Basic Unix Command

The Unix command has the following common pattern

command_name *options* argument(s)

Here we are trying to give some of the basic unix command in Unix

Information Related

man	It is used to see the manual of the various command. It helps in selecting the correct options to get the desirable output. Just need to give man command
finger	Getting information about a local and remote user. Sometimes you may want to know what all user are currently logged into the Unix system then this will help
passwd	Changing your password for the user. If you are logged in, then enter passwd,it will change the password for the current user. From root login you can change password for any user by using passwd
who	Finding out who is logged on
which	find the source directory of the executable being used

uname	print name of current system. On Solaris uname -X will give the cpu and various other informations
uptime	Time since the last reboot

File listing and directory manipulation command

ls	List contents of Directory(s) Syntax: ls options OPTIONS The following options are supported:-a Lists all entries, including those that begin with a dot (.), which are normally not listed. -d If an argument is a directory, lists only its name (not its contents); often used with -l to get the status of a directory. -n The same as -l, except that the owner's UID and group's GID numbers are printed, rather than the associated character strings. -r Reverses the order of sort to get reverse alphabetic or oldest first as appropriate. -R Recursively lists subdirectories encountered.-t Sorts by time stamp (latest first) instead of by name. The default is the last modification time. -h Give the file size in Giga bytes
mkdir	Creates directory(s) or directory structure Syntax: - mkdir Options: m & p Option p checks if the directory already exists,then leave it without issuing any error. If it does not exist, it will create it

rm, rmdir	remove directory entries -r Recursively remove directories and subdirectories. -i Interactive. With this option, rm prompts for confirmation before removing any files. -f Remove all files (whether write-protected or not) in a directory without prompting the user Example rm -f * Remove all files in the working directory. rm will not prompt you for any reason before deleting them rm -i * Attempt to remove every file in the working directory, but prompt before each file to confirm.
cd	Changes current working directory to the directory specified Syntax: - cd dirname [cd without any arguments will changes current working directory to home directory]
pwd	Prints full path name of the current working directory
cp	Options: -i Interactive. cp will prompt for confirmation whenever the copy would overwrite an existing target. -r Recursive. cp will copy the directory and all its files, including any subdirectories and their files to target. -p Preserve. cp duplicates not only the contents of source_file, but also preserves the owner and group id, permissions modes, modification and access time.

mv	move files(s) /dir(s) Options: - -f mv will move the file(s) without prompting even if it is writing over an existing target. -i mv will prompt for confirmation whenever the move would overwrite an existing target. Examples: mv –i x1 x2 [moves x1 to x2, if x2 exists prompts for confirmation.] mv *lst dst_directory. [moves all files like lst to dst_directory.]

mv - move files(s) /dirs

Options: -

-f mv will move the file(s) without prompting even if it is writing over an existing target.

-i mv will prompt for confirmation whenever the move would overwrite an existing target

Examples

1 mv –i f1.txt f2.txt

[moves f1.txt to f2.txt, if f2.txt exists prompts for confirmation.]

2. mv file1 file2 file3 dest_directory.

[moves file1 to 3 to dst_directory.]

3. mv –r oracle/app /u000

[moves directory app and all its subdirectories files to /u000 directory]

4. mv f1.txt oracle/f2.txt

[moves directory app and all its subdirectories files to /u000 directory]

file, head, tail & alias commands

file - determine file type
Syntax: file <file_name>

Example

```
$ file workfile: ascii text
```

head - display first few lines of files
Syntax: head –n <file_name>

Example

```
$ head -10 workfile NDAYS = 10 User "oracle" does not exist. expr: syntax error

User "oracle" does not exist.

expr: syntax error
```

tail - displays the last part of a file
Syntax: tail –n <file_name>
Other Option: -f
Example

```
$ tail -f check_password_oracle.log

User "oracle" does not exist.

expr: syntax error

//tmp/x: test: argument expected NDAYS = 16990 16961 13 9
```

Defining command aliases:
Remembering the name of a command and how it is used can be difficult. Assigning your own name for a command - an alias - is very easy.
Assigning an alias is done with the command:
alias name definition
unalias name
e. g: alias del='rm -I'

How to get History of last commands

Using the command history

Each time that you enter a command it is added to a command history list. You can reuse commands from this list

Example

```
$ history

1       cd /tmp

2       ls

3       file workfile

4       file ext.1.out

5       ls -lrtt check_password_expire_.log

6       file  check_password_expire_.log

7     head -10 check_password_expire.log

8       tail -f check_password_expire_oracle_grid.log

9       history
```

Cut Command

cut command displays selected columns or fields from each line of a file.
cut -clist [file_list]
cut -flist [-dchar] [-s] [file_list]
options:
-clist Display (cut) columns, specified in list, from the input data. No spaces are allowed within the list.
Multiple values must be comma (,) separated. The list defines the exact columns to display.
For example,
-c1,4,7 notation displays columns 1, 4, and 7 of the input.
-c1-10,50 format would display columns 1 through 10 and 50 through end-of-line.

```
Example

$ cut -c2 file.txt

a

b

c

It displays the second character

grep '^Subject:' read-messages | cut -c10-80

It displays the character from 10-80
```

-flist Display (cut) fields, specified in list, from the input data.
No spaces are allowed within the list. Multiple values must be comma (,) separated. The list defines the exact field to display.
For example, -f1,4,7 would display fields 1, 4, and 7. The -f2,4-6,8 would display fields 2, 4, 5, 6, and 8.
dchar The character char is used as the field delimiter. The default delimiter is a tab character. To use a character that has special meaning to the shell, you must quote the character so the shell does not interpret it.
For example, to use a single space as a delimiter, type -d' '

Examples

```
cat file1.txt | cut -f1,2 -d":"

cut -f -3,5,7-9 -d ' ' infile1 > outfile1

ls –l | cut –f5 –d” “



grep "/bin/bash" /etc/passwd | cut -d':' -f1,6

root:/root

tech:/home/tech
```

Sort Command

sort - sort command sorts data. If multiple input files are given, the data from each file is merged during the sort. You use the sort command to sort data alphabetically or numerically, in ascending or descending order. You can sort based on entire lines, fields, or character columns. You can merge files using sort and remove duplicate lines with it.

Options

- Forces sort to read from the standard input. Useful for reading from pipes and files simultaneously.

-m Merges the sorted input.

-o output Sends the output to file output instead of the standard output.

-u Suppress all but one occurrence of matching keys.

-n Restricts the sort key to an initial numeric string

-r Reverses the sense of comparisons.

-t char Use char as the field separator character. char is not considered to be part of a field (although it can be included in a sort key). if - t is not specified, blank characters are used as default field separators

-k field_start[type][,field_end[type]]

It defines a key field that begins at field_start and ends at field_end inclusive

type - is a modifier from the list of characters bdfiMnr. They have this effect if specified with field_start, field_end or both.

-[+pos1[-pos2]] - Provide functionality equivalent to the -k keydef option.
pos1 and pos2 each have the form m.n optionally followed by one or more of the flags bdfiMnr. A starting position specified by +m.n is interpreted to mean the n+1st character in the m+1st field.
A last position specified by -m.n is interpreted to mean the nth character (including separators) after the last character of the mth field.

Examples
sort -r f1.txt -o o.txt [sorts f1.txt and stores result in o.txt]
sort -r -o outfile -k 2.2,2.2 f1.txt f2.txt - sorts, in reverse order, the contents of f1.txt and f2.txt, placing the output in outfile and using the second character of the second field as the sort key

sort -r -o outfile +1.1 -1.2 f1 infile2 – same result as was with –k option.
sort +1 -2 infile - sorts the contents of infile with the second field as the sort key
Either of the following commands sorts the contents of infile1 and infile2 using the second non-blank character of the second field as the sort key:
sort -k 2.2b,2.2b infile1 infile2 or sort +1.1b -1.2b infile1 infile2
Either of the following commands prints the passwd file sorted by the numeric user ID (the third colon-separated field):
sort -t : -k 3,3n /etc/passwd or sort -t : +2 -3n /etc/passwd

Who Command

who [options] to see who is logged in to the computer.
who -T Shows the IP address of each connection
who -r Shows when the computer was last rebooted, run-level.

```
$ who -r

.       run-level 2 Jun 8 00:23      2    0    S

you have mail in /usr/spool/mail/
```

env command

To see value of all environment variables.
To set an environment variable:

In ksh or sh

"export VARIABLENAME=value"

Example

export ORACLE_SID=TECH
In csh "setenv VARIABLENAME value"

Example

setenv ORACLE_SID TECH
echo $VARIABLENAME See value of an environment variable

Example

echo $ORACLE_SID