ls Displays the contents of a directory.

1. To list all files in the current directory, type: ls -a

This lists all files, including . (dot), .. (dot-dot), and other files with names beginning with a dot.

2 To display detailed information, type: ls -l chap1 .profile

This displays a long listing with detailed information about chap1 and .profile.

3 To display detailed information about a directory,

```
$ ls -d -l . manual manual/chap1
```

This displays a long listing for the directories . and manual, and for the file manual/chap1.
Without the -d flag, this would list the files in the . and manual directories instead of the detailed information about the directories themselves.

4 To list the files in order of modification time,

```
$ ls -l -t
```

This displays a long listing of the files that were modified most recently, followed by the older files.

5 To display detailed information with expanded user and group name,

```
$ls  -lX .profile
```

This displays a long listing with detailed information about .profile.

6 To display about whether extended attributes are set on the files in current directory, type:

```
ls -U
```

Example output:

```
-rwSr-x---+   1 root     system           28 Apr 29 03:23 only_aixc
-rwSr-x---E   1 root     system            4 Apr 29 03:23 only_aixc_ea
-rw-r--r--E   1 root     system            4 Apr 29 03:23 only_ea
----------+   1 root     system          265 Apr 29 03:23 only_nfs4
----------E   1 root     system           64 Apr 29 03:23 only_nfs4_ea
-rw-r--r---   1 root     system            4 Apr 29 03:23 only_regular
```

---

pwd purpose Displays the path name of the working directory.

```
$ pwd
```

displays the current directory as:

```
/home/thomas
```

---

id Displays the system identifications of a specified user.

1 To display all system identifications for the current user, enter:

```
$ id
```

Output for the id command is displayed in the following format:

```
uid=1544(sah) gid=300(build) euid=0(root) egid=9(printq) groups=0(system),10(audit)
```

In this example, the user has user name sah with an ID number of 1544; a primary group name of build with an ID number of 300; an effective user name of root with an ID number of 0; an effective group name of printq with an ID number of 9; and two supplementary group names of system and audit, with ID numbers 0 and 10, respectively.

2 To display all group ID numbers for the current user, enter:

```
$ id -G
```

Output is displayed in the following format:

```
0 10 300 9
```

The -G flag writes only the group IDs for a user. In this example, user sah is a member of the system (0), audit (10), build (300), and printq (9) groups.

3 To display all group names for the current user, enter:

```
$ id -Gn
```

Output is displayed in the following format:

```
system audit build printq
```

The -n flag writes only the names instead of the ID numbers.

4 To display the real group name for the current user, enter:

```
$ id -gnr
```

Output is displayed in the following format:

```
Build
```

5 To display the login UID after logging in as root and running the su command to user sah, type:

```
$ id –lu
```

6 To display the primary group name of the user who actually logged in, type:

```
$ id -lgn
```

7 To display the primary group ID of the user who actually logged in, type:

```
id -lg
```

---

cal Displays a calendar.

1 To display a calendar for February, 1994, at your workstation, enter:

```
$  cal 2 1994
```

2 To print a calendar for 1994, enter:

```
$ cal 1994
```

3 To display a calendar for the year 84, enter:

```
$ cal 84
```

date   Displays or sets the date or time.

1   To display current date and time, enter:

```
$ date
```

2   To set the date and time, enter:

```
$ date 0217142590
```

3   To display the date and time in a specified format, enter:

```
$ date +"%r %a %d %h %y (Julian Date: %j)"
```

This displays the date shown in Example 2 as:

```
02:25:03 PM Fri 17 Feb 90 (Julian Date: 048)
```

---

cd   Changes the current directory.

1   To change the current working directory to the login (home) directory, type:

```
$ cd
```

2   To change to an arbitrary directory, type:

```
$ cd /usr/include
```

This changes the current directory to /usr/include.

3   To go down one level of the directory tree, type:

```
$ cd sys
```

If the current directory is /usr/include and it contains a subdirectory named sys, then /usr/include/sys becomes the current directory.

4   To go up one level of the directory tree, type:

```
$ cd ..
```

---

touch   Updates the access and modification times of a file.

1   To update the access and modification times of a file, enter:

```
$ touch program.c
```

This sets the last access and modification times of the program.c file to the current date and time. If the program.c file does not exist, the touch command creates an empty file with that name.

2   To avoid creating a new file, enter: touch -c program.c

3   To update only the modification time, enter: touch -m *.o

This updates the last modification times (not the access times) of the files that end with a .o extension in the current directory. The touch command is often used in this way to alter the results of the make command.

4   To explicitly set the access and modification times, enter:

```
$ touch -c -t 02171425 program.c
```

This sets the access and modification dates to 14:25 (2:25 p.m.) February 17 of the current year.

5   To use the time stamp of another file instead of the current time, enter:

```
$ touch -r file1 program.c
```

This gives the program.c file the same time stamp as the file1 file.

6 To touch a file using a specified time other than the current time, enter:

```
$ touch -t 198503030303.55   program.c
```

This gives the program.c file a time stamp of 3:03:55 a.m. on March 3, 1985.

---

mkdir Creates one or more new directories.

1 To create a new directory called Test in the current working directory, enter:

```
$ mkdir Test
```

The Test directory is created with default permissions.

2 To create a new directory called Test with rwxr-xr-x permissions in the previously created /home/demo/sub1 directory, enter:

```
$ mkdir -m 755 /home/demo/sub1/Test
```

3 To create a new directory called Test with default permissions in the /home/demo/sub2 directory,

```
$ mkdir -p /home/demo/sub2/Test
```

The -p flag creates the /home, /home/demo, and /home/demo/sub2 directories if they do not already exist.

----

rmdir Removes a directory.

1 To empty and remove a directory, type:

```
$ rm mydir/* mydir/.*
$ rmdir mydir
```

This command removes the contents of the mydir file and then removes the empty directory. The rm command displays an error message about trying to remove the directories . (dot) and .. (dot, dot), and then the rmdir command removes them.

Note that the rm mydir/* mydir/.* command first removes files with names that do not begin with a dot, and then removes those with names that do begin with a dot. You may not realize that the directory contains file names that begin with a dot because the ls command does not usually list them unless you use the -a flag.

2 To remove the /home, /home/demo, and /home/demo/mydir directories, type:

```
$ rmdir -p /home/demo/mydir
```

This command removes first the /mydir directory and then the /demo and /home directories, respectively. If a directory is not empty or does not have write permission when it is to be removed, the command terminates.

---

rm Removes (unlinks) files or directories.

1 To delete a file, enter:

```
$ rm myfile
```

If there is another link to this file, then the file remains under that name, but the name myfile is removed. If myfile is the only link, the file itself is deleted.

2 To delete a file without first receiving a confirmation prompt, enter:

```
      $ rm -f core
```

No confirmation prompt is issued before the rm -f command attempts to remove the file named core. However, an error message displays if the core file is write-protected and you are not the owner of the file or you do not have root authority. No error message displays when the rm -f command attempts to remove nonexistent files.

3 To delete files one by one, enter:

```
     $ rm -i mydir/*
```

After each file name is displayed, enter y to delete the file, or press the Enter key to keep it.

4 To delete a directory tree, enter:

```
      $ rm -ir manual
```

This command recursively removes the contents of all subdirectories of the manual directory, prompting you regarding the removal of each file, and then removes the manual directory itself, for

---

mv Moves files.

1 To rename a file, enter:

```
            $ mv appendix apndx.a
```

This command renames appendix to apndx.a. If a file named apndx.a already exists, its old contents are replaced with those of appendix.

2 To move a directory, enter:

```
           $ mv book manual
```

This command moves all files and directories under book to the directory named manual, if manual exists. Otherwise, the directory book is renamed manual.

3 To move a file to another directory and give it a new name, enter:

```
          $ mv intro manual/chap1
```

This command moves intro to manual/chap1. The name intro is removed from the current directory, and the same file appears as chap1 in the directory manual.

4 To move a file to another directory, keeping the same name, enter:

```
           $ mv chap3 manual
```

This command moves chap3 to manual/chap3

Note: Examples 1 and 3 name two files, example 2 names two existing directories, and example 4 names a file and a directory.

5 To move several files into another directory, enter:

```
          $  mv chap4 jim/chap5 /home/manual
```

This command moves the chap4 file to the /home/manual/chap4 file directory and the jim/chap5 file to the /home/manual/chap5 file.

6 To use the mv command with pattern-matching characters, enter:

```
        $ mv manual/* .
```

This command moves all files in the manual directory into the current directory . (period),retaining the names they had in manual. This move also empties manual. You must type a space between the asterisk and the period.

Note: Pattern-matching characters expand names of existing files only. For example, the command

```
    $mv intro man*/chap1
```

does not work if the file manual/chap1 does not exist.

---

cat Concatenates or displays files.

1   To display a file at the workstation, enter:

```
         $ cat notes
```

This command displays the data in the notes file. If the file is more than one less than the number of available display lines, some of the file scrolls off the screen. To list a file one page at a time, use the pg command.

2   To concatenate several files, enter:

```
       $ cat section1.1 section1.2 section1.3 >section1
```

This command creates a file named section1 that is a copy of section1.1 followed by section1.2 and section1.3.

3   To suppress error messages about files that do not exist, enter:

```
     $ cat -q section2.1 section2.2  section2.3 >section2
```

If section2.1 does not exist, this command concatenates section2.2 and section2.3. The result is the same if you do not use the -q flag, except that the cat command displays the error message:

```
        $ cat: cannot open section2.1
```

You may want to suppress this message with the -q flag when you use the cat command in shell procedures.

4   To append one file to the end of another, enter:

```
          $ cat section1.4 >> section1
```

The >> (two carets) appends a copy of section1.4 to the end of section1. If you want to replace the file, use the > (caret).

5   To add text to the end of a file, enter:

```
        $ cat >>notes
          Get milk on the way home
          Ctrl-D
```

This command adds Get milk on the way home to the end of the file called notes. The cat command does not prompt; it waits for you to enter text. Press the Ctrl-D key sequence to indicate you are finished.

6   To concatenate several files with text entered from the keyboard, enter:

```
        $cat section3.1 - section3.3 >section3
```

This command concatenates the file section3.1 with text from the keyboard (indicated by the minus sign), and the file section3.3, then directs the output into the file called section3.

---

more   Displays file contents one screen at a time.

1   To view a file named myfile, enter:

```
$ more myfile
```

2   To view output from the nroff command, enter:

```
$ls -l | more
```

3   To view each file starting at its last screen, enter:

```
$more -p G file1 file2
```

4   To view each file with the 100th line at the current position, enter:

```
$more -p 100 file1 file2
```

---

pg   Formats files to the display.

To look at the contents of a file one page at a time, enter:

```
$ pg filename
```

---

head   Displays the first few lines of a file.

1. To display the first five lines of the Test file, enter:

```
$ head -5 Test
```

OR

```
$  head -n 5 Test
```

---

tail   Displays the last few lines of a file.

1   To display the last 10 lines of the notes file, enter:

```
$tail notes
```

2   To specify the number of lines to start reading from the end of the notes file,

```
$tail -n 20 notes
```

3   To display the notes file a page at a time, beginning with the 200th byte, enter:

```
$ tail -c +200 notes | pg
```

4   To follow the growth of a file, enter:

```
$tail -f accounts
```

This displays the last 10 lines of the accounts file. The tail command continues to display lines as they are added to the accounts file. The display continues until you press the Ctrl-C key sequence to stop it.

uname Displays the name of the current operating system.

1. To display the complete system name and version banner, enter:

```
$ uname –a
```

uptime Shows how long the system has been up.

```
$ uptime
```

wc Counts the number of lines, words, bytes, or characters in a file.

1. To display the line, word, and byte counts of a file, enter:

```
$wc chap1
```

The wc command displays the number of lines, words, and bytes in the chap1 file.

2 To display only byte and word counts, enter:

```
$ wc -cw chap*
```

The wc command displays the number of bytes and words in each file that begins with chap. The command also displays the total number of bytes and words in these files.

3 To display the line, word, and character counts of a file, enter:

```
$ wc -k chap1
```

The wc command dispays the number of lines, words, and characters in the chap1 file.

4 To display the word and character counts of a file, enter:

```
$wc -kcw chap1
```

The wc command displays the number of characters and words in the chap1 file.

5 To use the wc command on standard input, enter:

```
$ wc -klw
```

The wc command displays the number of lines and words in standard input. The -k flag is ignored.

6 To display the character counts of a file, enter:

```
$ wc -m chap1
```

The wc command displays the number of characters in the chap1 file.

7 To use the wc command on standard input, enter:

```
$wc -mlw
```

The wc command displays the number of lines, words, and characters in standard input.

diff Compares text files.

1 To compare two files, enter:

```
diff chap1.back chap1
```

This displays the differences between the files chap1.bak and chap1.

2 To compare two files while ignoring differences in the amount of white space, enter:

```
$ diff –w   prog.c.bak prog.c
```

If two lines differ only in the number of spaces and tabs between words, the diff -w command considers them to be the same.

3 To create a file containing commands that the ed command can use to reconstruct one file from another, enter:

```
$ diff -e chap2 chap2.old >new.to.old.ed
```

This creates a file named new.to.old.ed that contains the ed subcommands to change chap2 back into the version of the text found in chap2.old. In most cases, new.to.old.ed is a much smaller file than chap2.old. You can save disk space by deleting chap2.old, and you can reconstruct it at any time by entering: (cat new.to.old.ed ; echo '1,$p') | ed - chap2 >chap2.old

The commands in parentheses add 1,$p to the end of the editing commands sent to the ed editor. The 1,$p causes the ed command to write the file to standard output after editing it. This modified command sequence is then piped to the ed command (| ed), and the editor reads it as standard input. The - flag causes the ed command not to display the file size and other extra information because it would be mixed with the text of chap2.old.

last Displays information about previous logins.

Examples

1 To display all the recorded logins and logoffs by user root or from the console terminal, type:

```
last root console
```

2 To display the time between reboots of the system, type:

```
last reboot
```

The reboot pseudo-user logs in when the system starts again.

3 To display all the users still logged in at 10.30 am on April 15th, enter:

```
last -t 04151030
```

4 To display 10 lines in the list, type:

```
last -n 10
```

5 To display all the recorded logins and logoffs without truncating the user name, type:

```
last -X
```

uniq Reports or deletes repeated lines in a file.

To delete repeated lines in a file named fruit and save it to a file named newfruit, enter:

```
$ uniq fruit newfruit
```

If the fruit file contains the following lines:

```
apples
apples
peaches
pears
bananas
cherries
cherries
```

then the newfruit file will contain the following lines after you run the uniq command:

```
apples
peaches
pears
bananas
cherries
```

---

who   Identifies the users currently logged in.

1. To display information about who is using the local system node, type:

```
who
```

Information similar to the following is displayed:

```
pts/1      Nov  9 00:20   long_username_greater_than_eight_characters  (localhost)
```

2   To display your user name, type:

```
who am i
```

Information similar to the following is displayed:

```
george lft/0 Jun 8 08:34
```

3   To display a history of logins, logouts, system startups, and system shutdowns, type:

```
who /var/adm/wtmp
```

Information similar to the following is displayed:

```
hank   lft/0   Jun   8   08:34  (ausnix5)
john   lft/0   Jun   8   08:34  (JIKey)
mary   lft/0   Jun   8   08:22  (machine.austin.ibm)
jan    pts4    Jun   8   09:19  (puff.wisc.edu)
```

4   To display the run-level of the local system node, type: who -r

Information similar to the following is displayed:

```
. run-level 2 Jun 8 04:15 2 0 s
```

5   To display any active process that is currently actively and has been previously generated by init, type:

```
who -p
```

Information similar to the following is displayed:

```
srcmstr   .    Jun 8  04:15  old  2896
cron      .   Jun 8  04:15  old  4809
uprintfd  .    Jun 8  04:15  old  5158
```

6 To process the /var/adm/wtmp file with the -bdlprtTu flags specified, type: who -a /var/adm/wtmp

Information similar to the following is displayed:

```
   .       system boot Jun 19 10:13
   .       run-level 2 Jun 19 10:13
   .          .      Jun 19 10:14    old
   .          .      Jun 19 10:14    old
   .          .      Jun 19 10:14    old
rc        -    .      Jun 19 10:13    old
   .          .      Jun 19 10:16    old
   .          .      Jun 19 10:14    old
srcmstr   -     .      Jun 19 10:14    old
rctcpip   -     .      Jun 19 10:14    old
rcdce     -     .      Jun 19 10:14    old
rccm      -     .      Jun 19 10:15    old
dceupdt   -     .      Jun 19 10:15    old
rcnfs     -     .      Jun 19 10:15    old
cron      -     .      Jun 19 10:16    old
piobe     -     .      Jun 19 10:16    old
qdaemon   -     .      Jun 19 10:16    old
writesrv  -     .      Jun 19 10:16    old
uprintfd  -     .      Jun 19 10:16    old
   .          .      Jun 19 10:16    old
LOGIN     - lft0       Jun 19 10:16    old

   .          .      Jun 19 10:16    old
   .          .      Jun 19 10:16    old
```

whoami Displays your login name.

```
whoami
```

df Reports information about space on file systems. This document describes the AIX df command as well as the System V version of df.

1 To display information about all mounted file systems, enter:

```
df
```

If your system has the /, /usr, /site, and /usr/venus file systems mounted, the output from the df command resembles the following:

```
Filesystem 512-blocks Free   %Used   Iused  %Iused  Mounted on
/dev/hd0    19368     9976    48%     4714    5%     /
/dev/hd1    24212     4808    80%     5031   19%     /usr
/dev/hd2     9744     9352     4%     1900    4%     /site
/dev/hd3     3868     3856     0%      986    0%     /usr/venus
```

2 To display information about /test file system in 1024-byte blocks, enter:

```
df -k /test
```

```
Filesystem    1024 blocks    Free    %Used   Iused  %Iused  Mounted on
/dev/lv11         16384     15824       4%      18      1%  /tmp/ravi1
```

This displays the file system statistics in 1024-byte disk blocks.

3 To display information about /test file system in MB blocks, enter:

```
df -m /test
```

```
Filesystem    MB blocks    Free    %Used    Iused  %Iused  Mounted on
/dev/lv11       16.00     15.46       4%       18      1%  /tmp/ravi1
```

This displays file system statistics in MB disk blocks rounded off to nearest 2nd decimal digit.

4 To display information about the /test file system in GB blocks, enter:

```
df -g /test
```

```
Filesystem    GB blocks   Free     %Used    Iused  %Iused  Mounted on
/dev/lv11          0.02   0.02        0%       18      1%  /tmp/ravi1
```

This displays file system statistics in GB disk blocks rounded off to nearest 2nd decimal digit.

5 To display available space on the file system in which your current directory resides, enter:

```
cd/
df .
```

The output from this command resembles the following:

```
Device   512-blocks  free   %used   iused   %iused  Mounted on
/dev/hd4    19368    9976    48%    4714    5%     /
```

System V df Command

Reports number of free disk blocks and files.

Syntax

```
/usr/sysv/bin/df [ -a ] [ -l ] [ [ [ -e ] [-g ] [ -n ] ] | [ [ -i ] [ -v ] ] | -t ] ] [FileSystem ...]
```

The df command displays information about total space and available space on a file system. File system statistics are displayed in units of 512-byte blocks

Flags

-a
Performs the default operation and prints the mount point, the device name, number of free blocks and number of used inodes (files).

-e
Print only the number of free files.

-g
Print the entire statvfs structure. This option overrides the -a , -e, -i, -n, -t and -v options.
The numbers for available, total, and free blocks are reported in 512 byte blocks.

-i
Displays the total number of inodes, the number of free inodes, the number of used inodes, and the percentage of inodes in use.

-l
Reports on local file systems only.

-n
Prints the type of filesystem.

-t

Causes total allocated block figures to be reported.

-v

Reports percent of blocks used as well as the number of blocks used and free.

1 To display information about all mounted file systems, enter:

```
/usr/sysv/bin/df
```

The output looks similar to the following:

```
/          (/dev/hd4      ):    19656 blocks    1504 files
/usr        (/dev/hd2      ):  1139904 blocks   20254 files
/var        (/dev/hd9var   ):    23096 blocks     512 files
/tmp         (/dev/hd3      ):     2464 blocks     204 files
/home         (/dev/hd1      ):    44208 blocks     146 files
/proc        (/proc         ):        0 blocks       0 files
/opt        (/dev/hd10opt  ):    13880 blocks     310 files
```

2 To display information about the file system in which your current directory resides, enter:

```
/usr/sysv/bin/df .
```

3 To display the total number of inode, the number of free inodes and the number of available inodes in all mounted file systems, enter:

```
/usr/sysv/bin/df -i
```

The output looks similar to the following:

```
Mount Dir  Filesystem      iused    avail    itotal  %iused
/         /dev/hd4        1504     6688      8192     19%
/usr       /dev/hd2        20254   127202    147456     14%
/var       /dev/hd9var       512     3584      4096     13%
/tmp       /dev/hd3          204     5940      6144      4%
/home       /dev/hd1          146    14190     14336      2%
/proc      /proc              0        0         0       0
/opt       /dev/hd10opt      310     5834      6144      6%
```

4 To display the total number of blocks , the number of used blocks and the number of free blocks on a the /tmp file system, enter:

```
/usr/sysv/bin/df -v /tmp
```

5 To display the type of filesystem, enter:

```
/usr/sysv/bin/df -n
```

6 To display inode information on all local filesystems, enter:

```
/usr/sysv/bin/df -i -l
```

7 To display the statvfs structure information on all the filesystems, enter:

```
/usr/sysv/bin/df -g
```

8 To display the number of free files on filesystems, enter:

```
/usr/sysv/bin/df -e
```

/usr/sysv/bin/df Contains the System V df command.
/usr/bin/df command.

du Summarizes disk usage.

1 To summarize the disk usage of a directory tree and each of its subtrees, enter:

```
du /home/fran
```

This displays the number of disk blocks in the /home/fran directory and each of its subdirectories.

2 To summarize the disk usage of a directory tree and each of its subtrees in 1024-byte blocks, enter:

```
du -k /home/fran
```

This displays the number of 1024-byte disk blocks in the /home/fran directory and each of its subdirectories.

3 To summarize the disk usage of a directory tree and each of its subtrees in MB blocks, enter:

```
du -m /home/fran
```

This displays the number of MB disk blocks rounded off to nearest 2nd decimal digit in the /home/fran directory and each of its subdirectories.

4 To summarize the disk usage of a directory tree and each of its subtrees in GB blocks, enter:

```
du -g /home/fran
```

This displays the number of GB disk blocks rounded off to nearest 2nd decimal digit in the /home/fran directory and each of its subdirectories.

5 To display the disk usage of each file, enter:

```
du  -a /home/fran
```

This displays the number of disk blocks contained in each file and subdirectory of the /home/fran directory. The number beside a directory is the disk usage of that directory tree. The number beside a regular file is the disk usage of that file alone.

6 To display only the total disk usage of a directory tree, enter:

```
du  -s /home/fran
```

The -s flag instructs the du command to display only the sum total disk usage of the /home/fran

directory and the files it contains. By default, the du command displays an error message if it cannot read a file or directory.

7 To display the disk usage of the files and file hierarchies referenced by all the symbolic links in addition to the normal files found during traversal of a the /home/fran directory, type:

```
du -L /home/fran
```

8 To report the disk usage of the file or file hierarchy referenced by the symbolic link mylink, type:

```
du -H mylink
```

---

su Changes the user ID associated with a session.

1. To obtain root user authority, type:

```
su
```

This command runs a subshell with the effective user ID and privileges of the root user. You will be asked for the root password. Press End-of-File, Ctrl+D key sequence, to end the subshell and return to your original shell session and privileges.

2 To obtain the privileges of the jim user, type:

```
su jim
```

This command runs a subshell with the effective user ID and privileges of jim.

3 To set up the environment as if you had logged in as the jim user, type: su - jim

This starts a subshell using jim's login environment.

4 To run the backup command with root user authority and then return to your original shell, type:

```
su root "-c /usr/sbin/backup -9 -u"
```

This runs the backup command with root user authority within root's default shell. You must give the correct root password when queried for the command to execute.

---

tty Writes to standard output the full path name of your terminal.

1 To display the full path name of your display:

```
tty
```
