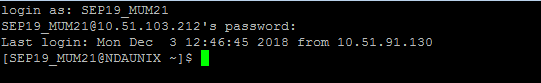
1. Connecting to the Unix Server

|  |  |
| --- | --- |
| **Goals** | * Learn to connect to the Unix server * Learn to log out of the Unix server |
| **Time** | 5 min |

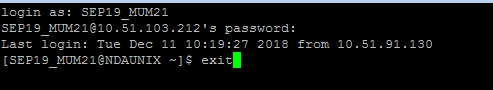
## 1.1: Connecting to the Unix Server

**Step 1:** Enter your login name and password to login to the UNIX system.



## 1.2: Logging out of the system

**Step 1**: Type the exit command at $ prompt or else, press ctrl and d together to log out.



1. Unix Basic Command

|  |  |
| --- | --- |
| **Goals** | * Learn to use basic Unix commands |
| **Time**  **Lab Setup** | 100 min  Telnet with Unix Server |

## 2:1 Executing basic commands:

1. To display the current working directory, the command is:

**pwd**

The output is as follows.

/home/SEP19\_MUM21

Ans: pwd

3

1. Display the path to and name of your HOME directory.

**Echo $HOME**

The output is as follows.

/home/SEP19\_MUM21

4

1. Display the login name using which you have logged into the system

Ans: whoami

5

1. Display the hidden files of your current directory.

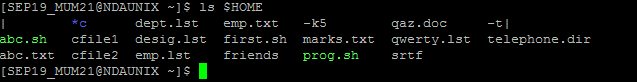
Ans: ls -a

Since there are no hidden files no file name is displayed

6

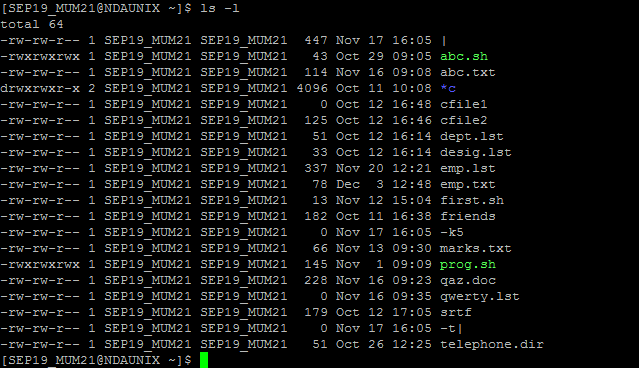
1. List the names of all the files in your home directory.

Ans: ls $HOME



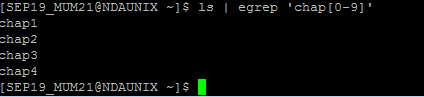
1. Using the long listing format to display the files in your directory.

Ans: ls -l



1. List the files beginning with chap followed by any number or any lower case alphabet. (Example, it should display all files whose names are like chap1, chap2, chap3 ……., chapa,ahapb,chapc,……..)

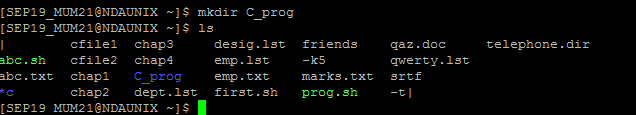
Ans: ls | egrep ‘chap[0-9] | chap[a-z]’



1. Give appropriate command to create a directory called C\_prog under your home directory. (Note: Check the directory using ls)

Mkdir C\_prog

ls



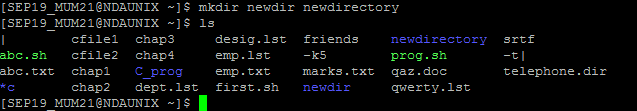
1. Create the following directories under your home directory. (Note: Check using ls)

newdir

newdirectory

Ans: mkdir newdir newdirectory

ls



1. List the names of all the files, including the contents of the sub directories under your home directory.

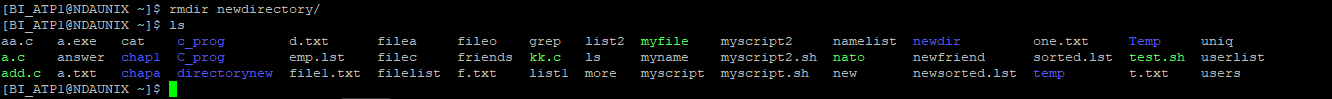
Ans: ls -LR



1. Remove the directory called newdirectory from your working directory.

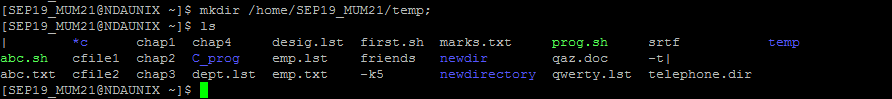
Ans: rmdir newdirectory

ls



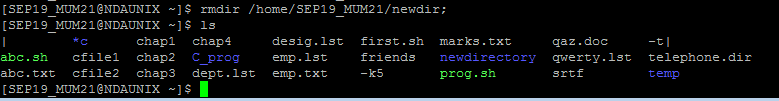
1. Create a directory called temp under your home directory.

Ans: mkdir /home/SEP19\_MUM21/temp;



1. Remove the directory called newdir under your home directory and verify the above with the help of the directory listing command.

Ans: rmdir /home/SEP19\_MUM21/newdir;

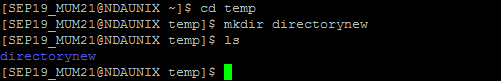


1. Create another directory directorynew under the temp directory.

Ans: cd temp

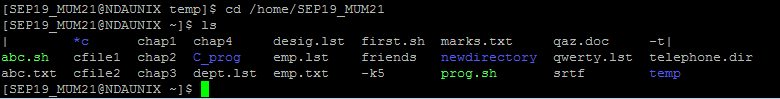
mkdir directorynew

ls



1. Change the directory to your home directory.

Cd /home/SEP19\_MUM21



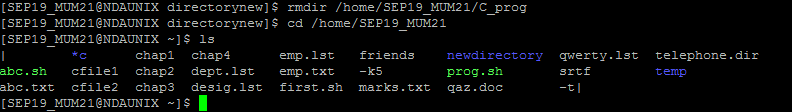
1. From your home directory, change the directory to directorynew using relative and absolute path.

cd /home/SEP19\_MUM21/temp/directorynew

17

1. Remove the directory called c\_prog, which is in your home directory.

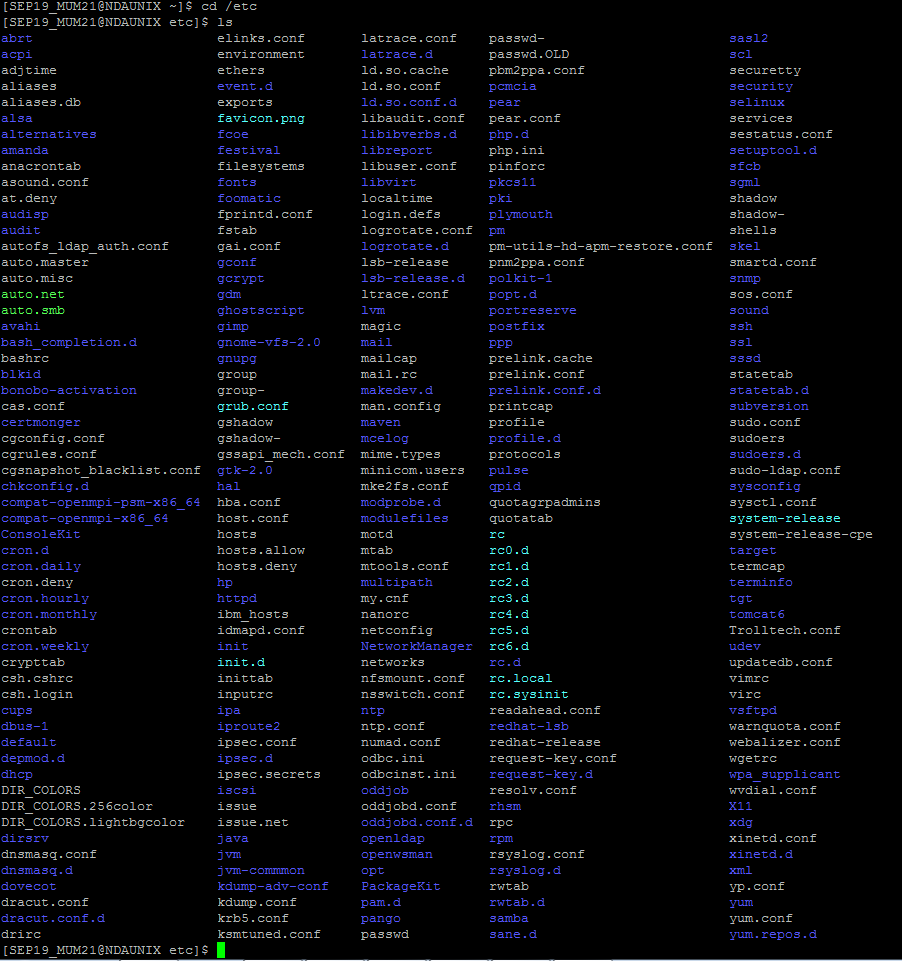
rmdir /home/SEP19\_MUM21/c\_prog



1. Change to the directory /etc and display the files present in it.

Ans: cd /etc

ls



1. List the names of all the files that begin with a dot in the /usr/bin directory.

Ans: cd user/bin

ls -F

1. Create a file first.unix with the following contents.

Hi! Good Morning everybody.

Welcome to the First exercise on UNIX.

Hope you enjoy doing the assignments.

Ans:

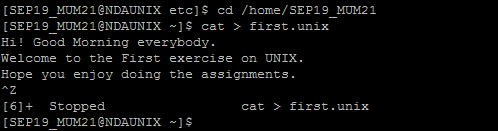
[SEP19\_MUM21@NDAUNIX etc]$ cd /home/SEP19\_MUM21

[SEP19\_MUM21@NDAUNIX ~]$ cat > first.unix

Hi! Good Morning everybody.

Welcome to the First exercise on UNIX.

Hope you enjoy doing the assignments.



1. Copy the file first.unix in your home directory to first.unics.

(Note: checked using ls, first.unix file also should exist along with first.unics)

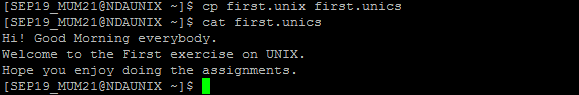
Ans: SEP19\_MUM21@NDAUNIX ~]$ cp first\_unix first\_unics

[SEP19\_MUM21@NDAUNIX ~]$ cat first\_unics

Hi! Good Morning everybody.

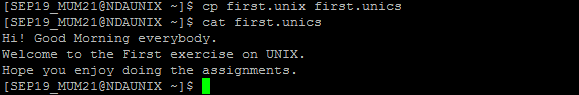
Welcome to the First Exercise on UNIX

Hope you enjoy doing the assignments



1. List the contents of first.unix and first.unics with a single command.

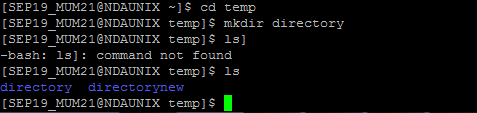
[SEP19\_MUM21@NDAUNIX ~]$ cat first\_unix first\_unics



1. Create a new directory under the temp directory.

Ans: [SEP19\_MUM21@NDAUNIX ~]$ cd temp

[SEP19\_MUM21@NDAUNIX temp]$ mkdir directory



1. From your home directory, copy all the files to the directory created under the temp sub directory.

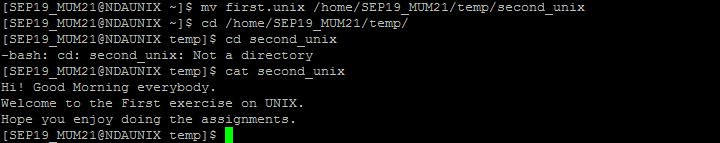
Ans: [SEP19\_MUM21@NDAUNIX ~]$ cp /home/SEP19\_MUM21/\* /home/SEP19\_MUM21/temp/directory/

1. Move the file first.unix to the directory temp as second.unix

Ans: [SEP19\_MUM21@NDAUNIX ~]$ mv first\_unix /home/SEP19\_MUM21/temp/second\_unix

[SEP19\_MUM21@NDAUNIX ~]$ cd temp

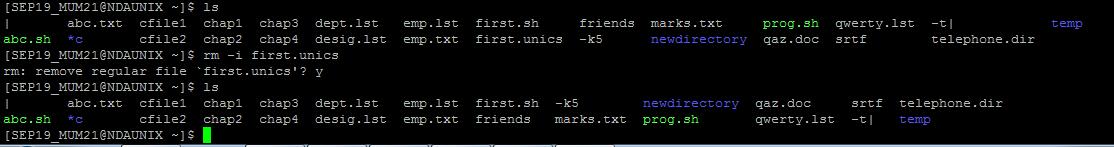
[SEP19\_MUM21@NDAUNIX temp]$ ls



1. Remove the file called first.unics from the home directory.

Ans: [SEP19\_MUM21@NDAUNIX ~]$ rm -i first\_unics

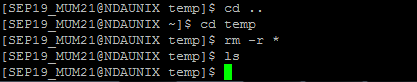
rm: remove regular file `first\_unics'? y



1. Change your directory to temp and issue the command rm \*. What do you observe?

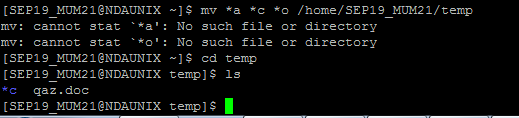
Ans: cd temp

[SEP19\_MUM21@NDAUNIX temp]$ rm –r \*



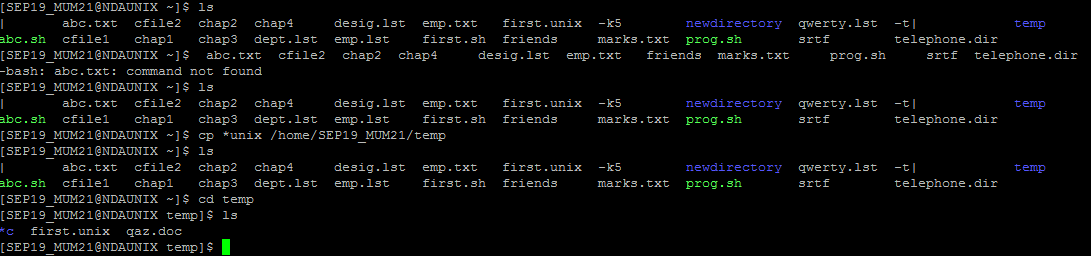
1. Move all files whose names end with a, c and o to the HOME directory.

Ans: [SEP19\_MUM21@NDAUNIX temp]$ mv \*a \*c \*o /home/SEP19\_MUM21



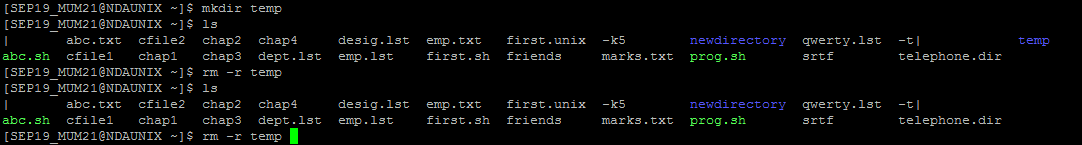
1. Copy all files that end with a ‘UNIX’ to the temp directory.

Ans: [SEP19\_MUM21@NDAUNIX ~]$ cp \*unix /home/SEP19\_MUM21/temp



1. Issuing a single command, remove all the files from the directory temp and the directory itself.

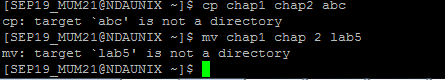
Ans: [SEP19\_MUM21@NDAUNIX ~]$ rm -r temp



1. Try commands cp and mv with invalid number of arguments and note the results.

Ans: [SEP19\_MUM21@NDAUNIX ~]$ cp chap1 chap2 abc

[SEP19\_MUM21@NDAUNIX ~]$ mv chap1 chap 2 lab5



1. Use the cat command to create a file friends, with the following data:

Madhu 6966456 09/07/68

Jamil 2345215 08/09/67

Ajay 5546785 01/04/66

Mano 7820022 09/07/68

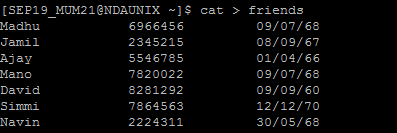
## David 8281292 09/09/60

## Simmi 7864563 12/12/70

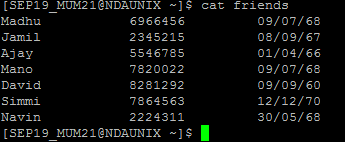
## Navin 2224311 30/05/68

The fields should be separated by a tab.

Ans: [SEP19\_MUM21@NDAUNIX ~]$ cat > friends

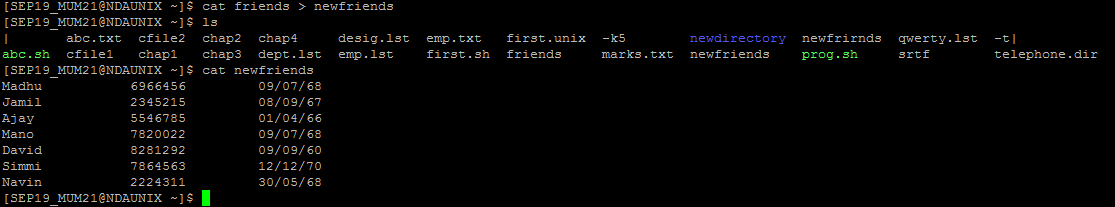
1. Display contents of the file friends.

Ans: [SEP19\_MUM21@NDAUNIX ~]$ cat friends



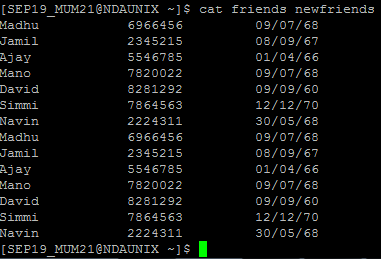
1. Copy contents of friends to newfriend without using the cp command.

Ans: [SEP19\_MUM21@NDAUNIX ~]$ cat friends > newfrirnds



1. Display contents of the file friends and newfriends in a single command.

Ans: [SEP19\_MUM21@NDAUNIX ~]$ cat friends newfriends

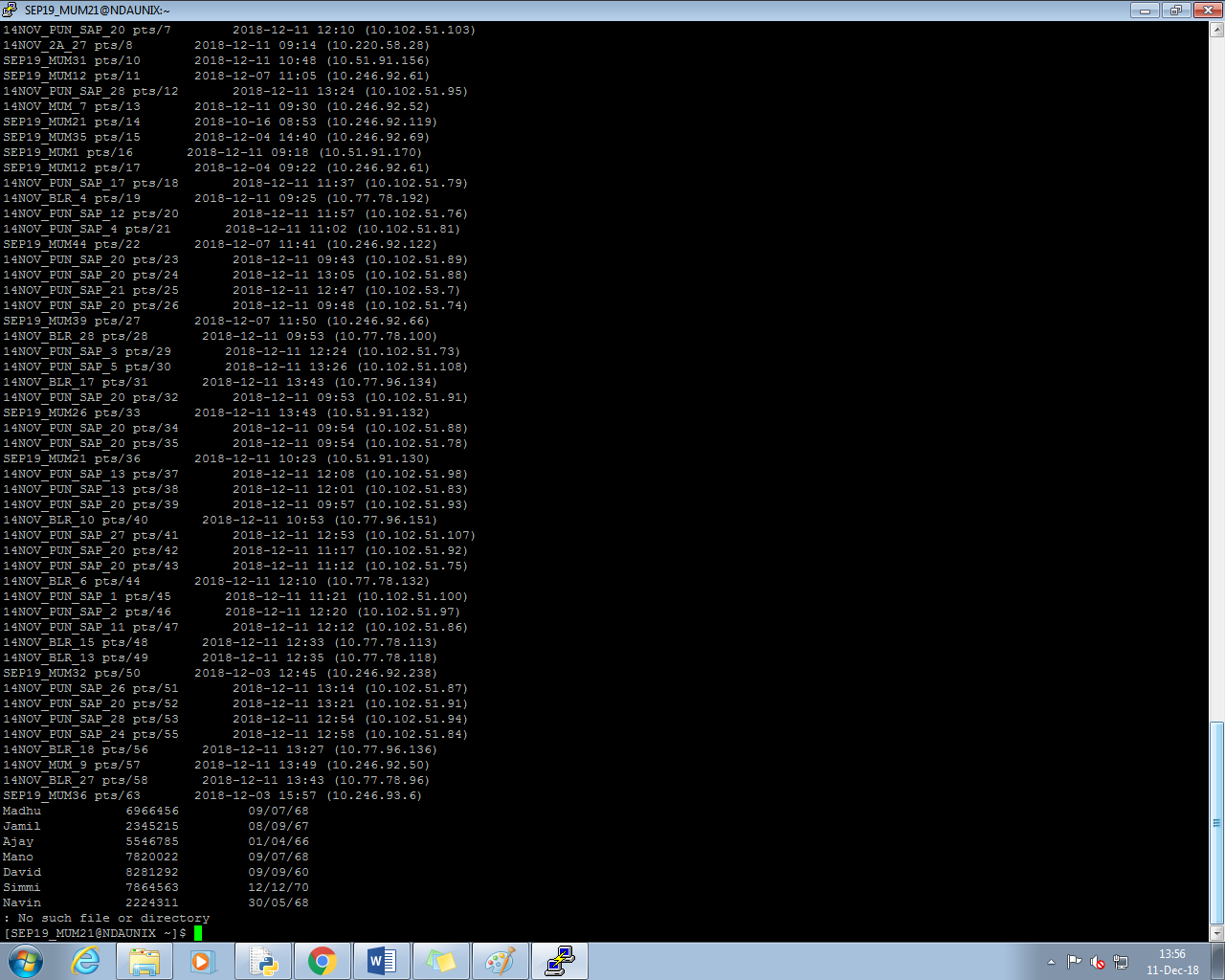


1. Find all users currently working on the system and store the output in a file named as users.

Ans: [SEP19\_MUM21@NDAUNIX ~]$ who|cat > users

1. Append contents of friends file to the file, users.

Ans: [SEP19\_MUM21@NDAUNIX ~]$ cat friends.lst > file | cat users



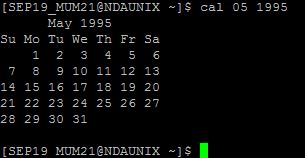
1. Display current system date and time and record your observations. How is the time displayed?

Ans: [SEP19\_MUM21@NDAUNIX ~]$ date

34

1. Display calendar for the month and year of your birth.

Ans: [SEP19\_MUM21@NDAUNIX ~]$ cal 12 1996



1. Try following commands and record your observations.

date “+ %”

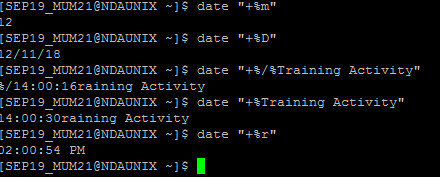
date “+%m”

date “+%D”

date “+%/%Training Activity”

date “+%Training Activity”

date “+%r”



1. UNIX File System & Permissions

|  |  |
| --- | --- |
| **Goals** | * Learn to grant and to remove permissions and to view the file system |
| **Time**  **Lab Setup** | 15 min  Telnet with Unix Server |

## 3.1: Viewing the File System and Granting/Removing Permissions

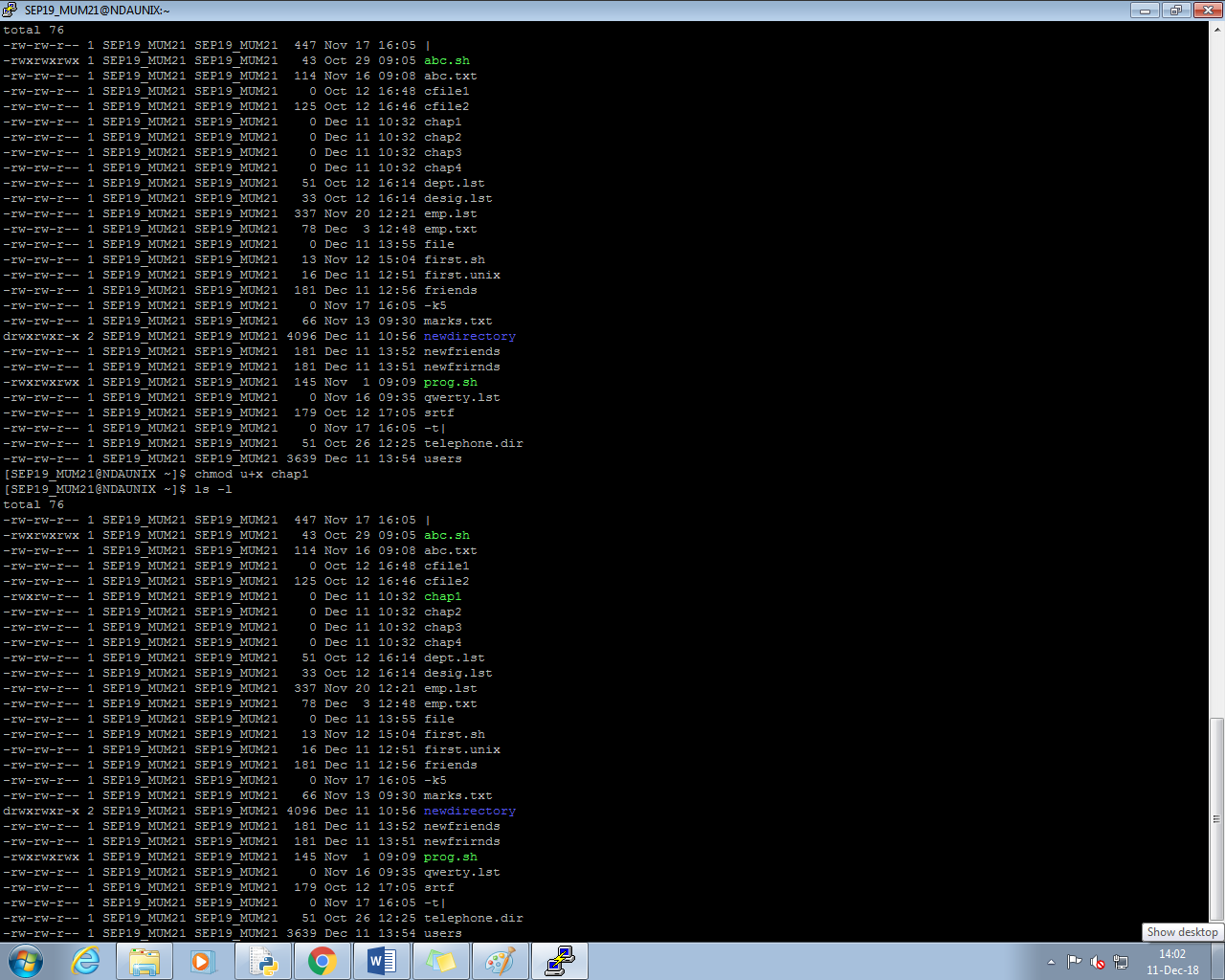
## (Note: Create required files if doesn’t exists.)

1. Give the execute permission for the user for a file chap1

Ans:

[SEP19\_MUM21@NDAUNIX ~]$ chmod u+x chap1

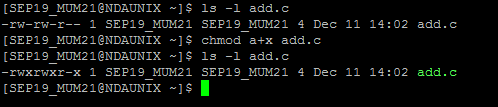
[SEP19\_MUM21@NDAUNIX ~]$ ls -l chap1



1. Give the execute permission for user, group and others for a file add.c

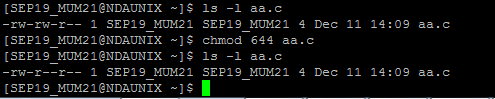
Ans: [SEP19\_MUM21@NDAUNIX ~]$ chmod a+x add.c

[SEP19\_MUM21@NDAUNIX ~]$ ls -l add.c



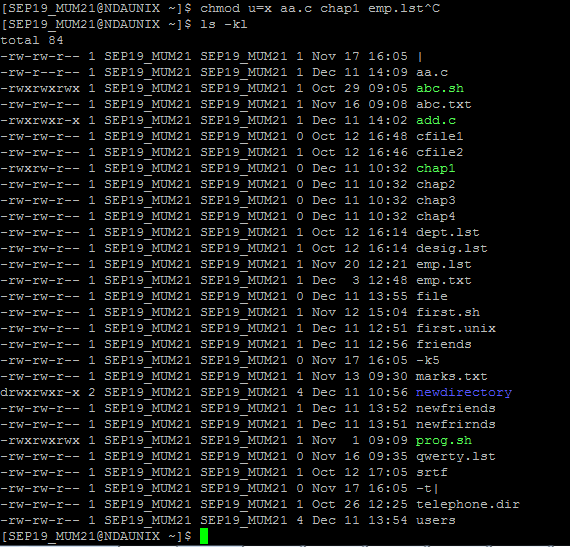
1. Remove the execute permission from user, give read permission to group and others for a file aa.c

Ans: [SEP19\_MUM21@NDAUNIX ~]$ chmod 644 aa.c



1. Give execute permission for users for a.c, kk.c, nato and myfile using single command

Ans: [SEP19\_MUM21@NDAUNIX ~]$ chmod u=x a.c kk.c nato myfile



1. Change the directory to root directory. Check the system directories, like bin, etc, usr etc  
   Ans: cd /

ls –l/ bin

ls –l /etc

ls –l /usr

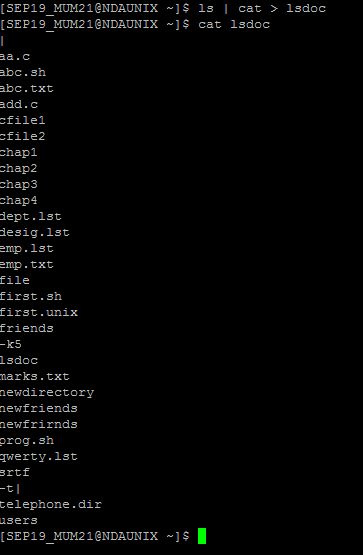
1. Simple and Advance Filetrs

|  |  |
| --- | --- |
| **Goals** | * Learn to use Pipes & Filters in UNIX |
| **Time**  **Lab Setup** | 100 min  Telnet with Unix Server |

## 4.1: Using Pipes and Filters:

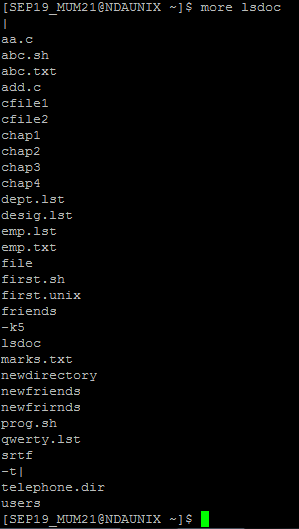
1: Redirect the content of the help document ls, into a file called as lsdoc.

Ans: [SEP19\_MUM21@NDAUNIX ~]$ ls | cat > lsdoc



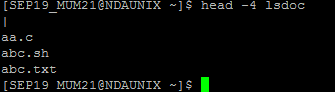
2: Display the content of the lsdoc page wise.

Ans. [SEP19\_MUM21@NDAUNIX ~]$ more lsdoc



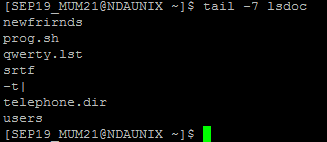
3: Display only the first 4 lines of the lsdoc file.

Ans. [SEP19\_MUM21@NDAUNIX ~]$ head -4 lsdoc



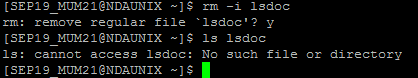
4: Display only the last 7 lines of the file lsdoc.

Ans. [SEP19\_MUM21@NDAUNIX ~]$ tail -7 lsdoc



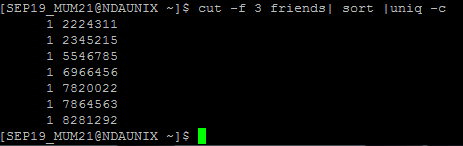
5: Remove the file lsdoc.

Ans. [SEP19\_MUM21@NDAUNIX ~]$ rm -i lsdoc



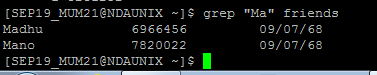
6: There will be B’day celebration from the friends file, find how many B’day parties will be held. If two of the friends have the B’date on the same day, then we will be having one party on that day.

Ans: [SEP19\_MUM21@NDAUNIX ~]$ cut -f 3 friends| sort |uniq –c



7: Display the lines starting with Ma, in the file friends.

Ans: [SEP19\_MUM21@NDAUNIX ~]$ grep "Ma" friends



8: Display the lines starting with Ma, ending with i or ending with id, in the file friends.

Ans: [SEP19\_MUM21@NDAUNIX ~]$ grep -E '^Ma.i|^Ma.id' friends

9: Print all the files and the directory files from the current directory across all the sub directories, along with its path

Ans: [SEP19\_MUM21@NDAUNIX ~]$ find $(pwd)



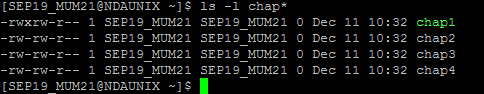
10: Print only the Directory files.

Ans: [SEP19\_MUM21@NDAUNIX ~]$ ls -d \*/

48

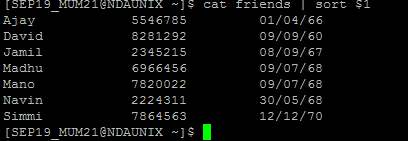
11: Display the files starting with chap, along with its path.

Ans: [SEP19\_MUM21@NDAUNIX ~]$ ls -l chap\*



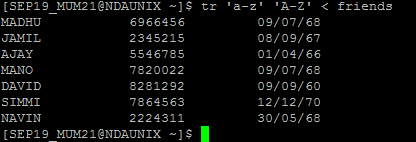
12: Sort the file friends in ascending order of names.

Ans: [SEP19\_MUM21@NDAUNIX ~]$ cat friends | sort $1

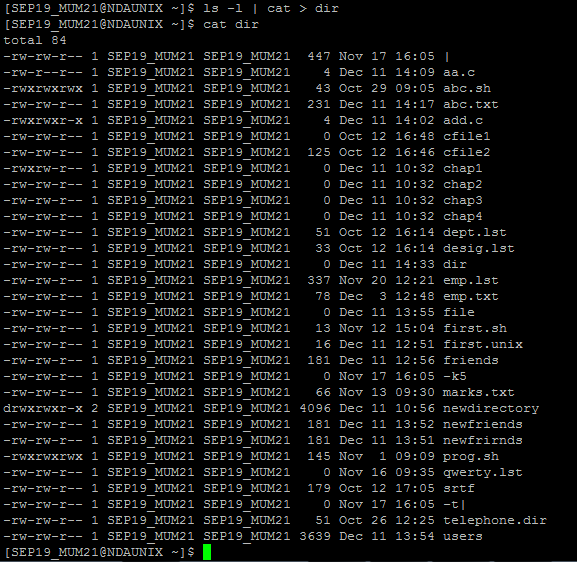


13: Display the contents of the file friends in uppercase letters.

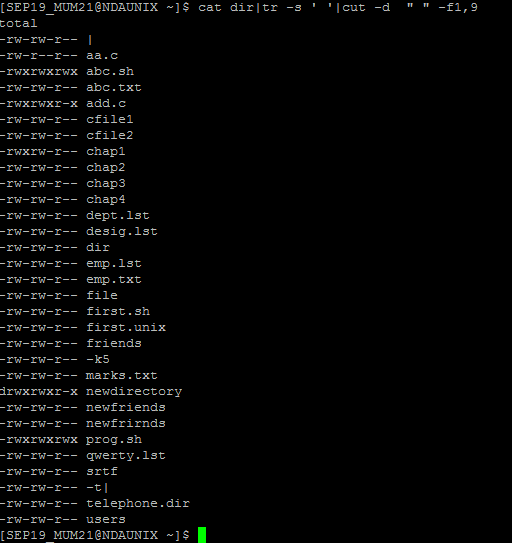
Ans: [SEP19\_MUM21@NDAUNIX ~]$ tr 'a-z' 'A-Z' < friends



14: Store the contents of your home directory in a file called dir.

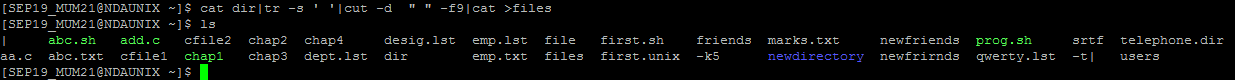


15: From the above file dir, display the file permissions and the name of the file only.

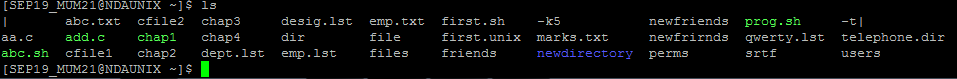


16: From the same dir file, store only the file names in a file called files.

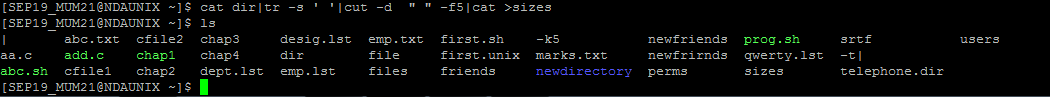
cat dir|tr -s ' '|cut -d  " " -f9|cat >files



17: From the same dir file, store only the permissions of files in a file called perms.



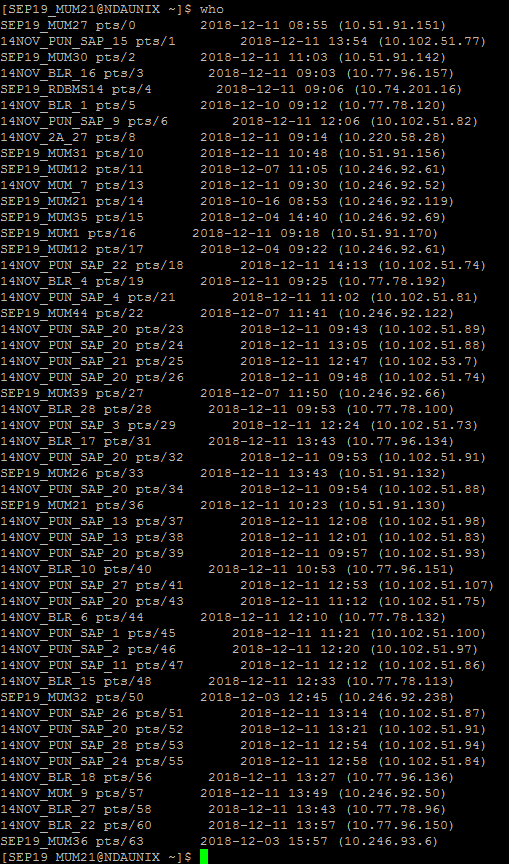
18: From the same dir file, store only the file sizes in a file called sizes.



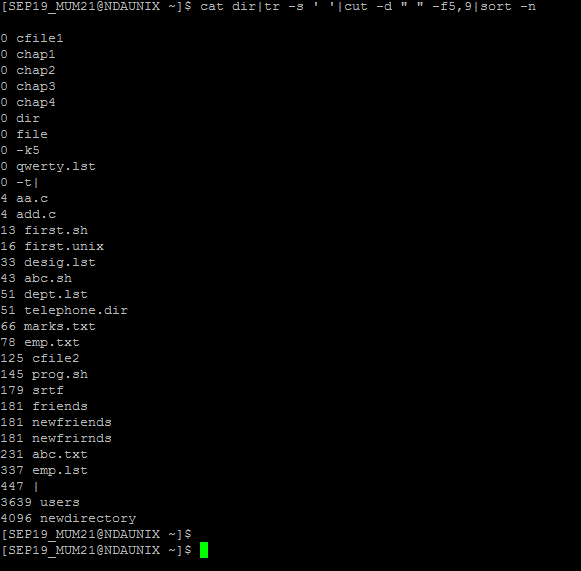
19: Display the file names, sizes and permissions from your directory in that order.

Ans.cat files sizes perms

20: Display the number of users working on the system.



21: Find out the smallest file in your directory.



22: Display the total number of lines present in the file friends.

59

23:Create the following fixed record format files (with “|” delimiter between fields) with the structure given below, and populate them with relevant data use these files to solve following questions

emp.lst: Empid(4),Name(18),Designation(9),Dept(10),Date of Birth(8),Salary(5)

dept.lst : Dept.Code(2),Name(10),Head of Dept’s id(4)

desig.lst: Designation Abbr.(2), Name (9)

1. Find the record lengths of each file.

wc -l emp.lst

output: 6 emp.lst

wc -l dept.lst

output: 4 design.lst

1. Display only the date of birth and salary of the last employee record.

cat emp.lst|cut -d '|' -f5,6

1. Extract only employee names and designations. (Use column specifications). Save output as cfile1.

cat emp.lst|cut -d '|' -f2,3|cat >cfile1

1. Extract Emp.id, dept, dob and salary. (Use field specifications). Save output as cfile2.

cat emp.lst|cut -d '|' -f1,2,4,6|cat >cfile2

1. Fix the files cfile1 and cfile2 laterally, along with the delimiter.
2. Sort the emp.lst file in reverse order of Emp. Names.

sort -r -k2 emp.lst

1. Sort the emp.lst file on the salary field, and store the result in file srtf.

sort -k6 emp.lst >srtf

1. Sort the emp.lst file on designation followed by name.

sort -k2 -k3 emp.lst

1. Sort the emp.lst file on the year of birth.

sort -t"|" -k5.7 emp.lst

1. Find out the various designations in the employee file. Eliminate duplicate listing of designations.

cut -d "|" -f3 emp.lst >temp |uniq temp

1. Find the non-repeated designation in the employee file.

cut -d "|" -f3 emp.lst >temp |uniq temp

1. Find the number of employees with various designations in the employee file.

cut -d "|" -f3 emp.lst |uniq -c

1. Create a listing of the years in which employees were born in, along with number of employees born in that year.

cut -d "|" -c70-75 emp.lst |uniq -c

1. Use nl command to create a code table for designations to include designation code (Start with dept. code 100, and subsequently 105, 110 …).

cut -d"|" -f4,3 emp.lst|n1

24: PCS has its offices at Pune, TTC and Mumbai. The employees’ data is stored separately for each office. Create appropriate files (with same record structure as in previous assignment) and populate with relevant data.

1. List details about an employee ‘Manu Sharma’ in the Mumbai office.

grep "Manu Sharma" emp\_pcs\_pune.dat

1. List only the Emp.Id. And Dept. of Manu Sharma.

grep "Manu Sharma" emp\_pcs\_pune.dat |cut

1. List details of all managers in all offices. (O/P should not contain file names.).

grep "Manager"emp\_pcs\_mumbai.dat emp\_pcs\_ttc.dat emp\_pcs\_pune.dat

1. Find the number of S.E. in each office.

grep "SSE"emp\_pcs\_mumbai.dat emp\_pcs\_ttc.dat emp\_pcs\_pune.dat -c

1. List only the Line Numbers and Employee names of employees in ‘H/W’ in Pune file.

grep "SSE" emp\_pcs\_pune.dat |n1 | cut -d "," -f1,2

1. Obtain a listing of all employees other than those in ‘HR’ in the Mumbai file and save contents in a file ‘nonhr’.

grep -v 'HR' emp\_pcs\_pune.dat

1. Find the name and designation of the youngest person who is not a manager.

grep -v 'HR' emp\_pcs\_pune.dat emp\_pcs\_mumbai.dat emp\_pcs.ttc.dat| sort -k5 |head -1

1. Display only the filename(s) in which details of employee by the name ‘Seema Sharma’ can be found.

grep 'Seema Sharma' emp\_pcs\_pune.dat emp\_pcs\_mumbai.dat emp\_pcs.ttc.dat

1. Locate the lines containing saxena and saksena in the Mumbai office.

fgrep check emp\_pcs\_mumbai.dat |n1|cut -f1

1. Find the number of managers who earn between 50000 and 99999 in the Pune office.
2. List names of employees whose id is in the range 2000 – 2999: in Pune Office; in all offices.
3. Locate people having same month of birth as current month in Pune office.

egrep '[1-30]/11/[1900-2999]' emp\_pcs\_pune.dat

1. List details of all employees other than those of HR and Admin in file F1.

grep -v 'HR' 'Admin' emp\_pcs\_pune.dat

1. Locate for all Dwivedi, Trivedi, Chaturvedi in Pune file.

egrep title emp\_pcs\_pune.dat

1. Obtain a list of people in HR, Admin and Recr. depts. sorted in reverse order of the dept.

**Stretched assignments:**

25: Write a command sequence that prints out date information in this order: time, day of week, day number, month, year:

13:44:42 IST Sun 16 Sept 1994

echo `date '+%H:%M:%S %Z %a %d %b %Y '`

26: Write a command sequence that prints the names of the files in the current directory in the descending order of number of links

ls|sort -d

27: Write a command sequence that prints only names of files in current working directory in alphabetical order

ls|sort -d

28: Write a command sequence to print names and sizes of all the files in current working directory in order of size

ls -l|tr -s ' '|cut -d ' ' -f5,9|sort -n

29: Determine the latest file updated by the user

ls -l|tr -s ' '|cut -d ' ' -f6,7,8,9|sort

1. Vi Editor

|  |  |
| --- | --- |
| **Goals** | Work with Vi Editor in Unix |
| **Time**  **Lab Setup** | 15 min  Telnet with Unix Server |

## 5.1: Working wth Vi Editor

1. Create a file using Vi. Enter the following text:

*A network is a group of computers that can communicate with each other, share resources, and access remote hosts or other networks. Netware is a computer network operating system designed to connect, manage, and maintain a network and its services. Some of the network services are Netware Directory Services (NDS), file system, printing and security.*

1. Change the word “Netware” in the second line to “Novell Netware”.

Press Esc and go to the line where we need to insertNovell and press "i"

1. Insert the text “(such as hard disks and printers)” after “share resources” in the first line.

Press Esc and go to the line where we need to add content (such as hard disks and printers) and press "i"

1. Append the following text to the file:

“Managing NDS is a fundamental administrator role because NDS provides a single point for accessing and managing most network resources.”

Ans: a) Press Esc and go to the line where we need to insert Novell and press "i"

b) Press Esc and go to the line where we need to add content (such as hard disks and printers) and press "i"

c) Go to end of the line by pressing Esc and $ and then press "a" to append the line

“Managing NDS is a fundamental administrator role because NDS provides a single point for accessing and managing most network resources.”

Go to end of the line by pressing Esc and $ and then press "a" to append the line

2: Create the data files, used in the previous lab sessions using vi editor.

1. Shell Script

|  |  |
| --- | --- |
| **Goals** | Learn to write simple shell scripts |
| **Time**  **Lab Setup** | 150 min  Telnet with Unix Server |

## 6.1: Writing Shell-Scripts

1. Display the Primary and Secondary prompt. Change the primary prompt to your name: temporarily

$ PS2="prompt\_name $"

$echo "This is the example of

prompt\_name $ secondary prompt"

This is the example of secondary promp

2: As soon as you login, the prompt should be changed to your name: also the name of the home directory should be automatically displayed.

$PS1="[\u@\h\w\$HOME]\$"

3: Check the content of the Environmental variable SHELL.

$ printenv SHELL

4: Try the below exercise and check the output.

**Note**: Type every line and press enter, do not type the entire code in a vi editor.

$continent=”Africa”

$echo “$continent”

------------🡪 Africa

$sh

$echo “$continent”

------------🡪 No Response

$continent=”Asia”

$echo “$continent”

------------🡪 Asia

$ctrl + d

$echo “$continent”

------------🡪 Africa

$sh

$echo “$continent”

------------🡪 No Response

$ctrl + d

5: Try the below exercise and check the output. (Export variables)

**Note**: Type every line and press enter, do not type the entire code in a vi editor.

$continent=”Africa”

export continent

$echo “$continent”

------------🡪 Africa

$sh

$echo “$continent”

------------🡪 Africa

$continent=”Asia”

$echo “$continent”

------------🡪 Asia

$ctrl + d

$echo “$continent”

------------🡪 Africa

6: Write a shell script that takes the user name as input and reports whether he / she has logged in or not.

Logincheck.sh

-read

- w |grep $a

7: Write a shell script to display the file name and its contents of all the files that is there in the current directory.

8: Write a shell script, which will take a file name as argument and check whether the file exists and display its access permissions for user.

File.sh

-read filename

-if test -f $filename

- then

- echo "File Exist" `ls -l $filename`

else

echo "File does not Exist"

fi

9: Pass three numbers as command line arguments and display the largest number in the given three numbers.

10: Write a shell script which will accept a pattern and a file name. The pattern will be searched in the file provided. Display appropriate messages and perform necessary validations on file.

validation.sh

pattern(){

echo "Enter the filename"

read filename

echo "Enter the pattern"

read pattern

if test -f $filename

then

grep $pattern $filename > out.txt

if [[ -s out.txt ]]

then

cat out.txt

return 0

else

return 1

fi

fi

}

if pattern

then

echo "success"

else

echo "Failed"

fi

11: To create a menu program for a) creating a file, b) Creating a directory, c) copying a file, d) moving a file. (use functions)

* 1. If the file exists already give the appropriate message
  2. If the dir exists already give the appropriate error message
  3. Source file should exist if not give a message, It should have read permission if not another message, Destination file either there or not, if not there then create it and copy it. If there, then ask whether to overwrite or not, if yes then overwrite it or else give a message file exists already and not overwritten.

create\_file(){

echo "Enter file name"

read file

if [ ! -f $file ];

then

touch $file

echo "Successfully created file $file"

else

echo "File already exists! Do you want to override the file?(Y/N)"

read answer

if [ $answer = "Y" -o $answer = "y" ];

then

touch $file

echo "Successfully created file $file"

else

echo "Not creating file $file...";

fi

fi

}

create\_dir(){

echo "Enter directory name"

read dir

if [ ! -d $dir ];

then

mkdir $dir

echo "Successfully created directory $dir"

else

echo "Directory already exists!"

fi

}

copy\_file(){

echo "Enter source file name"

read source

echo "Enter destination file name"

read destination

if [ -f $destination ];

then

echo "Files exists at destination! Want to override(Y/N)"

read answer

if [ $answer = "Y" -o $answer = "y" ];

then

cp $source $destination

echo "Successfully copied $source to $destination"

else

echo "Not copying file...";

fi

else

cp $source $destination

echo "Successfully copied $source to $destination"

fi

}

move\_file(){

echo "Enter source file name"

read source

echo "Enter destination file name"

read destination

if [ -f $source ];

then

if [ -f $destination ];

then

echo "Files exists at destination! Want to override(Y/N)"

read answer

if [ $answer = "Y" -o $answer = "y" ];

then

mv $source $destination

echo "Successfully moved $source to $destination"

else

echo "Not moving file...";

fi

else

mv $source $destination

echo "Successfully moved $source to $destination"

fi

else

echo "Source file does not exists"

fi

}

echo "Menu"

echo "1. Create a file"

echo "2. Create a directory"

echo "3. Copy file"

echo "4. Move file"

echo "5. Exit"

echo " enter your choice"

read choice

case $choice in

1)create\_file;;

2)create\_dir;;

3)copy\_file;;

4)move\_file;;

5)exit;;

esac

12: Write a function yesno() to display question to user and accept answer as y/n. If answer to the question is y the function should return 0 otherwise 1.

Use yesno functions for asking different questions. Question will be passed as parameter to the function.

Accept filename from user check whether it is file or directory. Use yesno() function to display question do you really want to delete file? If the ans is y, then delete the file or directory.

Yes/No Function

echo "choose the command you want to run 1.copy 2.Move 3.Delete

read choice

case $choice in

1) g= "Do you want to copy the file to another file?";;

2) g= "Do you want to move the file?";;

3)g= "Do you want to delete the file?";;

\*)echo wrong choice;

esac

yesno()

echo "${a}"

echo "y/n"

read option

case "$option" in

y)

if [ $choice -eq 1]

then

echo "Enter the filename"

read filename

if test -f $filename

then

echo "Enter the destination"

read destin

cp -v $filename $destin

else

echo "Filename invalid"

fi

elif [ $choice -eq 2 ]

then

echo "Enter the filename"

read filename

if test -f $filename

then

echo "Enter the new file name"

read destin

mv -v $filename $destin

else

echo "invalid filename"

fi

else

echo "Enter the filename"

read filename

if test -f $filename

then

rm -v $filename

else

echo "invalid file"

fi

fi;;

n)

exit;;

\*) echo "invalid";;

esac

}

yesno $g $choice

13: Write a shell script to store names of four employees and check whether those employees are currently logged in or not. Display appropriate message.

14: Accept the user's first and last name and the echo the entire name along with some suitable comment.

15: List all files that have been modified today.

16: Display long listing of only the regular files in the current directory.

17: Display details of all files in the 2 “paths” accepted from user. The display should be screen by screen.

18: Let the script display its name and its PID.

19: Get the concatenated o/p of 2 files into a third file: Take 3 command line arguments: The first argument is the name of a destination file, and the other two arguments are names of files whose contents are to be placed in the destination file.

**Stretched Assignments:**

20: Write a menu driven shell program to:

a. Display calendar of current month

1. Search for a pattern in all the files/subdirectories from current directory.
2. Count the no. of directories / sub directories in current directory

21: Display day of week for a given date. (ddmmyyyy)

If day is Monday, display message “Monday Blues”

Friday display message “yeh! It’s week end.”

Similarly display different messages for each day of the week.

22: Display the contents of all .lst files in the current directory.

23: Design a simple calculator, which will add/subtract/multiply/divide 2 numbers.

eg. cal 10 20 + will give o/p as 30.

24: For a student file with the following fields, rollno, name, marks, Generate 2 files ‘Pass’ and ‘Fail’ containing records of student who have passed or failed. Also count the number of students who have passed or failed.

25: Accept a date string from terminal and display employees born after the input date.