Q 1.1

login as: SEP19\_MAA3

SEP19\_MAA3@10.51.103.212's password:

Last login: Wed Oct 10 12:10:03 2018 from 10.219.34.106

[SEP19\_MAA3@NDAUNIX ~]$

Q 1.2

[SEP19\_MAA3@NDAUNIX ~]$ exit

logout

Q 2.1

$ pwd

Q 2.2

$ pwd

Q 2.3

$ whoami

Q 2.4

$ ls -a

Q 2.5

$ ls ~

Q 2.6

$ ls -l

Q 2.7

$ ls | grep -x 'chap[0-9a-z]\{1,\}'

Q 2.8

$ mkdir ~/C\_prog

Q 2.9

$ mkdir ~/newdir newdirectory

$ ls ~

Q 2.10

$ ls -R

Q 2.11

$ rmdir ~/newdirectory

Q 2.12

$ mkdir ~/temp

Q 2.13

$ rmdir ~/newdir

Q 2.14

$ mkdir temp/directorynew

Q 2.15

$ cd ~

Q 2.16

$ cd temp/directorynew

Q 2.17

$ rmdir ~/C\_prog

Q 2.18

$ cd /etc

$ ls

Q 2.19

$ ls /usr/bin -d .\*

Q 2.20

$ vi first.unix

i

Hi! Good Morning everybody.

Welcome to the First exercise of UNIX.

Hope you enjoy doing the assignments.

Esc

:wq

Q 2.21

$ cp first.unix first.unics

Q 2.22

$ cat first.unix first.unics

Q 2.23

$ mkdir ~/temp/abc

Q 2.24

$ cp -r ~/\* ~/temp/abc

Q 2.25

$ cp ~/first.unix ~/temp/second.unix

Q 2.26

$ rmdir ~/first.unics

Q 2.27

$ cd ~/temp

$ rm \*.

No such file or directory

Q 2.28

$ mv ~/temp/\*{a,c,o} ~/

Q 2.29

$ cp ~/\*UNIX ~/temp

Q 2.30

$ rm -r ~/temp

Q 2.31

for cp-> last parameter should be a present directory

for mv-> last parameter should be a present directory

Q 2.32

$ cat > friends

Madhu 6966456 09/07/68

Jamil 2345215 08/09/67

Ajay 5546785 01/04/66

Mano 7820022 09/07/68

David 8281292 09/09/60

Simmi 7864563 12/12/70

Navin 2224311 30/05/68

Q 2.33

$ cat friends

Q 2.34

$ cat friends > newfriend

Q 2.35

$ cat friends newfriend

Q 2.36

$ who > users

Q 2.37

$ cat friends >> users

Q 2.38

$ date

Q 2.39

$ cal 8 1996

Q 2.40

$ date '+%'

%

$ date '+%m'

08

$ date '+%D'

08/29/18

$ date '+%/%Training Activity'

%/14:34:07raining Activity

$ date '+%Training Activity'

14:34:27raining Activity

$ date '+%r'

02:34:49 PM

Q 3.1

$ chmod +x chap1.txt

Q 3.2

$ chmod 775 add.c

$ chmod u+x,g+x,o+x c

Q 3.3

$ chmod -x add.c

$ chmod g+r,o+r add.c

Q 3.4

$ chmod +x a.c kk.c nato myfile

Q 3.5

$ cd /root

Q 4.1

$ help > lsdoc

Q 4.2

$ cat lsdoc | more -d

Q 4.3

$ head -n4 lsdoc

Q 4.4

$ tail -n7 lsdoc

Q 4.5

$ rm lsdoc

Q 4.6

Q 4.7

$ grep '^Ma' friends

Q 4.8

$ grep -E '[^Ma ]+[i id]$' friends

grep '^Ma.\*[id,i]\s.\*' friends

Q 4.9

$ ls -d $PWD/\*

Q 4.10

ls -d \*/

Q 4.11

$ ls -d $PWD/\* | grep chap

Q 4.12

$ sort -k1 friends

Q 4.13

$ cat friends | tr '[:lower:]' '[:upper:]'

Q 4.14

$ ls > dir

Q 4.15

$ awk {'print $1,$9'} dir

Q 4.16

$ awk {'print $9'} dir > files

Q 4.17

$ awk {'print $1'} dir > perms

Q 4.18

$ awk {'print $5'} dir > sizes

Q 4.19

$ cat files sizes perms

Q 4.20

$ who

Q 4.21

$ ls -lSr | head -n 2 | grep -v 'total'

Q 4.22

$ cat friends | wc -l

Q 4.23

1)

$ cat emp.lst | wc -L

$ cat dept.lst | wc -L

$ cat desiq.lst | wc -L

2)$ awk '{print $9,$11;}' emp.lst | tail -1

3)$ $ awk '{print $3,$5;}' emp.lst > cfile1

4)$ awk '{print $1,$7,$9,$11;}' emp.lst > cfile2

5)

6)$ sort -r -k3 emp.lst

7)$ sort -k11 emp.lst > srtf

8)$ sort -k 5,5 -k 3,3 emp.lst

9)$ sort -k9 emp.lst

10)$ awk '{print $5;}' emp.lst | uniq

11)$ awk '{print $5;}' emp.lst | uniq

12)$ awk '{print $5;}' emp.lst | uniq | wc -l

13)$ awk -F'/' '{print $3;}' emp.lst | awk '{print $1}' | uniq -c

14)$ nl -v 100 -i5 emp.lst

Q 4.24

Q 4.25

$ date '+%H:%M:%S %Z %A %d %B %G'

Q 4.26

$ ls -l | sort -k2 -r

Q 4.27

$ ls -1 | sort

Q 4.28

$ ls -l | awk '{print $9,$5;}' | sort -nk2

Q 4.29

$ ls -t1 | head -n 1

**Lab 5:**

Q 5.1

a)

:%s/Netware/Novell Netware

b)

Go to insert mode

move to word Shared resource using w command

go to insert mode and type the text

c)

Place cursor at end of file and press a in esc mode.

Q 5.2

vi emp.txt go to insert mode and type data

1 | Ram | analyst | BI | 06/08/96 | 240000

2 | Shyam | analyst | BI | 23/03/96 | 240000

3 | John | associate | BI | 01/04/95 | 240000

:wq

**Lab 6:**

Q 6.1

$ cat > Employee.dat

James 76382 PACE Chennai

John 34228 GRIT Hyderabad

Peter 22321 GE Bangalore

Albert 32342 GRIT Pune

Mathew 23222 PACE Mumbai

Richard 23232 ACS Pune

a)

$ sed -n '2'p Employee.dat

$ sed -n '/[pune]$/p' Employee.dat

b)

$ sed -n 1,5p Employee.dat

c)

$ sed -i 's/chennai/pune/g' Employee.dat

d)

$ sed -i -e 's/x/UNIX/g' Employee.dat

e)

$ sed '/^$/d' Employee.dat

f)$ sed -i Employee.dat -re '3,5d'

$ cat Employee.dat

Q 6.2

$ sed -n '/PACE/p' Employee.dat > Pace.dat

$ cat Pace.dat

**Lab 7:**

Q 7.1

$ ps -u BI\_USER21

Q 7.2

$ ps -e

Q 7.3

$ pidstat -p 20991

cat > lab8.txt

E001 Ram Unix 30

E002 Shyam DSA 20

E003 John Unix 30

E004 Tommy DSA 20

E005 Sanjay Unix 30

E006 Atul DSA 10

E007 Vishal Unix 30

E008 Philip DSA 20

E009 Rishab Unix 30

E010 Rohit DSA 10

**Lab 8:**

Q 8.1

$ awk '{print $0} NR==5{exit}' OFS=' ' lab8.txt

Q 8.2

$ awk '{if($3=="Unix") print $0}' lab8.txt

Q 8.3

$ awk '{if($3=="DSA" && $4<30) print $0}' lab8.txt

Q 8.4

$ awk '{if(NR%2==0) print $0}' lab8.txt

Q 8.5

$ awk '{if(substr($2,1,1)=="N") print $0}' lab8.txt

**Lab 9:**

1.

echo $PS1

echo $PS2

PS1='Ram'

echo $PS1

2.

echo $SHELL

3)printenv SHELL

4)continent=”Africa”

[BI\_USER21@NDAUNIX /]$ echo “$continent”

“”Africa””

5)$continent=”Africa”

export continent

$echo “$continent”

Africa

$sh

$echo “$continent”

Africa

$continent=”Asia”

$echo “$continent”

Asia

ctrl + d

$echo “$continent”

Africa

6.

read name

status=0

for u in $(who | awk '{print $1}' | sort | uniq)

do

if [ "$u" = "$name" ];

then

status=1

fi

done

if [ "$status" = "1" ];

then

echo "$name is Logged in"

else

echo "Not Logged In"

fi

7.

for f in \*;

do

echo $f

c=`cat $f`

echo $c

done

8.

if [ -e $1 ];

then

p=`ls -l $1|awk '{print $1}'`

echo $p

else

echo "FILE doesn't exists"

fi

9.

if [ $1 -gt $2 ];

then

if [ $1 -gt $3 ]

then

echo $1

else

echo $3

fi

else

if [ $2 -gt $3 ]

then

echo $2

else

echo $3

fi

fi

10.

echo "Enter Pattern:"

read p

echo "Enter file:"

read f

if [[ -f $f ]]

then

c=`grep "$p" "$f"|wc -l`

if [ $c -gt 0 ]

then

echo "Pattern Found"

else

echo "NOT FOUND"

fi

else

echo "FILE NOT FOUND"

fi

11.

create () {

echo "Enter file name: "

read f

if [ -e $f ]

then

echo "FILE EXISTS"

else

touch $f

echo "File Created"

fi

}

crd () {

echo "Enter dir name:"

read d

if [ -e $d ]

then

echo "Dir EXISTS"

else

mkdir $d

echo "DIR CREATED"

fi

}

copy () {

echo "Enter First file:"

read f1

if [ -e $f1 ]

then

echo "Enter 2nd file name: "

read f2

cat $f1 > $f2

else

echo "File doesnt exist"

fi

}

move () {

echo "Enter First file:"

echo "Enter First file:"

read f1

if [ -e $f1 ]

then

echo "Enter 2nd file name: "

read f2

mv $f1 $f2

else

echo "File doesnt exist"

fi

}

echo "1. create file"

echo "2. create dir"

echo "3. copy file"

echo "4. move file"

read op

if [ $op = 1 ]

then

create

elif [ $op = 2 ]

then

crd

elif [ $op = 3 ]

then

copy

elif [ $op = 4 ]

then

move

fi

12.

yesno () {

echo $1

echo "(y/n)"

read a

if [ "$a" = "y" ]

then

r=0

else

r=1

fi

return $r

}

echo "Enter Filename: "

read f

if [[ -d $f ]]

then

yesno "Do you want to delete the directory??"

if [[ $? = 0 ]]

then

rm -r $f

elif [[ $? = 1 ]]

then

echo " OK Thanks "

fi

elif [[ -f $f ]]

then

yesno "Do you want to delete the file??"

if [[ $? = 0 ]]

then

rm $f

elif [[ $? = 1 ]]

then

echo " OK Thanks "

fi

else

echo "FILE DOES NOT EXISTS"

fi

13.

userf () {

for u in $(who | awk '{print $1}' | sort | uniq)

do

if [ "$u" = "$1" ]

then

return 0

fi

done

return 1

}

echo "ENTER 4 NAMES: "

read w

read x

read y

read z

userf $w

if [ $? = 0 ]

then

echo "User $w is IN"

else

echo "USER IS OUT"

fi

userf $x

if [ $? = 0 ]

then

echo "User $x is IN"

else

echo "USER IS OUT"

fi

userf $y

if [ $? = 0 ]

then

echo "User $y is IN"

else

echo "USER IS OUT"

fi

userf $z

if [ $? = 0 ]

then

echo "User $z is IN"

else

echo "USER IS OUT"

fi

14.

echo "ENTER 1st NAME"

read f

echo "ENTER 2nd NAME"

read l

echo "Hello, $f $l"

15.

echo `find -mtime -1 -ls`

16.ls -l| grep '^-'| awk '{print $9}'

17)ls -l $1

ls -l $2

18.

echo "Name is"

echo `basename $0`

echo "PID is: "

echo $$

19.

cat $2 > $1

cat $3 >> $1

20.

echo "a. Display calendar for current month"

echo "b. Search Pattern"

echo "c. Count directories"

read op

if [ "$op" == "a" ]

then

cal

elif [ "$op" == "b" ]

then

elif [ "$op" == "c" ]

then

else

echo "WRONG OPTION!!!!"

fi

21.

echo "Enter the date"

read d

da=`echo $d|cut -c 1-2`

m=`echo $d|cut -c 3-4`

y=`echo $d|cut -c 5-8`

d=`printf "%s/%s/%s" {$y,$m,$da}`

day=`date -d $d|cut -d ' ' -f 1`

if [ "$day" == "Mon" ]

then

echo "Monday Blues"

elif [ "$day" == "Fri" ]

then

echo "yeh!! Its's Weekend"

else

echo "Yeaaahhhh!! It's $day"

fi

22

for f in \*.lst

do

c=`cat $f`

printf "%s: %s \n" {"$f","$c"}

done

23.

if [ "$3" = "+" ]

then

echo `echo "$1+$2"|bc`

elif [ "$3" = "-" ]

then

echo `echo "$1-$2"|bc`

elif [ "$3" = "\*" ]

then

echo `echo $1\*$2|bc`

elif [ "$3" = "/" ]

then

echo `echo "$1/$2"|bc`

else

echo "WRONG OPTIONS"

fi

24.

awk '{if($3 < 50) print $0}' student > fail

awk '{if($3 >= 50) print $0}' student > pass

echo "pass Students"

echo `wc -l pass`

echo "Failed Students"

echo `wc -l fail`

25.

25)day=$1

lines=$(cat emp.txt | wc -l)

#echo "$lines"

day=$(date -d $day +%s)

for i in {1..6}

#i=1

#while [ $i -le $lines ]

do

d=$(sed "${i}q;d" emp.txt | cut -d "|" -f 5)

#echo $d

n=$(sed "${i}q;d" emp.txt | cut -d "|" -f 2)

#echo $n

bday=$(date -d $d +%s)

#echo "$bday $day"

if [ $day -lt $bday ];

then

echo "$n"

fi

done