present size of each users directory.

Many options are available, which the user is referred to the man pages.

```
$ du
4       ./.gnupg/.gnupg
72      ./.gnupg
8       ./.gnome2_private
12      ./.gftp/cache
44      ./.gftp
8       ./.macromedia/Macromedia/Flash
    Player/#SharedObjects/5Y5BZFSZ/logicalsecurity.com/education/education.swf
12      ./.macromedia/Macromedia/Flash
    Player/#SharedObjects/5Y5BZFSZ/logicalsecurity.com/education
16      ./.macromedia/Macromedia/Flash
    Player/#SharedObjects/5Y5BZFSZ/logicalsecurity.com
8       ./.macromedia/Macromedia/Flash
    Player/#SharedObjects/5Y5BZFSZ/internetnews.com
16      ./.macromedia/Macromedia/Flash
    Player/#SharedObjects/5Y5BZFSZ/ebags.com/is-viewers/flash/genericzoom.swf/#ebg
20      ./.macromedia/Macromedia/Flash
    Player/#SharedObjects/5Y5BZFSZ/ebags.com/is-viewers/flash/genericzoom.swf
24      ./.macromedia/Macromedia/Flash
    Player/#SharedObjects/5Y5BZFSZ/ebags.com/is-viewers/flash
28      ./.macromedia/Macromedia/Flash
    Player/#SharedObjects/5Y5BZFSZ/ebags.com/is-viewers
32      ./.macromedia/Macromedia/Flash Player/#SharedObjects/5Y5BZFSZ/ebags.com
60      ./.macromedia/Macromedia/Flash Player/#SharedObjects/5Y5BZFSZ
64      ./.macromedia/Macromedia/Flash Player/#SharedObjects
8       ./.macromedia/Macromedia/Flash
    Player/macromedia.com/support/flashplayer/sys/#logicalsecurity.com
8       ./.macromedia/Macromedia/Flash
    Player/macromedia.com/support/flashplayer/sys/#ebags.com
8       ./.macromedia/Macromedia/Flash
    Player/macromedia.com/support/flashplayer/sys/#internetnews.com
32      ./.macromedia/Macromedia/Flash
    Player/macromedia.com/support/flashplayer/sys
36      ./.macromedia/Macromedia/Flash Player/macromedia.com/support/flashplayer
40      ./.macromedia/Macromedia/Flash Player/macromedia.com/support
44      ./.macromedia/Macromedia/Flash Player/macromedia.com
112     ./.macromedia/Macromedia/Flash Player
. . .
```

6.5 Formatting an ext2 Floppy Disk

When you need to format a floppy disk in the default Linux format, as **ext2,** you need to issue the command:

fdformat /dev/fd0H1440

This will create a Linux formatted disk – which can not be read by Windows. Although you can format a disk for Windows also under Linux, it is easier to do such on a Windows machine.

fd0 specifies floppy drive 0 (the first drive, or drive A: in Microsoft).

H1440 specifies that the formatting is to be high density with 1440 Kbytes.

6.6 System Analysis

After your system has been installed and is operational, it is generally advantageous to know what IRQs and I/O Ports are assigned. This is a similar to the requirement as found in Microsoft Windows.

6.6.1 CLI Mode

In order to determine what the Interrupts and IO Ports are that have been assigned to your system, change to the **/proc** directory. Two files are available to review the assignments. DO NOT edit these files, only view it!

less interrupts and
less ioports

6.6.2 GUI Mode

When using the KDE, this can be done using the following process:

1. Click on the ***Menu Button***
2. Click on ***System Tools***
3. Click on ***Info Center***
4. Click on ***Information***
   a. Click on ***Interrupts***
   b. Click on ***IO Ports***

When configuring the system for sound, it is a good idea to check this file, because you will need to know what values to use – better than the trial and error method.

Another command that can be utilized to learn how the system is configured is **sysinfo**.

6.6.3 uname Utility

We are able to determine specific information about our system by using the **uname** utility. The syntax of the command is:

uname –option

Several options are available to specify several or all information. These are:

-a	Display all attributes.
-m	Display the hardware name.
-n	Display the ***node*** name.
-p	Display the processor type.
-r	Display the OS release number.
-s	Display the OS name.
-v	Display the OS version number.

6.7 User Environment

Every user is provided a shell operating area, known as the environment. This environment defines or specifies various default variables that other commands utilize. You are able to modify these values or add new ones that are applicable to your personal requirements. Although the default values are the same for each user, each user may modify them to their own desire, thus making them unique.

To display one's environment, you issue the command:

env

and you obtain a listing of the values that look something like the following:

USER=me
HOME=/home/me
PATH=/usr/bin:/bin:/usr/local:/sbin
SHELL=/bin/bash
…

Normally this list is longer than the screen, so you are best to pipe the output through the **less** command.

You may also display individual environmental values by using the **echo** command:

echo $HOME

If you determine that there is a situation where you could utilize an environmental value, but it does not exist, then you may create one. Creating the new variable is accomplished by the following process:

variable=value Note: Do not place spaces around the equal sign.
export variable

For example, we might wish to set pico (nano) as the default editor. Issue the following commands:

$ EDITOR=nano
$ echo $EDITOR
nano
$ export EDITOR
$ env | grep EDIT
EDITOR=nano

First we have defined the variable by assigning a value, and then we export it to reside in the environmental listing. The above example is valid for the bash shell, other shells may accomplish this by a different procedure.

One of the more important values listed in the environment is the user's path. Quite often one desires to expand the list of directories that one can have immediate access to. A change to the path may be made in one of two methods, directly to the environment and indirectly via one's profile. As an example, assume we wish to append the directory /data directly to the

environment to our existing path; the following commands at the command line interface are issued:

$ echo $PATH
/usr/bin:/bin:/usr/local:/sbin
$ PATH=$PATH:/data
$ export PATH

The above is not the desired method of appending a value to the user's environment. The preferred method is to modify a user's profile. This may be accomplished by the user or globally to all users.

For an individual user, edit the individual's profile, change to their home directory and edit the **.bash_profile** file. Append to the end:

PATH=$PATH:/desired_path
export PATH

To modify the environment for all users, edit the **/etc/profile** file. Append to the end of the file:

PATH=$PATH:/desired_path
export PATH

6.8 System Memory

As a Network Administrator you should be aware of the each system operational performance. There are a number of utilities available to monitor performance.

6.8.1 Free

One measurement of performance is the tracking of available free memory. Using the syntax:

free [–option]

```
# free
             total       used       free     shared    buffers     cached
Mem:        515388     502484      12904          0      85592      94908
-/+ buffers/cache:     321984     193404
Swap:      1020116        192    1019924
```

This displays the amount of available memory. Judgment as to whether there is sufficient amount left is to the administrator, but having a bare minimum of 10% with minimal traffic would be a good start, much greater would be much appreciated.

Options include:

- b Show memory in bytes
- k Show memory in Kilobytes
- m Show memory in MegaBytes
- s n Show memory, continuously update every n seconds

6.9 User Groups

As a system administrator, it is sometimes necessary to determine what groups a specific user is a member of.

The syntax of the command is:

groups username

As an example, if we wish to observe who is a member of the group **wheel**, issue the command:

$ groups dennis
dennis : dennis wheel

6.10 System Booting

When a computer boots, it goes through a thee step process. When first powered up, the computer does an initial test of itself. This is called Power On Startup Test (**POST**). After the test has been completed, it directs the computer to go to the boot sector of the first available hard drive (IDE then SCSI). Stored in the **Master Boot Record (MBR)** is another boot program, this is the program that actually reads in the desired operating system. For Linux, two options are now available to boot a computer.

6.10.1 Boot Password Protection

Prior to discussing the details of configuring either LILO or GRUB, it is a good idea to discuss some issues of password protection. Linux is capable of adding on a password that will prevent the system from booting at all – it will not boot into the operating system at all! So why would one want to add this level of protection? Lets list several reasons:

1. If a system uses the LILO boot process, the unit may be booted into the Single User Mode. A Unix / Linux system is wide open to being booted into single user mode, thus bypassing all system security.
2. If a system uses the GRUB boot process, they have access to the GRUB console, and can thereby circumvent the login password.
3. If a system is dual booted with DOS, it has zero protection. This would allow someone to gain access to the unit.

Within the following discussions, reference is made to adding a password to the boot process, prior to the system actually booting the Operating System. The password must be entered prior to the computer completing the boot process. This password is optional, but recommended for a secure system.

6.10.2 Lilo Configuration

Lilo is the **Linux Loader** – or a small program that is responsible for booting Linux, or other Operating Systems from a powerup or reboot. Lilo is slowly being depreciated in favor of GRUB. As of Fedora Core 3, Lilo is no longer provided or supported.

On installation, lilo is normally set up as the default loader with the basic configuration. For this discussion we are assuming that only Linux is being installed on the computer – dual booting is discussed in another topic paper.

Lilo may be installed in either the Master Boot Record (MBR) or on the "superblock" of the root Linux partition. On a normal basis, we want to install it in the MBR, but would install it in the "superblock" if another OS requires its booter to be installed in the MBR, such as OS2.

Several options are available for configuring **lilo**. To add options, we need to edit the **/etc/lilo.conf** file.

The general content of the /etc/lilo.conf file is:

boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
message=/boot/message
lba32
default=linux
password= "desired-password" optional

image=/boot/vmlinuz-2.4.0-0.43.6
label=linux
initrd=/boot/initrd-2.4.0-0.43.6.img
read-only
root=/dev/hda5

6.10.2.1 lilo.conf Options

The **lilo.conf** file may be viewed as two section's, global and booter. The global section specifies basic options for general booting of all OS's, whereas the booter specifies where to look in order to boot the specific OS.

To explain what each line means:

boot	Specifies that the drive to boot off of is **/dev/hda**.
map	Specifies the location of the map file, **/boot/map**. This file should not be modified.
install	Specifies to LILO to install the **/boot/boot.b** file as the new boot sector. This file should not be altered. If the line is missing, LILO will assume a default of /boot/boot.b is to be used.
prompt	Specifies that the message line is to be used for displaying the boot options. It is not recommended that this line be removed.
timeout	Specifies the number of tenths of a second that the prompt will wait for the user to enter which Operating System to use prior to booting. If an entry is not made, the default will be automatically selected.
message	Specifies the screen that will be displayed by LILO for selecting which Operating System is to be used. This screen is displayed only if more than one option is available.

lba32	Specifies the hard drive geometry. The alternative entry is "linear". This value should not be modified.
default	Specifies to which Operating System will be booted if no action is taken by the user. The specified name refers to the ***label*** line within the specific image.
image	Specifies where the Operating System image is located and which file that is to be loaded.
label	Specifies the name that is to be displayed on the LILO selection screen when the system boots. If only one system is installed, then the selection screen is bypassed. This name is used by the default parameter previously specified.
initrd	Specifies the ***initial ram disk*** image that is to be used when the system boots. This image is a minimal operating with ability to load necessary drivers to install the full Operating System Kernel.
read-only	Specifies that the initial loading of the Kernel is to be read only and is not to be altered during the boot process. This is later changed to read-write after the boot process is virtually complete.
root	Specifies where the root partition exists on the designated hard drive and partition.
password	Specifies that a password is required to boot the system prior to logging onto the computer. This is explained below in further detail.

This tells the system that in order to boot Linux, we must look to hda1, the first partition on the first hard drive, and the boot loader is located in the vmlinuz directory. The default name for the Linux installation is linux, but this could be changed, especially if you wish to add another level of security. If you want to perform something other than the default, it must be typed in at the LILO: prompt.

6.10.2.2 Enabling Lilo Modification

After the lilo.conf file has been modified, in order to enable those modifications, it must be reinstalled. This is done by issuing the command:

lilo

This will write the updated configuration to the MBR.

6.10.2.3 Problems Booting at LILO

If during the lilo boot a problem is encountered, an interpretation of the "LILO" characters may be made to potentially understand what the problem is.

The following table may be used to assist in determining the problem.

L(nn)	Disk error, where **nn** is the specific disk error.
LI	Second stage boot loader loaded, but could not run.
LIL	Descriptor table could not be read.
LIL?	Second stage boot loader loaded at incorrect address.
LIL–	LILO found a corrupted descriptor table.

LILO LILO ran successfully.

Before you can make any changes to the lilo.conf file, make sure you have a rescue disk!

6.10.2.4 LILO Boot Options

The following options are available at the **LILO:** prompt. At the **LILO:**, type:

- **Return – or nothing**
  This is the normal boot process. You can hit return to do it immediately, or it will automatically boot after 5 seconds (default).
- **linux rescue**
  Boots into the single-user mode from the backup floppy disk to allow system fixes to the hard drive.
- **linux single**
  Boots from the hard drive as a stand-alone system with no network connection.
- **linux root = <device>**
  Allows user to boot from an external file which is similar to the **/etc/lilo.conf** file.
- **linux vga = <mode>**
  Allows the user to change the resolution of the monitor. Try the "ask" mode.

6.10.3 GRUB Configuration

GRUB, or **GRand Unified Booter**, is now the preferred system booter. The booting process is almost identical to that of Lilo, but an extra step is added. On the Master Boot Record (MBR), the boot program actually points to another program, called GRUB. This then allows one to select which operating system is booted and proceeds to boot it. The major difference is the available options.

The default configuration of the **/boot/grub/grub.conf** file will appear similar to the following:

```
# cat grub.conf
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to
this file
# NOTICE:  You have a /boot partition.  This means that
#         all kernel and initrd paths are relative to /boot/, eg.
#         root (hd0,0)
#         kernel /vmlinuz-version ro root=/dev/hda3
#         initrd /initrd-version.img
#boot=/dev/hda
default=0
timeout=5
password –md5 'encrypted-password'                    optional
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Fedora Core (2.6.12-1.1381_FC3)
```

root (hd0,0)
kernel /vmlinuz-2.6.12-1.1381_FC3 ro root=LABEL=/1 rhgb quiet
initrd /initrd-2.6.12-1.1381_FC3.img

6.10.3.1 GRUB Options

The **/boot/grub/grub.conf** file may also be viewed as two sections, global and booter. The global seciont sepcifies options for gneral booting of OS's, whereas the booter specifies where to look in order to boot the specific OS.

To explain what each line means:

default	Specifies which of the Operating System options is to be booted if no action is taken by the user.
timeout	Specifies the amount of time a user has in seconds before the system automatically boots to the default OS.
splashimage	Specifies the graphical image that is displayed when booting for selecting which OS is to be booted. If only one option is available, this action will be skipped.
hiddenmenu	This line commonly appears by default. The intent is to not display the splashimage option, but rather to just boot the default partition after the timeout option. Clicking any key will display the menu if one wishes to view the menu options. If one wishes to view the OS options (splashimage) screen by default, this line should be commented out.
title	This specifies the OS name that is to be displayed in the splashimage.
root	Specifies the drive and partition where the image is installed. Note that the values start with 0 (zero) rather than 1, as was done for LILO.
kernel	This specifies the initial ram drive image that is to be installed. By default, the image is stored in the /boot/ directory. Additional parameters specified include: ro Installation is to be read only. root Specifies the “root device” and its designator (/). rhgb This specifies the video card loading process. quiet Suppresses normal output.
initrd	Specifies the location of the initrd image file.
password	Specifies an encrypted password that must be entered prior to the system entering into the GRUB2 boot program. The password may be either in clear text or encrypted.
lock	Specifies that a password must be entered when booting a specific operating system.

6.10.3.1 Boot Password

Grub allows the user to set a password in order to boot Linux, just like LILO, with a major enhancement. When the password was set in LILO, it was in plain text, for GRUB the password may be encrypted.

In order to encrypt the desired password, issue the command:

 # grub
GNU GRUB version 0.95 (640K lower / 3072K upper memory)

[Minimal BASH-like line editing is supported. For the first word, TAB
lists possible command completions. Anywhere else TAB lists the possible
completions of a device/filename.]

grub> **md5crypt**

Password: ****** *enter your desired password*
Encrypted: 1DB54D1$sMZfymzeCLeJWbD576XCH/

grub>**quit**

Either copy the encrypted password to the clipboard or copy it down by hand (make sure that you do not make an error). It is best to perform this process when using X Windows. An easy editor to use is **gedit**.

You have now been issued the md5 encryption hash of one's boot password. In the **/boot/grub/grub.conf** file, the line "password" probably does not exist. Therefore it needs to be added. Add the following line immediately following the ***timeout*** line:

password - -md5 ***1DB54D1$sMZfymzeCLeJWbD576XCH/.***

After copy and pasting the encrypted value into the **/boot/grub/grub.conf** file, one must also add the word "**lock**" in appropriate locations, which is immediately after the "**Title**" line for the desired boot process. This is the actual command that demands the entry of a password. If the system is configured as multi-boot, the lock statement must be entered after each title statement. If only one boot is desired to be password protected, then the alternate boot would not include the "lock" statement. The grub.conf file will appear something like the following:

 # cat grub.conf
 # grub.conf generated by anaconda
#
 # Note that you do not have to rerun grub after making changes to this file
 # NOTICE: You have a /boot partition. This means that
 # all kernel and initrd paths are relative to /boot/, eg.
 # root (hd0,0)
 # kernel /vmlinuz-version ro root=/dev/hda3
 # initrd /initrd-version.img
#boot=/dev/hda
default=0
timeout=5
password - -md5 1DB54D1$sMZfymzeCLeJWbD576XCH/.

splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu

title Fedora Core (2.6.12-1.1381_FC3)
lock
root (hd0,0)
kernel /vmlinuz-2.6.12-1.1381_FC3 ro root=LABEL=/1 rhgb quiet
initrd /initrd-2.6.12-1.1381_FC3.img

OK, now you have added the GRUB password, and you have already forgotten it. So how does one get back onto the system? To get back onto the system, circumventing the boot password, one must boot the computer in the rescue mode from the first CD that was used to install the Operating System. From this point, the system drive must be mounted, and the **/boot/grub/grub.conf** file must be edited to either remove the password or to create a new one. At this point the system may now be rebooted.

When booting, the boot screen will be displayed. At this point, the "**P**" key must be clicked, which prompts the user for a password in the lower portion of the screen.

6.11 Operating System Run Level

When Linux boots, during the initial installation, the default setup for booting is to start up in the Command Line Mode (CLI). As an alternative, one may set up the system to automatically boot to the X Windows mode.

The options for Run Level for Red Hat are:

0	Halt	DO NOT set your system to this or it will never operate.
1	Single User Mode	You can achieve this by booting **linux single** at the LILO prompt. Networking not supported..
2	Multiuser	Networking supported without NFS support.
3	Full Multiuser Mode	The fall user mode supporting networking.
4	unused	Reserved for future use.
5	X11	Starts Linux in the X windows mode.
6	Reboot	DO NOT SET system to this or you will never operate.

Note: Other versions of Linux may vary with the assignments of 2 – 5.

You need to edit the file **/etc/inittab.** Find the line:
id: 3 : initdefault:

To convert your system to automatically boot to the X Windows screen, change the above line to:
id: 5 : initdefault:

These are the only two options that we need to be concerned with. Level 1 is normally used during the boot process when the root password has been forgotten. Level 2 would only be necessary if one needed to maximize security by not allowing network access.

6.12 Lost Root Password Recovery

There are three methods to reset the Root Password. To put it in simple terms, there is virtually no security to a Linux machine unless you password protect the boot process. First we will discuss how one can break the administrator's password, then we will discuss how to protect it.

6.12.1 Recovering the Administrator's Password

If the Administrator should loose their password for a given system, then the administrator must circumvent the basic system security to reset their password.

6.12.1.1 Linux Single

Linux Single is the easiest and most direct way to circumvent the logon password.

6.12.1.1.1 LILO

1. Reboot the machine.
2. At the Graphic Red Hat entry screen, as noted in the lower left hand corner, hit **CTRL-X**
3. At the LILO command, type:

   linux single

This puts the user into the single user mode (Run Level 1) with no network connection and does not require a password to boot the system – is that a good situation or not?

The first method after booting into the Linux single mode is to issue the command:

passwd

The system will respond with a request to enter the new password. By booting into the single mode, you are automatically the **root** administrator with the network interface disabled. After you have reestablished your password, reboot the system and do a normal startup. When you are in the single mode, you do not have network connectivity, so a reboot reestablishes your network connectivity.

6.12.1.1.2 GRUB

A system may be booted into the single mode through Grub by using the following process:

1. Reboot the system.
2. Select the desired kernel that is to be booted if multiple boot options are available. Click the **e** key in order to edit the entry, then the **enter** key. A list of options from the configuration file for the selected kernel will be displayed.

3. Select the line that starts with ***kernel***, click the **spacebar** and then the **e** keys, then type in the word **single**. Then click the **Enter** key. You will be returned to the GRUB screen.
4. At the GRUB screen, click the **b** key to boot the computer into the single user mode.
5. After booting, you will be in the single user mode with a basic prompt, a password for logging in will not be required.

The computer will now be booted up into the single user mode. After the system has been fixed, key in **reboot**, the system will then reboot back to the normal system.

6.12.1.2 Editing the Shadow File

The second method accomplishes the same exact function, but is a manual process. After booting into the single mode, you need to modify the **passwd** file using either the **vi** or **nano** text editor.

cd /etc
vi shadow

Find the line that says ROOT
ROOT:xxxxxxxxxxxxxxx: and a lot of other info

Using the insert mode, delete the second field (all the characters represented by the x's). This removes the encrypted password. A new password for the ROOT user may then be entered using the Passwd application.

This is not an optimum process, especially if you have implemented the shadow password file.

6.12.1.3 Last Resort

The third and hard method, which must be used if you are not allowed to boot in the Linux Single mode:

This method requires that you use the Linux Boot Disk and Linux Rescue Disk.

a. Reboot the machine from of the linux boot disk.
b. At the LILO command, type **rescue**
c. Insert Rescue disk.
d. Type:

mount -text2 dev/hdaX /mnt	where X is the Linux drive (not the swap drive)
cd /mnt	
ls	get listing of directories and files
cd /etc	manually mount hard drive because Linux OS is image of that from the floppy
pwd	should get **/etc**
/mnt/bin/vi passwd	(or use nano) change the second field of the ROOT account as described above. (Place cursor

	at start of second field and type DW)
Shift-ZZ	to quit / save / exit
cd /	{Root of floppy imager} must unmount
umount /mnt	previously mounted hard drive in order to fully save file remove floppy and reboot

There is a process where the **linux single** may be set up to prevent it from booting adding more security. If the system is set up in this mode, then the administrator will be required to use the third method to restore the password. Unless you are in an extremely sensitive area requiring a system to be set up to prevent single user access, the configuration should not be implemented.

6.12.2 Adding Security to the Boot Process

During the boot process, we can add one of two levels of security to our system. Each will prevent the computer from either booting at all or booting to a specified operating system. As system may be totally locked down so that no OS may be booted, or a specific OS may be locked from booting.

6.12.3 Recovery In Spite Of

If for some reason you have forgotten the boot password, you are down to two options – one of which you really do not want to do.

6.12.3.1 Total Disaster

Oh well, just reinstall. You got yourself into this mess. Hope you backed up all of your data.

6.12.3.2 Knoppix to the Rescue

This is where Knoppix, or similar operating system can come to the rescue. Just boot up the computer from the CD and make the whatever changes are necessary. Oh, by the way, the system must be CD bootable from BIOS. If the system BIOS is password protected and you forgot the password, then you will have to remove the BIOS battery. There are other Linux OS, including some that boot from a floppy that can also be used.

6.13 User Messages

All of the shells under Linux have the ability to provide the user a message every time they log in to the system, and when exiting. There are several different files that are used. Note that these message files are intended for logging into a system from the CLI, they do not apply to X-Terminal from the GUI.

6.13.1 issue File

The system is able to provide a console message at the CLI login prompt. The default for Red Hat is:

Red Hat Linux release X.Y (RH-Name)	e.g.	7.2 (Enigma)
Kernel 2.X.Y.Z on an architecture		2.4.10 on an i586

This information is maintained in the /etc/issue.net file. The first line is specified in this case, and the second line contains variables. If necessary, the file may be modified.

6.13.2 issue.net File

If a user should log in via a telnet session, they will receive a similar message to that of a console message. In general, it is the same, but it is recommended that it be modified to add a message similar to the following:

Access is monitored – unauthorized access is prohibited.

6.13.3 Message of the Day – motd

The login message is maintained in the /etc/motd file Every time a user logs on, the message is displayed. It is generic, so cannot be username specific (it does not support the $USER variable). The intent is to allow the administrator to send a message to all users when they log in, informing them for example that the system might be shut down at a specified hour.

A message is displayed after a successful login, but prior to the prompt. The default file (/etc/motd) on installation is empty, but can be modified with your preferred editor. It is typically used to announce when a system will be taken down for maintenance or to provide general messages.

6.13.4 Logout Message

When a user logs off, the **$HOME/.bash_logout** file is executed. This is a simple script file that normally contains an echo message and a sleep command. A typical script might look like:

```
clear
echo  $USER, thank you for using the system and have a nice day.
sleep  5
```

You need the sleep command to allow the user to have time to read the message, otherwise it will flash on and off before the user can see it. Don't blink !

6.14 Hard Drive Synchronization

The **sync** command is used to flush or write the data held in RAM buffers to the hard drive. In order to improve system performance, data is normally read from the hard drive and then modified, but is not immediately written to the hard drive. If a power failure should occur, then the modifications might be lost. To force the modifications to be immediately written to the hard drive, the sync command may be used. To synchronize the hard drive to the data in the RAM, issue the command:

```
$ sync
```

In “the old days”, when shutting down Unix, one had to issue the sync command twice prior to issuing the shutdown command. Today, that functionality is built into the Kernel when a proper shutdown is processed.

6.15 Password Check

The first level of security on any system is the password. We must therefore pay particular attention to this.

We have previously learned the two actions by the administrator are required to establish a new user on a host computer. These are:

1. **adduser username**
2. **passwd username**

The first creates the user on the system – but the user is still unable to log on – they are blocked because they have not yet been issued a password. The second establishes the initial password to the user. The administrator must tell the user his/her password, then the user can log onto the system and issue the **passwd** command to change the temporary password to something personal.

If the chosen password is less than 6 characters, a warning message will be issued stating such – but if the user wants, the short password will be accepted. Passwords less than 4 characters will not be accepted.

Passwords on older systems are stored in an encrypted format in the file named ***passwd***. Each line of the file represents a single user. It has the format of:

Username:password:userid:groupid:user full name:user home directory:user shell

Recall that the password is now stored in the **/etc/shadow** file. The contents after the encrypted is different, but we are not worried about that.

The second field is 13 characters long – which is the encrypted format. Recall that newer systems maintain the password in the shadow file.

If you create two new users, both with the same password, you will see that the encrypted password for the two users is different for added security – a hacker can not go back and easily reverse engineer the encryption process. The encryption process used by Unix and Linux uses a ***trapdoor algorithm*** that utilizes a “predictable” key that is different for each user. Hence no two users will have the same encrypted password. This is uniquely different from Microsoft, which utilizes a standard key – hence two different users with the same password will have the same encrypted password. This can be demonstrated by observing the /etc/samba/smbpasswd file (if samba has been activated).

There is a security problem with the passwd file, everyone can read it, although only the root administrator may modify it. Hence anyone may read and then attempt to break the encryption. Developers have therefore extended the security by creating a new file called **shadow**, where the passwords are stored. The shadow password may only be read and modified by the root user or a process that has root privileges. When a challenge is made for the password, the system can read the file for verification purposes – but a user cannot.

When we add the **shadow** password file, replacing the encrypted password field with an x modifies the passwd file.

During the installation of Red Hat 6 and later, use of the shadow password file is the default installation, although it is easily over-ridden.

An older system, which does not have a shadow file, or one, that was intentionally installed without the shadow password file, can easily be upgraded. To upgrade the **passwd** file to the shadow file, issue the command:

pwconv

This will generate a report similar to the following:

This is no longer necessary when using Fedora Core, as the installation of Fedora without the shadow file is not permitted.

6.16 Some Issues with the Password[3,4]

User Passwords should be periodically changed to improve user security. As an administrator you can force users to perform the change. To do this we use the command:

chage

Several options are available, which may be reviewed from the ***man*** pages. One example might be:

chage -M 120 prof

This will require the user *prof* to immediately change their password every 120 days. On the 120th day, on attempting to log in, user *prof* will get the message:

Your password has expired; please change it!
Changing password for prof

The system then prompts the user for their old password and then for the new one. User ***prof*** will not be able to log on until the new password has been entered.

Recall from Chapter 4 that the administrator may modify a password's attributes using the User Manager GUI. This provides an easy method to enter data with a description of each field.

[3] Setting Up a Linux Internet Server – Visual Black Book; Hidenori Tsuji, Takashi Watanabe, Acrobyte; Impress / Coriolis; ISBN: 57610-569-5

[4] Building Linux and OpenBSD Firewalls; Wes Sonnenreich, Tom Yates; Wiley; ISBN 0-471-35366-3

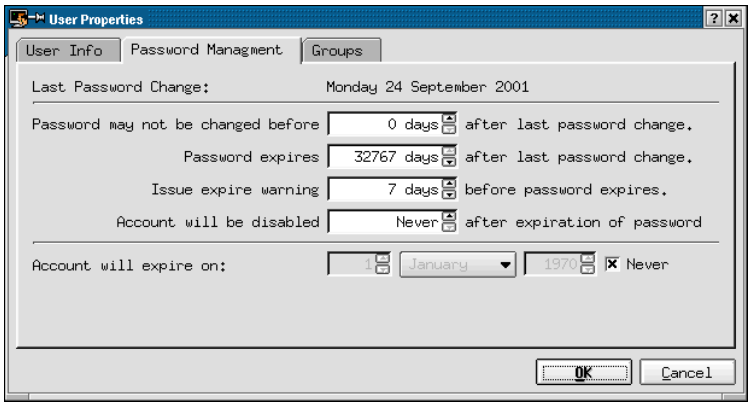

Figure 6.1: Setting User Password Aging

6.16.1 What is a Good Password

When you generate your personal password, you need to make it unique so that others will not be able to figure it out. Here are a few words regarding what is a “bad password”.

1. Any word or words in any language that can be represented in the Roman alphabet.
2. Any word or words varied by changing letters into numbers, or prefixing and / or suffixing numbers.
3. Anything that is all digits.
4. Any variation of the username; if your username is sbn914, then 419nbs is a lousy password.
5. Any variation of your full name or any partition thereof.
6. Anything based on publicly accessible information about you – particularly items such as your birthday.

Avoiding these and you will be doing better than 95 percent of everyone else.

6.16.2 Internet Security

The following are some security issues that one must be aware of when setting up a server on the Internet.

6.16.2.1 Telnet and File Transfer Protocol (FTP)

When logging into these services, the Username and Password are transmitted across the network as clear text. Thus they are able to be monitored and collected by a cracker for later intrusion of the server.

6.16.2.2 Anonymous FTP

Again the Username and Password are transmitted as clear text, but because the username is ***anonymous*** and the password is the users email, it is considered as ok. Secure files should not be placed in an area of the server where an anonymous user will have access. Make sure that the rest of the server is locked down.

6.16.2.3 POP or Internet Message Access Protocol (IMAP)

The Username and Password are transmitted as clear text, as is the transfer of mail messages. Caution must be used when accessing a server from an external Internet location.

6.16.2.4 APOP

The Username is transmitted in clear text, but the Password is encrypted. Transfer of mail messages are transmitted in clear text. Again use caution when accessing a server from an external Internet location.

6.16.2.5 Simple Mail Transfer Protocol (SMTP)

SMTP does not use Usernames or Passwords so there is no concern. Note that one must use caution when setting up a session, as these use POP, IMAP, or APOP. Messages are transmitted in clear text.

6.16.2.6 Hypertext Transfer Protocol (HTTP)

As a normal process, HTTP does not require a Username or Password for access the vast majority of web pages available on the Internet. If a page should require a Username / Password for access, they are transmitted as clear text.

6.16.2.7 Domain Name System (DNS)

All information is transmitted in clear text. Since no Username or Passwords are transmitted, everything is safe. The information transmitted across the Internet is considered public information.

6.16.3 Checking for Unauthorized Access

Unfortunately, a cracker may gain access to your server. There is no guarantee that you can trace the access, specifically if the cracker has logged in as ***root***. On a regular basis one should check out the following files to observe for suspicious activity.

/var/log/boot.log	Log of the starting and stopping of daemons
/var/log/cron	Log of crond (daemon for routinely executing commands)
/var/log/dmesg	Kernel boot messages
/var/log/messages	Log of Berkeley Internet name domain (BIND), the kernel, su, or the like
/var/log/mailog	Log of Sendmail or imapd
/var/log/secure	Login records or log of tcpd
/var/log/xferlog	Log of ftpd
/var/log/wtmp	Log of pass users that have logged into the system
/var/log/btmp	Log of failed login attempts
/var/log/squid/access.log	Log of access to Squid
/var/log/squid/cache.log	Log related to Squid operations
/var/log/squid/store.log	Log relating to Squid cache preservation

/usr/local/www/logs/access_log Access log of Apache
/usr/local/www/logs/error_log Error log of Apache

If there is concern, one can also check user login records.

last
username port day date hours log in duration

This will list past users along with dates and times, thus giving the administrator a good idea of who has been accessing a system. Be careful of what you observe, as a cracker may have logged into the system in a round-about method – telneting from one system to another to another to another. A good cracker will know about the tracking, so they may erase the various log files, thus covering up their tracks.

An alternative to last is the lastb command. This lists the attempts to log in that failed. Initially, the /var/log/btmp file, which is used to log failed logons, does not exist, so the administrator must create the file using the touch command.

More detail to the topic of security will be presented in a later chapter.

6.16.4 Password and Shadow File Checking

There is one method to check the passwd and shadow files. A check is made of the file to verify that the proper number of fields exist. Unfortunately, it does not verify if the password was created from a common word. The command to check a system is:

pwck [passwd | shadow]

6.17 Group Password

Gpasswd is a utility to administer the password for a group. A password for a group is maintained in either the /etc/group or /etc/shadowgrp file. In addition to specifying the group password, a group administrator may be specified. This ability to assign a password and administrator to a group is typically not used very much, but may be handy if one works in a very secure environment. For more information, refer to the man page.

6.18 System Bootdisk

Quite often, a boot disk is not created during the installation process, or for some reason it has been lost. Thus one needs to create a new disk to provide a means to rescue the system if a failure should occur.

The boot-disk is a self-contained Linux system on a floppy disk. In order to perform a rescue, the same basic functions of a complete full-size installation are required. For a general definition, when creating a boot-disk, one is creating both a Kernel and root file system that is capable of starting the computer. Note that if the computer does not have a floppy drive (as many now are no longer equipped with), a boot-disk will not be able to be created or the system rescued.

Log into the system as the administrator. The first thing that is required is to learn which Kernel is running. Issue the command:

 # ls /lib/modules

The response should show a directory that contains the Kernel version, such as 2.6.13-5.

Insert a floppy disk into the drive; recall that the drive is referred to as **fd0**. Next issue the command:

 # mkbootdisk - -device /dev/fd0 X.Y.Z-P

The X, Y, Z, and P values came from the Kernel extension.

After the floppy drive spins down (give it a couple of minutes), you will have an emergency recovery disk.

6.19 stat Utility

The **stat** utility allows one to observe all of the attributes of a file, including:

File Name
Size in Bytes
Number of Blocks occupied
IO Block size
Type of File
Storage Device Value
Inode Value
Hard Links pointing to file
Access Attributes – numeric / rwx
Owner ID – numeric / name
Group ID – numeric / name
Last date accessed
Last date modified
Date Changed

From this list of information, we can determine where a file is stored (Storage Device), its Inode value (where to look on the storage device), the type of file, and the appropriate dates that are relevant to the file. Note that the date of creation is not displayed.

For example, issue the command:

$ stat index.html

File: 'index.html'
Size 1625 Blocks: 8 IO Blocks: 4096
Regular File
Device: 303h/771d Inode: 2932739 Links: 1
Access: (0644/-rw-r- -r- -) Uid: (0/ root) Gid: (0/ root)
Access: 2005-04-23 20:03:56.00000000 -0500
Modify: 2005-03-11 12:10:15.00000000 -0600
Change: 2005-04-23 19:19:37.00000000 -0500

6.20 Hard Drive Statistics – smartctl

SMARTCTL is a command that allows for the Control and Monitor of SMART Disks. The name is an acronym for Self-Monitoring, Analysis and Reporting

Technology. It is a command line utility designed to perform tests on drives that are SMART enabled. Supported devices include latter ATA, IDE, and SCSI-3 hard drives.

The command has the format of:

smartctl [options] device

A few of the options include:

-h Help
-V Version
-i Device information, including vendor, model number, serial number, firmware version, and ATA Standard version information.
-a Displays all of the drive information and statistics. If the drive supports Logical Block Address mode (LBA mode), then the drive's capacity in bytes is also displayed.

For Unix and Linux, the device designator is specified as a device from the **/dev** directory. For the Primary-Master drive the designator is /dev/hda. The most common usage of the command is:

smartctl –a /dev/hda for the primary-master drive

```
$ smartctl -a /dev/hda
smartctl version 5.33 [i386-redhat-linux-gnu] Copyright (C) 2002-4 Bruce Allen
Home page is http://smartmontools.sourceforge.net/

=== START OF INFORMATION SECTION ===
Device Model:     WDC WD400EB-00CPF0
Serial Number:    WD-WCAAT5959658
Firmware Version: 06.04G06
User Capacity:    40,020,664,320 bytes
Device is:        In smartctl database [for details use: -P show]
ATA Version is:   5
ATA Standard is:  Exact ATA specification draft version not indicated
Local Time is:    Thu Feb 23 17:20:03 2006 CST
SMART support is: Available - device has SMART capability.
SMART support is: Enabled

=== START OF READ SMART DATA SECTION ===
SMART overall-health self-assessment test result: PASSED

General SMART Values:
Offline data collection status:  (0x84) Offline data collection activity
                    was suspended by an interrupting command from host.
                    Auto Offline Data Collection: Enabled.
Self-test execution status:      (   0) The previous self-test routine completed
                    without error or no self-test has ever
                    been run.
Total time to complete Offline
data collection:    (1754) seconds.
Offline data collection
capabilities:       (0x3b) SMART execute Offline immediate.
                    Auto Offline data collection on/off support.
                    Suspend Offline collection upon new
                    command.
                    Offline surface scan supported.
                    Self-test supported.
                    Conveyance Self-test supported.
                    No Selective Self-test supported.
SMART capabilities:            (0x0003) Saves SMART data before entering
                    power-saving mode.
                    Supports SMART auto save timer.
Error logging capability:        (0x01) Error logging supported.
                    No General Purpose Logging support.
```

```
Short self-test routine
recommended polling time:         (   2) minutes.
Extended self-test routine
recommended polling time:         (  30) minutes.
Conveyance self-test routine
recommended polling time:         (   5) minutes.

SMART Attributes Data Structure revision number: 16
Vendor Specific SMART Attributes with Thresholds:
ID# ATTRIBUTE_NAME          FLAG  VALUE WORST THRESH TYPE      UPDATED
                                                              WHEN_FAILED RAW_VALUE
1 Raw_Read_Error_Rate       0x000b 200   200   051    Pre-fail  Always   -      0
3 Spin_Up_Time              0x0007 104   088   021    Pre-fail  Always   -   2166
4 Start_Stop_Count          0x0032 100   100   040    Old_age   Always   -    249
5 Reallocated_Sector_Ct     0x0033 200   200   140    Pre-fail  Always   -      0
7 Seek_Error_Rate           0x000b 100   253   051    Pre-fail  Always   -      0
9 Power_On_Hours            0x0032 088   088   000    Old_age   Always   -   9393
10 Spin_Retry_Count         0x0013 100   100   051    Pre-fail  Always   -      0
11 Calibration_Retry_Count  0x0013 098   098   051    Pre-fail  Always   -      6
12 Power_Cycle_Count        0x0032 100   100   000    Old_age   Always   -    242
196 Reallocated_Event_Count 0x0032 200   200   000    Old_age   Always   -      0
197 Current_Pending_Sector  0x0012 200   200   000    Old_age   Always   -      0
198 Offline_Uncorrectable   0x0012 200   200   000    Old_age   Always   -      0
199 UDMA_CRC_Error_Count    0x000a 200   253   000    Old_age   Always   -      0
200 Multi_Zone_Error_Rate   0x0009 200   200   051    Pre-fail  Offline  -      0

SMART Error Log Version: 1
No Errors Logged

SMART Self-test log structure revision number 1
No self-tests have been logged.  [To run self-tests, use: smartctl -t]

Device does not support Selective Self Tests/Logging
```

In particular, look at line 9, "**Power_On_Hours**", this line shows how many hours the drive has been powered on. In the example above, the drive has 88 hours of usage. One might wish to monitor this value, and make sure that the drive is backed up when the hours get really high. Drives with nearly 2000 operation hours have been observed – I would be concerned with such a drive.

6.21 Shell Configuration Scripts

Various scripts are read during the login process in order to set the user environment. Which file is read is dependent upon the user's shell that is enacted. The files in the /etc directory are global, in that any configuration in them apply to all users. The files in a users home directory (~/) are applicable to only that specific user.

6.21.1 Bash

On login, the following scripts are read:

/etc/bashrc
~/.bashrc
/etc/bash_profile
~/.bash_profile
~/.bash_login
~/.profile
~/.bash_logout (when user logs off)

6.21.2 C Shell

On login, the following scripts are read:

/etc/csh.cshrc
/etc/csh.login
~/.cshrc
~/.history
~/.login
~/.cshdirs

6.21.3 TC Shell

On login, the following scripts are read:

/etc/csh.cshrc
/etc/csh.login
~/.tcshrc (~/cshrc if .tcshrc does not exist)
~/.history
~/.login
~/.cshdirs

6.21.4 Z Shell

On login, the following scripts are read:

/etc/zshenv
~/.zshenv
/etc/zprofile
~/.zprofile
/etc/zshrc
~/.zshrc
/etc/login

6.22 Internet Information Location

The following information is located in the respective files:

IP Address
    /etc/sysconfig/network-scripts/ifcfg-{interface}
Hostname
    /etc/sysconfig/network
    /etc/hosts
Default Gateway
    /etc/default-route
DNS Server
    /etc/resolv.conf
IP-Host Mapping
    /etc/hosts
Hostname Lookup Order
    /etc/nsswitch.conf
DNS Resolver
    /etc/hosts.conf

6.23 Listing Locked Files

Different system processes and application typically look up data in various files. If they need to write to the file, then the file needs to be locked from other users, allowing only one user to the file at a time. Some applications may totally lock a file preventing others from even reading it, while others will allow a user to only read. If allowed to read, then the second application may read inaccurate information if a change is made before the first application has made its changes. The user can check to see if a file is locked by issuing the command:

$ lsof filename

6.24 Builtin Commands

Linux includes a number of builtin commands – that is, commands that are included within the Kernel. These allow the basic operation of the operating system without external applications or utilities. In fact, many external commands make use of the internal commands by enhancing their basic performance. To understand the full benefit of commands not previously covered, refer to the man bash page (SHELL BUILTIN COMMANDS).

Commands include:

alias, bg, bind, break, builtin, case, cd, command, continue, declare, dirs, disown, echo, enable, eval, exec, exit, export, fc, fg, for, getopts, hash,
help, history, if, jobs, kill, let, local, logout, popd, pushd, pwd, read, readonly, return, set, shift, shopt, source, suspend, test, times, trap, type, typeset, ulimit, umask, unalias, unset, until, wait, while

6.25 Commands Used in this Chapter

alias	Creates a new command that invokes another standard command
at	Sets up a one time delayed action process
atq	Lists at jobs that are in queue
bzip2	A compression / un-compression utility
chage	A utility to modify the aging of a password
chattr	Modifies the “back-end” attributes of a file
chfn	Modifies the finger information of a user
chmod	Modifies the permissions of a file
chown	Modifies the owner of a file
chgrp	Modifies the group of a file
compress	A compression utility
cron	Sets up periodic processes
crontab	Utility to set up cron processes
df	Lists filesystem structure (tree structure)
du	Lists filesystem usage (file size)
env	Lists a user’s environment (also ‘set’)
export	A utility to add a variable to a user’s environment
fc	Displays the last 20 commands that have been used on the system

fdformat	A utility for formatting a floppy disk
file	Lists the file type
finger	Displays the comment field of a user in the /etc/passwd file
free	Displays the amount of free memory
gpasswd	A utility to administer the group password
groups	Lists the groups that a specified username belongs to
gunzip	A utility to un-compress a file that was compressed using 'gzip'
gzip	A compression utility
history	Displays the last 1000 commands that have been used on a system
kill	Terminates an active program
last	Lists users who have been logged on, displaying last time and duration of logon
lilo	A utility to write the modifications to the lilo.conf file to the Master Boot Record
lock	Command under GRUB to require a boot password
lsattr	Lists the "back-end" attributes of a file
lslk	Lists locked files
lsof	Lists files that have been locked out from operation
mc	Midnight Commander utility similar to Norton Commander DOS utility
mcopy	Utility to copy information to or from a DOS floppy disk
md5crypt	A command under GRUB to encrypt a password
mdir	Utility to list the contents of a DOS floppy disk
mkbootdisk	Utility to create a rescue floppy disk.
'mtools'	A set of stand-alone utilities for working with a DOS floppy disk
mtype	Utility to display the contents of a file that resides on a floppy disk
password	A command under GRUB to specify the password
ps	Lists processes that are active
pstree	Lists process, displaying in a parent-child tree format
pwck	Verifies that all fields of the /etc/passwd field exist
set	Lists a user's environment (also 'env')
sleep	Pauses all processes for a specified number of seconds
smartctl	Displays the statistics for the specified hard drive
stat	Lists all attributes of a file in a verbose mode
sync	A utility to flush the RAM information to the hard drive
tar	A utility to combine multiple files together, does not compress
top	Lists processes that are active and updated memory usage
umask	Lists the file permission mask
unalias	Deletes an existing alias command
uname	Lists information about a system's hardware and network configuration
uncompress	A utility to un-compress a file that was compressed using 'compress'

unzip	A utility to un-compress a file that was compressed using 'zip'
uptime	A utility to list the time a system has been operational
userid	Lists the ID of a user
usermod	Modifies attributes of a user
vmstat	Lists virtual memory statistics
w	Lists users who are presently logged onto a system with more detail than that provided by 'who'
who	Lists users who are presently logged onto a system
whoami	Lists information about the system and the user presently logged on, also may use format 'who am i'
zip	A utility to compress a file

6.26 Chapter Review Questions

1. What utility may be used to implement and configure password aging?
   a. pwage
   b. chage
   c. pwmod
   d. passwd
2. You have an image file on a floppy disk that you desire to copy to your /lab directory. What command is issued?
   a. cp a:file /lab/file
   b. cp a:file /lab/newfile
   c. copy a:\file /lab
   d. mcopy a:file /lab
3. You may created a logout message in which path/file?
   a. ~.bash_logout
   b. /home/.bash_logout
   c. /etc/.bash_logout
   d. /etc/bash_logout
4. You need to change the file "start" such that the users have the access of:

	R	W	X
a.			
b.	owner x	x	–
c.	group x	–	–
d.	world –	–	–

5. What command is issued?
   a. chmod 137 start
   b. chmod 210 start
   c. chmod 640 start
   d. filemod 640 start

6. You need to learn specific information about a system. What command is used?
   a. sysinfo
   b. uname
   c. proc
   d. list
7. What is the utility to display active processes?
   a. process
   b. proc
   c. ps
   d. tree
8. You created a file “start” while logged on as the administrator, but you need the owner to be jdoe. What command is issued?
   a. chown jdoe start
   b. chown start jdoe
   c. filemod jdoe start
   d. filemod start jdoe
9. A process has locked up and will not terminate. What command will terminate the process?
   a. terminate pid
   b. term pid
   c. kill pid
   d. bye pid
10. You need to keep an application resident in memory after it terminates. What must be done?
    a. Set UserID
    b. Set GroupID
    c. Set Sticky Bit
    d. Set Memory Stay Resident
11. You need to have a command issued every work day at 8 AM. What utility would be used?
    a. at
    b. cron
    c. run
    d. time
12. To read the directory’s contents of a floppy disk, what command is used?
    a. dir
    b. ls
    c. mdir
    d. mls
13. A message of the day is maintained in which path / file?
    a. var/motd
    b. ~.motd
    c. /etc/.motd
    d. /etc/motd

14. You list a file's attributes and observe that the first character is a "d". What type of file is it?
    a. ASCII file
    b. binary file
    c. directory
    d. link
15. You need to modify the group name for the file "start" to "students". What command is used?
    a. chgrp students start
    b. chgrp start students
    c. usermod students start
    d. usermod start students
16. It is necessary to combine multiple files together for backup purposes. What command is issued?
    a. zip
    b. tar
    c. group
    d. save
17. Virtual memory statistics are displayed using what command?
    a. memstat
    b. stat
    c. vmstat
    d. statvm
18. You have lost the administrator's password on a system that boots using LILO. What must be done to boot the system?
    a. Use rescue disk
    b. Re-install
    c. Boot to normal user
    d. Boot to single mode
19. You have developed a one-line set of commands to perform a function that is to be utilized again in the future. It is quite difficult to remember, so you decide to rename it. What command is used?
    a. mv
    b. rename
    c. mark
    d. alias
20. You need to determine how much space is remaining on your hard drive. Which command is issued?
    a. du
    b. df
    c. space
    d. mem
21. Displaying a user's environment uses what command?
    a. show
    b. ev
    c. env
    d. disp

22. To display child processes in a tree structure, what command is issued?
    a. pstree
    b. tree
    c. root
    d. leaf
23. You need to allow other users to inherit the rights of the owner when they use the file “start”. What command is issued?
    a. modgrp 4000 start
    b. modgrp u+g start
    c. chmod u+g start
    d. chmod 2000 start
24. You need to know what groups a user is a member of. What command is issued?
    a. member username
    b. group username
    c. groups username
    d. source username
25. On a single occasion, you need to have a command issued at a later hour. What utility would you use?
    a. cron
    b. at
    c. next
    d. time
26. What is the path and filename that specifies the system run level?
    a. /etc/run
    b. /etc/inittab
    c. /proc/inittab
    d. /var/inittab
27. You list the attributes of a file and observe the following:
    lrwSr - S- -
    What does this mean?
    a. Owner can scan file
    b. Group can scan file
    c. Link is scanable
    d. SUID and SGID are set
28. What run level is a system operating at to operate as a multiuser without NFS support?
    a. 1
    b. 2
    c. 3
    d. 5
29. You need to review previously issued commands on a system. What command is issued?
    a. list
    b. history
    c. cat
    d. command

30. You wish to make sure the file “start” is not modified or deleted. What command is issued?
    a. lsattr start
    b. view start
    c. set -i start
    d. chattr -i start
31. You wish to add a new variable to your environment. What command is issued?
    a. export variable
    b. import variable
    c. env variable
    d. set variable

Chapter Index