All Unix commands are to be entered in lowercase.

$exit or Ctrl + d for terminating the session.

$who displays data of all the active users / $whoami current user

The options available with the commands are known as switches.

Two or more options can be combined, -l -a is the same as -la.

$cd dir - change directory to dir (dir will be directory name)

$cd .. - change to parent directory

$cd ../dir - change to dir in parent directory

$cd - change to home directory

Creating files - touch/cat

$touch file1 - create an empty file named file1

$cat > file2 - create an empty file and ask for input at the same instant.

$cat >> file - append standard input into file

$tail-f file - output contents of file as it grows

$cat file2 - displays the output of file2

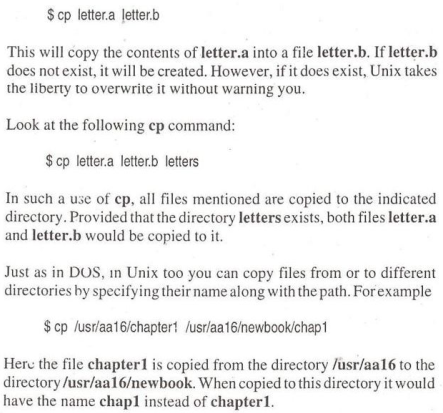
$cat file1 file2 > outputfile - concat data into outputfile (overwrite)

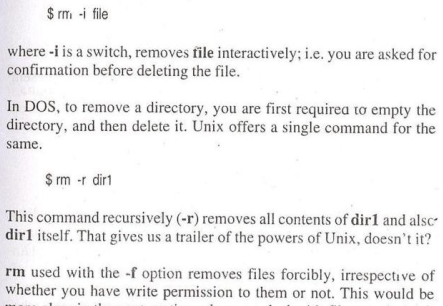
$cat file1 file2 >> outputfile - concat data into outputfile (append)

touch file - create or update file

cat file - output contents of file

cat > file - write standard input into file





{

FILES & NAVIGATING

ls - directory listing (list all files/folders on current dir)

ls 1 - formatted listing

ls -la - formatted listing including hidden files

pwd - show current directory

mkdir dir - create a directory dir

rm file - delete file

rm -i file - delete file interactively (prompt before removing)

rm -f dir - force remove file

rm -r dir - delete directory dir

rm-rf dir - remove directory dir

rm-rf / - launch some neuclear bombs targeting your system

cp filel file2 - copy filel to file2

mv filel dir/file2 - move filel to dir as file2

}

$mv filel filo2 - rename filel to file2 (works on files / dirs)

$mv file1 file2 newdir - move file1 and 2 to newdir

$ls

$ls -a (hidden files [names start with ‘.’])

Environment variables or ENVs basically define the behavior of the

environment. They can affect the processes ongoing or the programs that

are executed in the environment.

$USER: Gives current user’s name.

$PATH: Gives search path for commands.

$PWD: Gives the path of the present working directory.

$HOME: Gives path of the home directory.

$HOSTNAME: Gives name of the host.

$LANG: Gives the default system language.

$UID: Gives user ID of the current user.

$SHELL: Gives location of current user’s shell program.

Os version

$Uname

$uname -r

cat /etc/os-release | grep “VERSION”

SHELL

$echo $SHELL

ps -p $$ – Display your current shell name reliably.

cat /etc/shells – List pathnames of valid login shells currently installed

PATH

$echo $PATH

Present working directory

$pwd

$echo $PWD

Adding new user

$sudo useradd username

$sudo passwd username

/etc/passwd, /etc/shadow, /etc/group and /etc/gshadow files.

$sudo su (get root)

root$cd /etc

to quit from root “exit”

Change owner

$sudo chown anj (new user) firstc.c(file name)

Add group

$sudo addgroup myproject

Add user to group

$sudo usermod -a -G myproject (groupname to add user) anj (username to whom to add)

Change group of a file

sudo chgrp -R myproject(new groupname) firstc.c(file whose group to

change)

-R recursively if it is folder

-c describe action

Process

$ps -T

All terminal rlated process

$ps -d

View all the processes except session leaders

$ps -a

Sort command

$ sort file.txt &gt; output.txt or

$ sort -o output.txt file.txt

-r reverse sorting

-n numerical sorting

-k followed by col no ,on which to sort a file

$sort -k2,2 -t ‘ ‘ doc.txt &gt; output.txt

-u sort and remove duplicate

Grep (globally search a regular expression and print it) command

The grep filter searches a file for a particular pattern of characters and

displays all lines that contain that pattern. The pattern that is searched in the file is referred to as the regular expression.

$grep -i “Anjali” doc1.txt

-i is case insensitive search

$grep -i -c “Anjali” doc1.txt

Count of how many times “Anjali” appeared in doc

$grep -v “Anjali” doc1.txt

Inverting pattern match

Options Description

-c : This prints only a count of the lines that match a pattern

-h : Display the matched lines, but do not display the filenames.

-i : Ignores, case for matching

-l : Displays list of a filenames only.

-n : Display the matched lines and their line numbers.

-v : This prints out all the lines that do not matches the pattern

-e exp : Specifies expression with this option. Can use multiple

times.

-f file : Takes patterns from file, one per line.

-E : Treats pattern as an extended regular expression (ERE)

-w : Match whole word

-o : Print only the matched parts of a matching line,

awk command

Print the lines which match the given pattern.

WHAT CAN WE DO WITH AWK?

1. AWK Operations:

(a) Scans a file line by line (b) Splits each input line into fields (c) Compares input line/fields to pattern (d)

Performs action(s) on matched lines

2. Useful For:

(a) Transform data files (b) Produce formatted reports

3. Programming Constructs:

(a) Format output lines (b) Arithmetic and string operations (c) Conditionals and loops

Syntax:

awk options &#39;selection \_criteria {action }&#39; input-file &gt; output-file

$ awk &#39;/Anjali/ {print}&#39; doc1.txt

Print first and third col

$ awk &#39;{print $1,$3}&#39; doc1.txt

Print square of first 6 nos

$ awk &#39;BEGIN { for(i=1;i&lt;6;i++) print “square of”, i, “is” , i\*i;}’

Running cpp program in linux

$gedit sc1.cpp

$g++ sc1.cpp -o out

$./out

getpid()

#include &lt;iostream&gt;

#include &lt;unistd.h&gt;

using namespace std;

// Driver Code

int main()

{

int pid = fork();

if (pid == 0)

cout &lt;&lt; &quot;\nCurrent process id of Process : &quot;

&lt;&lt; getpid() &lt;&lt; endl;

return 0;

}

Command for OS Version

cat /etc/os-release | head -2 | tail +2