1. **Study of Basic commands of Linux/UNIX.**

**Files and Directories:**

These commands allow you to create directories and handle files.

**Command** **Description**

cat Display File Contents

cd Changes Directory to dirname

chgrp change file group

chmod Changing Permissions

cp Copy source file into destination

file Determine file type

find Find files

grep Search files for regular expressions.

head Display first few lines of a file

ln Create softlink on oldname

ls Display information about file type.

mkdir Create a new directory dirname

more Display data in paginated form.

mv Move (Rename) a oldname to newname.

pwd Print current working directory.

rm Remove (Delete) filename

rmdir Delete an existing directory provided it is empty.

tail Prints last few lines in a file.

touch Update access and modification time of a file.

**Manipulating data:**

The contents of files can be compared and altered with the following commands.

**Command** **Description**

awk Pattern scanning and processing language

cmp Compare the contents of two files

comm Compare sorted data

cut Cut out selected fields of each line of a file

diff Differential file comparator

expand Expand tabs to spaces

join Join files on some common field

perl Data manipulation language

sed Stream text editor

sort Sort file data

split Split file into smaller files

tr Translate characters

uniq Report repeated lines in a file

wc Count words, lines, and characters

vi Opens vi text editor

vim Opens vim text editor

fmt Simple text formatter

spell Check text for spelling error

ispell Check text for spelling error

ispell Check text for spelling error

emacs GNU project Emacs

ex, edit Line editor

emacs GNU project Emacs

emacs GNU project Emacs

**Compressed Files:**

Files may be compressed to save space. Compressed files can be created and examined:

**Command** **Description**

compress Compress files

gunzip Uncompress gzipped files

gzip GNU alternative compression method

uncompress Uncompress files

unzip List, test and extract compressed files in a ZIP archive

zcat Cat a compressed file

zcmp Compare compressed files

zdiff Compare compressed files

zmore File perusal filter for crt viewing of compressed text

**Getting Information:**

Various Unix manuals and documentation are available on-line. The following Shell commands give information:

**Command** **Description**

apropos Locate commands by keyword lookup

info Displays command information pages online

man Displays manual pages online

whatis Search the whatis database for complete words.

yelp GNOME help viewer

**Network Communication:**

These following commands are used to send and receive files from a local UNIX hosts to the remote host around the world.

**Command** **Description**

ftp File transfer program

rcp Remote file copy

rlogin Remote login to a UNIX host

rsh Remote shell

tftp Trivial file transfer program

telnet Make terminal connection to another host

ssh Secure shell terminal or command connection

scp Secure shell remote file copy

sftp secure shell file transfer program

Some of these commands may be restricted at your computer for security reasons.

**Messages between Users:**

The UNIX systems support on-screen messages to other users and world-wide electronic mail:

**Command** **Description**

evolution GUI mail handling tool on Linux

mail Simple send or read mail program

mesg Permit or deny messages

parcel Send files to another user

pine Vdu-based mail utility

talk Talk to another user

write Write message to another user

**Programming Utilities:**

The following programming tools and languages are available based on what you have installed on your Unix.

**Command** **Description**

dbx Sun debugger

gdb GNU debugger

make Maintain program groups and compile programs.

nm Print program's name list

size Print program's sizes

strip Remove symbol table and relocation bits

cb C program beautifier

cc ANSI C compiler for Suns SPARC systems

ctrace C program debugger

gcc GNU ANSI C Compiler

indent Indent and format C program source

bc Interactive arithmetic language processor

gcl GNU Common Lisp

perl General purpose language

php Web page embedded language

py Python language interpreter

asp Web page embedded language

CC C++ compiler for Suns SPARC systems

g++ GNU C++ Compiler

javac JAVA compiler

appletvieweir JAVA applet viewer

netbeans Java integrated development environment on Linux

sqlplus Run the Oracle SQL interpreter

sqlldr Run the Oracle SQL data loader

mysql Run the mysql SQL interpreter

**Misc Commands:**

These commands list or alter information about the system:

**Command** **Description**

chfn Change your finger information

chgrp Change the group ownership of a file

chown Change owner

date Print the date

determin Automatically find terminal type

du Print amount of disk usage

echo Echo arguments to the standard options

exit Quit the system

finger Print information about logged-in users

groupadd Create a user group

groups Show group memberships

homequota Show quota and file usage

iostat Report I/O statistics

kill Send a signal to a process

last Show last logins of users

logout log off UNIX

lun List user names or login ID

netstat Show network status

passwd Change user password

passwd Change your login password

printenv Display value of a shell variable

ps Display the status of current processes

ps Print process status statistics

quota -v Display disk usage and limits

reset Reset terminal mode

script Keep script of terminal session

script Save the output of a command or process

setenv Set environment variables

stty Set terminal options

time Time a command

top Display all system processes

tset Set terminal mode

tty Print current terminal name

umask Show the permissions that are given to view files by default

uname Display name of the current system

uptime Get the system up time

useradd Create a user account

users Print names of logged in users

vmstat Report virtual memory statistics

w Show what logged in users are doing

who List logged in users

2. **Study of Advance commands and filters of Linux/UNIX.**

**The grep Command:**

The grep program searches a file or files for lines that have a certain pattern. The syntax is:

$grep pattern file(s)

The name "grep" derives from the ed (a UNIX line editor) command g/re/p which means "globally search for a regular expression and print all lines containing it."

A regular expression is either some plain text (a word, for example) and/or special characters used for pattern matching.

The simplest use of grep is to look for a pattern consisting of a single word. It can be used in a pipe so that only those lines of the input files containing a given string are sent to the standard output. If you don't give grep a filename to read, it reads its standard input; that's the way all filter programs work:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| $ls -l | grep "Aug" | | |  |  |  |  |  |
| -rw-rw-rw- | 1 | john | doc | 11008 | Aug | 6 | 14:10 ch02 |
| -rw-rw-rw- | 1 | john | doc | 8515 | Aug | 6 | 15:30 ch07 |
| -rw-rw-r-- | 1 | john | doc | 2488 | Aug | 15 | 10:51 intro |
| -rw-rw-r-- | 1 | carol doc | | 1605 | Aug | 23 | 07:35 macros |
| $ |  |  |  |  |  |  |  |

There are various options which you can use along with grep command:

**Option -v**

**-n**

**-l**

**Description**

Print all lines that do not match pattern. Print the matched line and its line number.

Print only the names of files with matching lines (letter "l")

**-c** Print only the count of matching lines. **-i** Match either upper- or lowercase.

Next, let's use a regular expression that tells grep to find lines with "carol", followed by zero or more other characters abbreviated in a regular expression as ".\*"), then followed by "Aug".

Here we are using *-i* option to have case insensitive search:

$ls -l | grep -i "carol.\*aug"

-rw-rw-r-- 1 carol doc 1605 Aug 23 07:35 macros

$

**The sort Command:**

The **sort** command arranges lines of text alphabetically or numerically. The example below sorts the lines in the food file:

$sort food Afghani Cuisine Bangkok Wok Big Apple Deli Isle of Java Mandalay

Sushi and Sashimi Sweet Tooth

Tio Pepe's Peppers

$

The **sort** command arranges lines of text alphabetically by default. There are many options that control the sorting:

|  |  |  |
| --- | --- | --- |
| **Option** | **Description** |  |
| **-n** | Sort numerically (example: 10 will sort after 2), ignore blanks and |  |
| tabs. |  |
|  |  |

**-r** Reverse the order of sort.

**-f** Sort upper- and lowercase together. **+x** Ignore first x fields when sorting.

More than two commands may be linked up into a pipe. Taking a previous pipe example using **grep**, we can further sort the files modified in August by order of size.

The following pipe consists of the commands **ls, grep,** and **sort**:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| $ls -l | grep | "Aug" | | sort +4n | |  |  |  |
| -rw-rw-r-- 1 | carol | doc | 1605 | Aug | 23 | 07:35 macros |
| -rw-rw-r-- 1 | John | doc | 2488 | Aug | 15 | 10:51 intro |
| -rw-rw-rw- 1 | John | doc | 8515 | Aug | 6 | 15:30 ch07 |
| -rw-rw-rw- 1 | John | doc | 11008 | Aug | 6 | 14:10 ch02 |
| $ |  |  |  |  |  |  |

This pipe sorts all files in your directory modified in August by order of size, and prints them to the terminal screen. The sort option +4n skips four fields (fields are separated by blanks) then sorts the lines in numeric order.

**The pg and more Commands:**

A long output would normally zip by you on the screen, but if you run text through more or pg as a filter, the display stops after each screenful of text.

Let's assume that you have a long directory listing. To make it easier to read the sorted listing, pipe the output through **more** as follows:

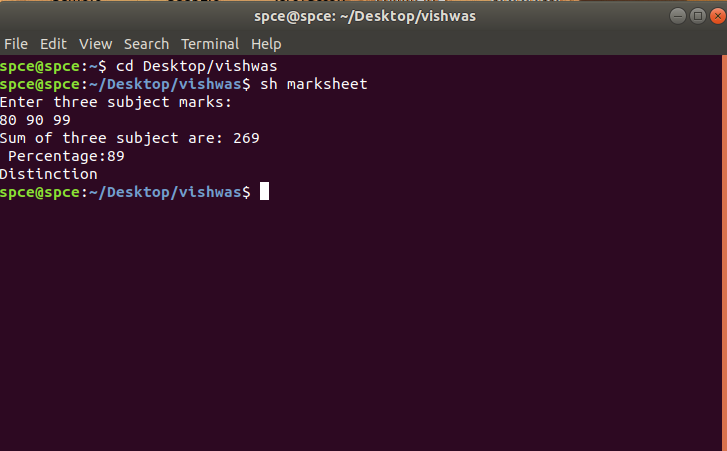
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| $ls -l | grep | "Aug" | | sort +4n | more | | |  |  |
| -rw-rw-r-- 1 | carol | doc | 1605 | Aug | 23 | 07:35 macros |
| -rw-rw-r-- 1 | John | doc | 2488 | Aug | 15 | 10:51 intro |
| -rw-rw-rw- 1 | John | doc | 8515 | Aug | 6 | 15:30 ch07 |
| -rw-rw-r-- 1 | John | doc | 14827 | Aug | 9 | 12:40 ch03 |
| . |  |  |  |  |  |  |
| . |  |  |  |  |  |  |
| . |  |  |  |  |  |  |
| -rw-rw-rw- 1 | John | doc | 16867 | Aug | 6 | 15:56 ch05 |
| --More--(74%) |  |  |  |  |  |  |

The screen will fill up with one screenful of text consisting of lines sorted by order of file size. At the bottom of the screen is the **more** prompt where you can type a command to move through the sorted text

3. **Write a shell script to generate marksheet of a student. Take 3 subjects, calculate and display total marks, percentage and Class obtained by the student.**

echo "Enter three subject marks:"  
read m1 m2 m3  
sum=`expr $m1 + $m2 + $m3`  
echo "Sum of three subject are:" $sum  
per=`expr $sum / 3`  
echo " Percentage:"$per  
if [ $per -ge 70 ]  
then  
echo "Distinction"  
elif [ $per -ge 60 ]  
then  
echo "First Class"  
elif [ $per -ge 50 ]  
then  
echo "Second Class"  
elif [ $per -ge 40 ]  
then  
echo "Pass Class"  
else  
echo "FAIL"  
fi

**OUTPUT:**

****

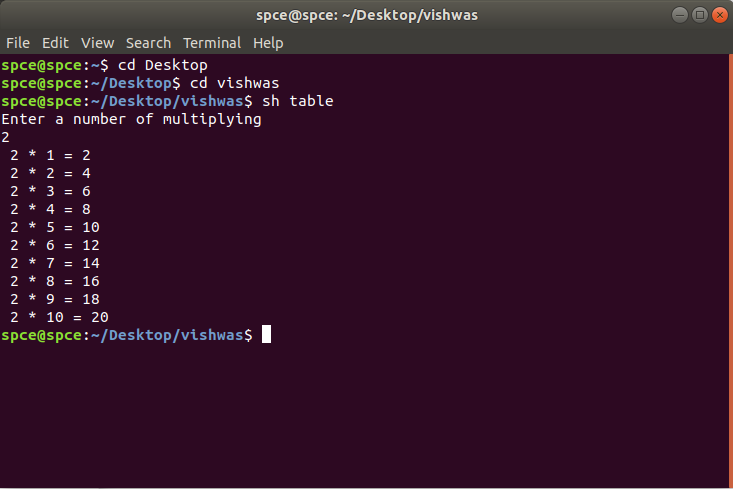
**4. Write a shell script to display multiplication table of given number**

echo "Enter a number of multiplying"

read a  
i=1  
while [ $i -le 10 ]  
do  
echo " $a \* $i = `expr $a \\* $i`"  
i=`expr $i + 1`

done

**OUTPUT:**

****

**5. Write a shell script to find factorial of given number n.**

echo "Enter a Number"  
read n  
fact=1  
while [ $n -gt 1 ]  
do  
fact=$((fact \* n))  
n=$((n - 1))  
done  
echo $fact

**OUTPUT:**

