Unix Commands

Basic Commands

•pwd	Print Working Dir.
•cd	Change Dir.
•ls	List files
•cat	Concatenate files (Open file).
•more	Display content page by page.
•pg	Display content page by page.
•mv	rename file or move files.
•cp or copy	Copy the files.
•ln	Creat an alias or Link.
•mkdir	to creat new dir
•rm	to remove files & dir.
•rmdir	to removr dir.
•wc	to count words, lines, characters
•find	to search for a file.
•du	disk utilization.

FILE management commands

ls [Options]
List the Directory Contents

Options	**-1**	**number one single column output**
	-l	**long format (ll also used)**
	-a	**all entries including dot files**
	-s	**gives no. of disk blocks**
	-i	**inode no.**
	-t	**ordered by modification time recent first**
	-r	**recursively display all directories, starting from specified or current directory**

FILE management commands

- **cat**

  Concatenate & Print on screen or printer

  ***syntax* : $ cat <option> <filename>**

  Options

  - **take input from stdin**

  -n no. of output lines

  -s squeeze adj. blank lines

  -v enable display of non - printing characters

  -b used with -n to avoid numbering blank lines

eg. **: $ cat try.c**

Display the contents of try.c on the screen

$ cat > test.c

If test.c is not present, takes input from stdin

i.e. keyboard and displays on screen and creates new file test.c on pressing Cntrl+d at the end.

FILE management commands

cp

copy a file

-i **user interactive mode**

***eg.*:** **$ cp test.c test.c.bak**

test.c and test.c.bak contain the same contents ,extra disk storage

ln

Create link

***eg*.:** **$ ln first.c second.c**

The file is referenced by two different names ,no extra disk storage

FILE management commands

mv

Moves or renames files and directories

-i interactive mode

eg.: $ mv old.c new.c

Renames the file old.c as new.c

rm

Removes files and directories

-i removes interactively

-f forcible remove

-r remove recursively

- **Dangerous**
- **Used in conjunction with -i**

eg.: $ mv old.c

Removes old.c

FILE management commands

split

Splits the file into different files as specified by the number of lines

***eg.*: $ split - 20 test.c**

Splits the file test.c in blocks of 20 lines and creates files xaa, xab, xac and so on such that

xaa - has first 20 lines of test.c

xab - has the next 20 lines of test.c

The file test.c is unaffected

$ split-20 test.c try

Generates files as tryaa, tryab, tryac

paste

Joins two or more files horizontally

***eg.*: $ paste xaa xab**

File xaa and xab are joined horizontally and output is given to the terminal

BASIC UNIX COMMANDS

- **wc**

  count lines words & characters

  Syntax:- wc [option] [files]

  -c output character counts

  -l output line counts

  -w output word counts

- **du**

  display disk usage

- **df**

displays the number of free blocks for mounted file system

Syntax:- du [-sar] [names]

-a Generate entry for each file

-r Complain about directories that can't be read

-s Only display a grand total summary for each of the specified names

BASIC UNIX COMMANDS

passwd

To change user password

After giving the *passwd* command, the changing of password is confirmed by asking the old password and then for the new password

***syntax* : $ passwd**

***eg.* : $ passwd**

passwd: changing password for user1

Old password :

New password :

Re-enter new password :

BASIC UNIX COMMANDS

man

Displays online helps for the specific command

- **It displays the online manual pages associated with the mentioned command**
- **It is handy to look for the exact syntax of a command or the arguments associated with a command**

  syntax **: man < command >**

BASIC UNIX COMMANDS

who

Who has logged into the system

Lists the user's who are currently logged in on to the UNIX system, their terminal number and login time.

***eg.* : $ who**

```
user1 tty1   Jan 14    12:01
user2 tty2   Jan 14    12:45
```

finger

More information about any logged in user

Displays additional information like home directory, shell etc about the logged in user.

***syntax*: finger < username >**

***eg.* : $ finger user1**

Login name : user1

Directory :/users/user1 Shell : /usr/bin/sh

Last login : Thurs Jan 14 12

Directory Management Command

cd

To change directory

syntax **: cd < directory name > -**

To change to a specific directory

cd.. - To change to parent directory

cd. - To change to current directory

e.g.**: $ cd / usr / trg / c (current Directory is *c*)**

$ cd.. (current Directory is *trg*)

$ cd ./c (current Directory is again *c*)

or $ cd c

$ cd (home directory-in this case /usr/trg)

mkdir

Make a Directory

***syntax* : $ mkdir < pathname >**

***eg.* : mkdir \trg2**

makes directory trg2 in 777 mode ie. with all permissions to all

pwd

Print present working directory

***syntax* : $ pwd**

- **Redirecting output and input**
- **In UNIX system, output or input of the command can be redirected into or from any file on the disk instead of to or from any physical device**
- **Different redirection commands are**

Symbol	Syntax	Output
>	$ command > filename	Output of the command is sent to the filename
<	$ command < $	Input from the filename is used command < filename by the command
>>	$ command >>	Output of command is appended $ command >>filename to the filename Redirecting output and input

eg. : $ who > test Output of the who command is inputed into the file test, and the old contents of the test are lost

$ ls -l >> test Output of the ls command is inputed into the test file after it is appended and the old contents of the test file is retained

$ cat < test cat command takes input from the test file and it displays the contents of the test file

$ cat< test > test1 cat command takes input from the test file and outputs it to the file test2 instead of displaying it on the screen

UNIX TOOLS

- grep
- grep searches for text in either a single file or in multiple number of files.
- It goes through the file and returns the lines containing specified text
- It is useful for locating the line of text that contain a specified text pattern
- *syntax :* $ grep pattern < file 0 >
- where all the lines in a file containing the pattern are sent to the standard output

eg. **: grep abc test.c**

grep looks for the word *abc* in the test.c

- **To search a phrase with a space, it must be enclosed in the quotation marks**

eg.**: grep "ab 1" test.c**

grep looks for the phrase *ab 1* in the file test.c

- **If the file is not specified on the command line it takes standard input file as an input**

File search command

find

- **To locate file or group of files**
- ***syntax* : $ find <directory> <parameters>**
- **directory must be the directory to be searched for the required file**
- **All the subdirectories in a directory are also searched**
- **parameters are special names and values that tell what to search for and they are :**

-name file	**search for the filename with specified pattern**
-atime	**search for file not accessed for specified time**
-group gname	**search for file belonging to specified group**
-links n	**search for files with n number of links**
-print	**prompts to display location found files**

***eg.*: $ find /usr -name s* -atime +7 -print**

It will display files starting with s and is not accessed for 7 days in the /usr directory

FILE management commands

tail

Display the last lines of a file

options **-n (n = no. of lines)**

eg.,**: $ tail - 30 test.c**

Displays the last 30 lines of file test. c

head

Display the top lines of a file

eg. **: $ head - 10 test.c**

Displays the first 10-lines of test.c

UNIX TOOLS

find

This command helps to locate the file.

***eg*.: To find a file named test.data, use following command**

$ find / -name TEST.data -print

banner

Displays upto 10 characters in large letters

***eg.* : $ banner CMS**

Displays CMS in large letters

bc

Calculator in unix system

cal

This command displays the calendar

***eg.*: $ cal 1993**

Displays the calendar for an entire year.

UNIX TOOLS

grep

grep searches for text in either a single file or in multiple number of files.

- **It goes through the file and returns the lines containing specified text**
- **It is useful for locating the line of text that contain a specified text pattern**

  syntax : **$ grep pattern < file 0 >**
- **where all the lines in a file containing the pattern are sent to the standard output**

  eg. **: grep abc test.c**

- grep looks for the word *abc* in the test.c
- To search a phrase with a space, it must be enclosed in the quotation marks

  eg.: grep "ab 1" test.c
- grep looks for the phrase *ab 1* in the file test.c
- If the file is not specified on the command line it takes standard input file as an input

UNIX TOOLS

sort

It arranges the lines in a file in the specified order*syntax :* $ sort -options < filename >

Options

-n sort the file by numeral order

-r sort the file in reverse order

cmp

It compares two files, and if the files are different it reports the first instance of a difference

It does not report all the differences between the files

***syntax :* cmp <file1> <file2>**

diff

- **It compares two files and reports each difference in the files**
- **It reports significantly more information than does cmp**

  syntax : **diff <file1> <file2>**

Pipelines

- **This is the processing of data from the output of the last command without storing the data on the disk as a file.**

  ***syntax* : $ Command 1 | Command 2 Pipe symbol tells the shell to connect the standard output of command 1 directly to the standard input of command 2.**

- **The output of command 1 does not appear on the terminal nor it is stored in some intermediate temporary file, instead Unix system "buffers" the output of the command1 as the input to the second command 2**

- **All the commands in the pipeline are invoked at the same time, and each command process begins working as soon as some input data is available to it, thus command 2 can begin work even before command 1 has finished executing, this helps to speed up the overall processing**

eg.: $ who | wc -l

- **Ouput of the who command is given to the wc command without being displayed on the screen and stored on the disk, finally only the output of the wc command is displayed**

PIPELINE WITH tee

- **With tee command it makes possible to send a copy of the data passing through a pipeline to a file on disk or to your terminal screen.**

syntax **: $ command 1 | tee filename | command 2**

- **Here the standard output of command1 becomes the standard input for tee**
- **The data is passed unchanged from the standard input to the standard output of tee, but a copy is stored in file.**
- **The standard output of tee is connected to the standard input of command 2.**
- **Command 1 tee Command 2file**

chmod

Change the permissions of the file

***syntax* : $ chmod who op permission <file list>**

where *who* is for which user

a : all
u : user
g : group
o : others

where *op* is what option

+ :add permissions
- : remove permissions
= : set permissions

where *permission* is which file permission

r : read
w : write
x : execute

FILE SECURITY COMMANDS

***eg*.:**

 # chmod a = rw test.c

Sets

users = read and write permissions
group = read and write permissions
others = read and write permissions

 # chmod u + r, g + w, o + x test.c

Sets

users = read permission
group = write permission
others = execute permission

 # chmod 777 test.c

Sets read, write, execute permissions for all

FILE SECURITY COMMANDS

chown

To change the ownership of the file

- **Only the owner or the superuser can change the ownership of the file**

syntax **: chown < owner > < filename >**

eg. **: # chown trg2 test.c**

Initially the owner is trg1 now it will be trg2

FILE SECURITY COMMANDS

Security for files in sticky directories

- A directory with the sticky bit set means that only the file owner and the super user can remove files from that directory, other users are denied the right to remove files irrespective of their direct permissions
- Only the superuser can place the sticky bit on a directory

- **Unlike with files, the sticky bit on directories remains there until the directory owner or super user removes the sticky bit using chmod**
- **A sticky directory contains a "t" at the end of the permisions field**

eg. **: $ ls -l**

drwxrwxrwxt 2 bin bin 1088 Mar 18 21:10 tmp

FILE SECURITY COMMANDS

su

using another account

- **The su command allows user to become another user without logging off**
- **su cannot be used to simply assume the login of another user, instead su can be used under four following circumstances :**
- **The super user can "su" to any account**
- **An administrative user with the su authorization can "su" to the super user account**

- A user can "su" to their own account
- A system daemon can "su" to an account
- To use su, the appropriate password must be supplied
- If the password is correct, su executes a new shell with the effective user ID set to that of the specified user.

FILE SECURITY COMMANDS

crypt

To encode and decode files

- **When encrypting a file using crypt command a password has to be assigned**
- **Same password is used to decrypt the encrypted file**
- **Without the valid password the contents of the encrypted file cannot be seen**

FILE SECURITY COMMANDS

- **To encrypt a file**

  syntax **: crypt < oldfile > newfile**

  then system prompts to enter the password

Note : Always have an habit to delete the oldfile after checking the encrypted file, because the Oldfile is not encrypt and can be accessed by anyone

eg. **: crypt test.c test.zip**

test.c is encrypted as test.zip

- **To decrypt a file**

  syntax **: crypt < crypted_file > new_filename**

  then system prompts for valid password

  eg. **: crypt test.zip**

  test.zip is decrypted

FILE SECURITY COMMANDS

pack

Compresses the file

Normally the executables are packed

The original file size is reduced by 25 - 40%

***syntax* : $ pack < filename >**

***eg.*: $ pack try**

Creates a file try.z which is packed

unpack

Decompress the file

***syntax* : $ unpack < filename >**

***eg.*: $ unpack try.z**

unpacks the file try.z

FILE SEARCH COMMAND

find

To locate file or group of files

syntax **: $ find <directory> <parameters>**

- **directory must be the directory to be searched for the required file**
- **All the subdirectories in a directory are also searched**
- **parameters are special names and values that tell what to search for and they are :**

FILE SEARCH COMMAND

-name file	search for the filename with specified pattern
-atime	search for file not accessed for specified time
-group gname	search for file belonging to specified group
-links n	search for files with n number of links
-print	prompts to display location found files

eg.: $ find /usr -name s* -atime +7 -print

It will display files starting with s and is not accessed for 7 days in the /usr directory FILE SEARCH COMMAND

Logging out of the unix

- There are two different ways to log out from the UNIX
- Through exit command
- User can also log out by pressing CNTR +D
- When user logs out UNIX system frees that terminal, to login again