Part A

1. Execution of various file/directory handling commands.

Change directory (cd):

[Student@localhost /]$ cd .. /\* is current working Directory\*/

Student@localhost

[Student@localhost /]$ cd / /\* change directory to root directory\*/

[LabExam@ISELAB1 local]$ cd ~ /\*~(Tilde) takes you back to the home directory\*/

[LabExam@ISELAB1 ~]$ cd - /\*switching between present and previous directories or (-)hyphen will take you to the last directory which you have worked on

\*/

/home/LabExam

[Student@localhost /]$ cd /usr/local /\*Using Absolute Path, We give entire path details here.\*/

[Student@localhost lib]$ cd /usr/local /\* Change Directory Again\*/

[Student@localhost local]$ cd lib /\*Using Relative Path, We give only the sub directory name to which we need to change\*/

Present working directory(pwd):

[Student@localhost lib]$ pwd /\*Present Working Directory, Prints the full path to the current Working Directory\*/

Output :/usr/local/lib

Ls command: listing files and directories

[Student@localhost local]$ ls /\*Lists the files and Directories under the current working directory

Output:

bin etc games include lib libexec sbin share src

[Student@localhost local]$ ls -t /\*List in the order of last modification time\*/

Output:

share bin etc games include lib libexec sbin src

[Student@localhost local]$ ls -l /\* List all the files, Directories and their modes, number of links,owner of the file, file size, modified date and time and file name\*/

total 36

Output:

drwxr-xr-x. 2 root root 4096 Oct 1 2009 bin

drwxr-xr-x. 2 root root 4096 Oct 1 2009 etc

drwxr-xr-x. 2 root root 4096 Oct 1 2009 games

[Student@localhost local]$ ls -a /\*Lists all entries including hidden files\*/

. .. bin etc games include lib libexec sbin share src

Single dot represents hidden directory, double dot represents the parent of the hidden directory

[Student@localhost local]$ ls -d /\*Lists the directory files instead of the contents, here there are no contents.

Output:

.

[Student@localhost local]$ ls -p /\*Puts slash at the end of each directory\*/

bin/ etc/ games/ include/ lib/ libexec/ sbin/ share/ src/

[Student@localhost local]$ ls -u /\* Lists in the order of last access time\*/

lib src games include libexec sbin share etc bin

[Student@localhost local]$ ls -i /\*display inode information\*/

2752577 bin 2752579 games 2752581 lib 2752583 sbin 2752607 src

2752578 etc 2752580 include 2752582 libexec 2752584 share

Each file has an inode and is identified by an inode number (i-number) in the file system where it resides. inodes provide important information on files such as user and group ownership, access mode (read, write, execute permissions) and type.

Combining ls options:

[LabExam@ISELAB1 local]$ ls -la

total 44

drwxr-xr-x. 11 root root 4096 Jul 7 2015 .

drwxr-xr-x. 14 root root 4096 Aug 4 2015 ..

drwxr-xr-x. 2 root root 4096 Oct 1 2009 bin

drwxr-xr-x. 2 root root 4096 Oct 1 2009 etc

drwxr-xr-x. 2 root root 4096 Oct 1 2009 games

drwxr-xr-x. 2 root root 4096 Oct 1 2009 include

[LabExam@ISELAB1 local]$ ls -l -a

total 44

drwxr-xr-x. 11 root root 4096 Jul 7 2015 .

drwxr-xr-x. 14 root root 4096 Aug 4 2015 ..

drwxr-xr-x. 2 root root 4096 Oct 1 2009 bin

drwxr-xr-x. 2 root root 4096 Oct 1 2009 etc

drwxr-xr-x. 2 root root 4096 Oct 1 2009 games

drwxr-xr-x. 5 root root 4096 Jul 7 2015 share

drwxr-xr-x. 2 root root 4096 Oct 1 2009 src

Echo command:

[Student@localhost local]$ echo haii /\*Echo is for printing\*/

Vi editor:

Output:

Haii

[LabExam@ISELAB1 ~]$ vi dg.sh /\*creating new file using vi editor\*/

Cat command:

1. Concatenation:

[LabExam@ISELAB1 ~]$ cat dg.sh td.sh /\* If there are two files, we can concatenate the contents of those two files by using cat command. \*/

ffhghngh

hjhjjj

fgfgff

hghgh

[LabExam@ISELAB1 ~]$ cat dg.sh

ffhghngh

hjhjjj

[LabExam@ISELAB1 ~]$ cat td.sh

fgfgff

hghgh

2. Creation of new file:

[LabExam@ISELAB1 ~]$ cat > l.txt /\* creating new file on the terminal page\*/

this is a beutiful world

3. Displaying contents of file:

[LabExam@ISELAB1 ~]$ cat l.txt /\*displays contents of a file\*/

this is a beutiful world

Options with cat command:

[LabExam@ISELAB1 ~]$ cat -n song.txt /\*numbering lines \*/

1 i am a student

2

3

[UNIXLAB@localhost~]$ cat -b t.txt

1 i am a student

[LabExam@ISELAB1 ~]$ cat -e test.sh /\*puts a $ at the end of each line\*/

hello everyone, how do you do?$

Hey, am fine.$

How's your training going on?$

Make Directory:(Mkdir)

[LabExam@ISELAB1 local]$ mkdir p

mkdir: cannot create directory `p': Permission denied

[LabExam@ISELAB1 local]$ cd ~

[LabExam@ISELAB1 ~]$ mkdir usp

[LabExam@ISELAB1 ~]$ cd usp /\*We were able to create new directory under home directory\*/

[LabExam@ISELAB1 usp]$ mkdir unix /\*making subdirectories under parent directory\*/

[LabExam@ISELAB1 unix]$

Move command:(mv)

[LabExam@ISELAB1 ~]$ mv c.txt l.txt /\* **mv** stands for **move**. mv is used to move one or more files or directories from one place to another in file system like UNIX. \*/

[LabExam@ISELAB1 ~]$ cat c.txt

bash: cd: usp: No such file or directory

[LabExam@ISELAB1 ~]$ cat l.txt

aaaaa

/\*moving contents from source file to destination file,source file gets removed permanently after moving\*/

Option with mv command :-i

[UNIXLAB@localhost~]$ mv -i ll3.sh l.sh /\* interactive - Attempt to move contents from source to destination file, but prompt before moving to confirm\*/

mv: overwrite `l.sh'? yes

[UNIXLAB@localhost~]$ cat ll3.sh

cat: ll3.sh: No such file or directory

[UNIXLAB@localhost~]$ cat l.sh

HHH

Moving group of files to a directory:

[LabExam@ISELAB1 ~]$mv t.txt n.txt y.txt usp /\*moving group of files to a directory\*/

[LabExam@ISELAB1 ~]$cd usp

[LabExam@ISELAB1 usp]$ls

t.txt n.txt y.txt

copy command:(cp)

[LabExam@ISELAB1 ~]$ cp b.txt c.txt /\* **cp** stands for **copy**. This command is used to copy files or group of files or directory. It creates an exact image of a file on a disk with different file name. cp command require at least two filenames in its arguments.\*/

[LabExam@ISELAB1 ~]$ cat b.txt

hello

[LabExam@ISELAB1 ~]$ cat c.txt

hello

Option with copy command:-i

[UNIXLAB@localhost sha]$ cp -i t1.txt t2.txt /\*/\* interactive - Attempt to copy contents from source to destination file, but prompt before copying to confirm\*/

cp: overwrite `t2.txt'? yes

[UNIXLAB@localhost sha]$ cat t1.txt

uuuuu

[UNIXLAB@localhost sha]$ cat t2.txt

uuuuu

Remove files : rm command

[LabExam@ISELAB1 ~]$ ls

add.sh mul.sh odd.sh sub.sh b.sh

[LabExam@ISELAB1 ~]$ rm b.txt /\* If you want to delete/remove any file, then using rm command\*/

[LabExam@ISELAB1 ~]$ cat b.txt

cat: b.txt: No such file or directory

[LabExam@ISELAB1 ~]$ ls

add.sh mul.sh odd.sh sub.sh

[UNIXLAB@localhost sha]$ ls

add.sh mul.sh odd.sh sub.sh

[UNIXLAB@localhost sha]$ rm \* /\*removing all files from current working directory\*/

[UNIXLAB@localhost sha]$ ls

[UNIXLAB@localhost ~]$ ls

21745.sh a.txt gokul new file p77.cv p.txt

aa.sh boi.py goutham oddeven.sh p.cpp Public

a.awk b.sh harshit1 p1.sh Pictures qq.sh

aayush b.sh~ hell p22.c pogram.sh sandesh

[UNIXLAB@localhost ~]$ rm -r \* /\*removing all files and directories from current working directory\*/

[UNIXLAB@localhost ~]$ ls

[LabExam@ISELAB1 ~]$ rm a.txt l.txt /\* Deleting/Removing multiple files using rm command\*/

[LabExam@ISELAB1 ~]$ cat a.txt

cat: a.txt: No such file or directory

[LabExam@ISELAB1 ~]$ cat l.txt

cat: l.txt: No such file or directory

rm command with option:

[LabExam@ISELAB1 ~]$rm -i t.txt /\*interactive - Attempt to remove file in the working directory, but prompt before each file to confirm\*/

rm: remove `t2.txt'? yes

[LabExam@ISELAB1 ~]$cat t.txt

cat: t.txt: No such file or directory

[LabExam@ISELAB1 ~]$ mkdir usp /\*create a directory\*/

Remove directory:(rmdir)

[LabExam@ISELAB1 ~]$ rmdir usp /\*Remove a directory\*/

[LabExam@ISELAB1 ~]$ cd usp

bash: cd: usp: No such file or directory

Date command:(date)

[UNIXLAB@localhost~]$ date /\*display date and time\*/

Thu Nov 22 15:11:53 IST 2018

Process status:(ps)

[UNIXLAB@localhost~]$ ps /\*display process details\*/

PID TTY TIME CMD

2354 pts/0 00:00:00 bash

2637 pts/0 00:00:00 cat

2640 pts/0 00:00:00 cat

10033 pts/0 00:00:00 cat

10034 pts/0 00:00:00 cat

10040 pts/0 00:00:00 ps

2. Simple shell script for basic arithmetic and logical calculations.

#!/bin/sh

a=2

b=4

c=3

val=`expr $a + $b`

echo "Sum = $val"

val=`expr $a - $b`

echo "Difference= $val"

val=`expr $a \\* $b`

echo "Product = $val"

val=`expr $b / $a`

echo "quotient = $val"

val=`expr $a % $b`

echo "Modulus = $val"

if [ $a == $b ]

then

echo “a is equal to b”

else

echo “a is not equal to b”

fi

if [ $a != $b ]

then

echo “a is equal to b”

else

echo “a is not equal to b”

fi

if [ $a -lt 10 -o $b -gt 20 ]

then

echo “true”

else

echo “false”

fi

if [ $a -lt 10 -a $b -gt 20 ]

then

echo “True”

else

echo “False”

fi

[LabExam@ISELAB1 ~]$chmod a+x pr2.sh /\*To execute your shell program, use chmod a+x program name(For changing mode), a+x means giving execute permission to all three category of people(user,group,others).\*/

[LabExam@ISELAB1 ~]$sh pr2.sh /\*To run the program use any of the methods: sh program name or ./ program name\*/

Sum = 8

Difference = -2

Product =8

Quotient =2

Modulus =2

a is not equal to b

a is not equal to b

True

False

3. Shell scripts to check various attributes of files and directories.

#!/bin/sh

file="filename.sh"

if [ -r $file ]

then

echo "File has read permission"

else

echo "File does not have read permission"

fi

if [ -w $file ]

then

echo "File has write permission"

else

echo "File does not have write permission"

fi

if [ -x $file ]

then

echo "File has execute permission"

else

echo "File does not have execute permission"

fi

if [ -f $file ]

then

echo "File is an ordinary file"

else echo "This is a special file"

fi

if [ -d $file ]

then

echo "File is a directory"

else

echo "File is not a directory"

fi

if [ -s $file ]

then

echo "File size is zero"

else

echo "File size is greater than zero"

fi

if [ -e $file ]

then

echo "File exists"

else

echo "File does not exist"

fi

if [ -b $file ]

then

echo "File is a block file "

else

echo "File is not a block file"

fi

if [ -c $file ]

then

echo "File is a character file "

else

echo "File is not a character file"

fi

output: [LabExam@ISELAB1 ~]$ sh pr3.sh

[LabExam@ISELAB1 ~]$chmod a+x pr3.sh

[LabExam@ISELAB1 ~]$sh pr3.sh

File has read permission

File has write permission

File has execute permission

File is an ordinary file

File size is greater than zero

File exists

File is not a character file is not a block file

4. Shell scripts to check and list attributes of processes.

!/bin/sh

ps

echo "ps : process status"

ps -f

echo "ps -f : full listing"

ps -l

echo "ps -l : along listing showing memory related information"

ps -u

echo "ps -u : process of user only"

ps -e

echo "ps -e : all process including user and system process"

ps -a

echo "ps -a : process of all user including processes not listed with terminal"

ps -t

echo "ps -t : processes running on terminal -l along listing showing memory related information"

output:

[LabExam@ISELAB1 ~]$ sh pr4.sh

[LabExam@ISELAB1 ~]$chmod a+x pr4.sh

[LabExam@ISELAB1 ~]$sh pr4.sh

PID TTY TIME CMD

2053 pts/0 00:00:00 bash

2422 pts/0 00:00:00 listing.sh

2423 pts/0 00:00:00 ps

ps = process status

UID PID PPID C STIME TTY TIME CMD

admin 2053 2043 0 10:46 pts/0 00:00:00 bash

admin 2422 2053 0 11:03 pts/0 00:00:00 /bin/sh ./listing.sh

admin 2424 2422 0 11:03 pts/0 00:00:00 ps -f

ps -f=full listing

F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD

0 S 1000 2053 2043 0 80 0 - 2306 wait pts/0 00:00:00 bash

0 S 1000 2422 2053 0 80 0 - 2045 wait pts/0 00:00:00 listing.sh

0 R 1000 2425 2422 0 80 0 - 2868 - pts/0 00:00:00 ps

ps -l=along listing showing memory related information

USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND

admin 1403 0.0 0.2 54036 9432 tty2 Ssl+ 10:46 0:00 /usr/libexec/gd

admin 1408 0.0 0.3 84184 12864 tty2 Sl+ 10:46 0:00 /usr/libexec/gn

admin 1510 3.3 3.8 1117800 135452 tty2 Sl+ 10:46 0:34 /usr/bin/gnome-

admin 1538 0.0 1.2 155672 44816 tty2 Sl+ 10:46 0:00 /usr/bin/Xwayla

admin 1571 0.3 0.2 52168 7748 tty2 Sl 10:46 0:03 ibus-daemon --x

ps -u=process of user only

PID TTY TIME CMD

1 ? 00:00:01 systemd

2 ? 00:00:00 kthreadd

3 ? 00:00:00 rcu\_gp

4 ? 00:00:00 rcu\_par\_gp

6 ? 00:00:00 kworker/0:0H-kb

ps -e=all process including user and system process

PID TTY TIME CMD

868 tty1 00:00:00 gnome-session-b

907 tty1 00:00:03 gnome-shell

1136 tty1 00:00:00 Xwayland

1163 tty1 00:00:00 ibus-daemon

1166 tty1 00:00:00 ibus-dconf

ps -a=process of all user including processes not listed with terminal

PID TTY STAT TIME COMMAND

2053 pts/0 Ss 0:00 bash

2422 pts/0 S+ 0:00 /bin/sh ./listing.sh

2429 pts/0 R+ 0:00 ps -t

ps -t=processes running on terminal -l along listing showing memoryb related information

5. Write awk script that uses all of its features.

#!/bin/awk -f

BEGIN {print “START”}

{print $1, “\t”, $3}

END {print “DONE”}

output:

[LabExam@ISELAB1 ~]$ awk -f pr5.awk a.txt

And am

You can

Me you

LabExam@ISELAB1 ~]$ cat a.txt

And is am I was

You they can take

Me as you print

6. Write a shell script to display list of users currently logged in.

#!/bin/sh

echo “User logged in:”

users

echo “Current logged in date and time :”

date

echo “Currently logged in users:”

who

echo “Currently logged in username:”

whoami

output:

[LabExam@ISELAB1 ~]$ sh pr6.sh

[LabExam@ISELAB1 ~]$chmod a+x pr6.sh

[LabExam@ISELAB1 ~]$sh pr6.sh

User logged in:

UNIX LAB UNIX LAB

Current logged in date and time:

Sat sep 29 12:25:30 IST 2018-11-22

Currently logged in users:

UNIXLAB TTY1 2018-09-29 11.08 (:0)

UNIXLAB pts/0 2018-09-29 12.03 (:0.0)

UNIXLAB pts/1 2018-09-29 12.28 (:0.0)

Currently logged in username:

UNIXLAB