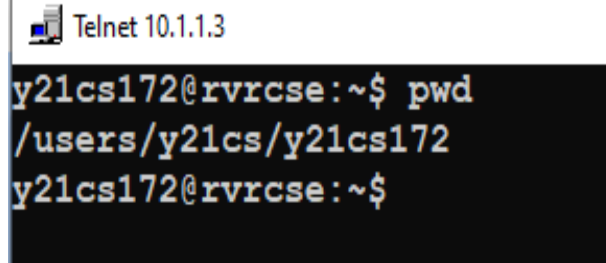


SKILL ORIENTED COURSE - 2
CSSL2 – LINUX PROGRAMMING

MODULE – 1

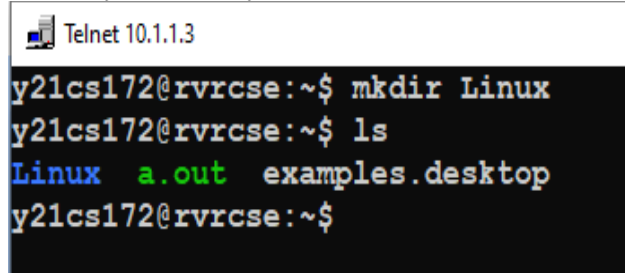
DIRECTORY RELATED UTILITIES

1. **pwd:** It is used to print the full path of the present working directory starting from the root.



```
Telnet 10.1.1.3
y21cs172@rvrcse:~$ pwd
/users/y21cs/y21cs172
y21cs172@rvrcse:~$
```

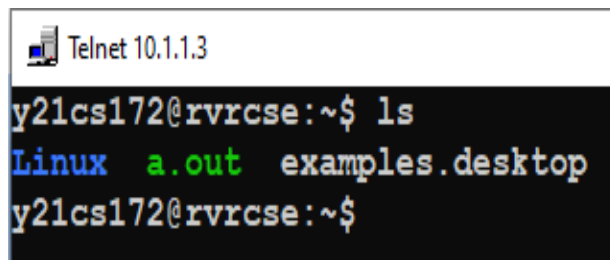
2. **mkdir:** It is used to create directory(ies), if they do not exist.



```
Telnet 10.1.1.3
y21cs172@rvrcse:~$ mkdir Linux
y21cs172@rvrcse:~$ ls
Linux a.out examples.desktop
y21cs172@rvrcse:~$
```

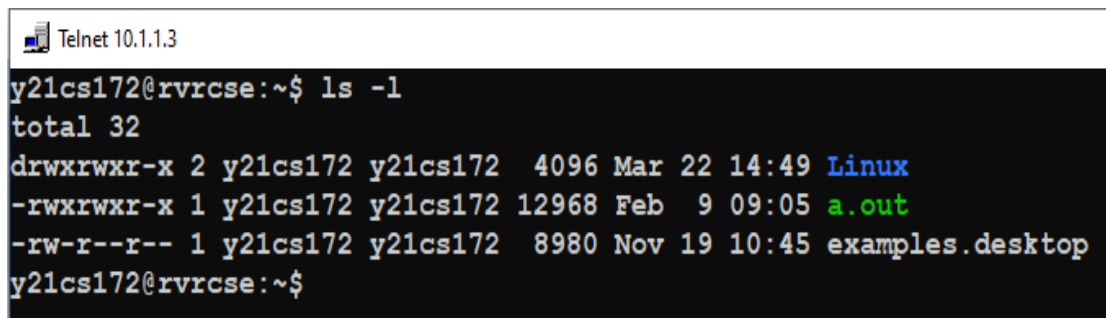
3. **ls:** It is used to list information about the files of the current directory by default (or) a specified directory.

- a. This command lists only the viewable contents of the directory.



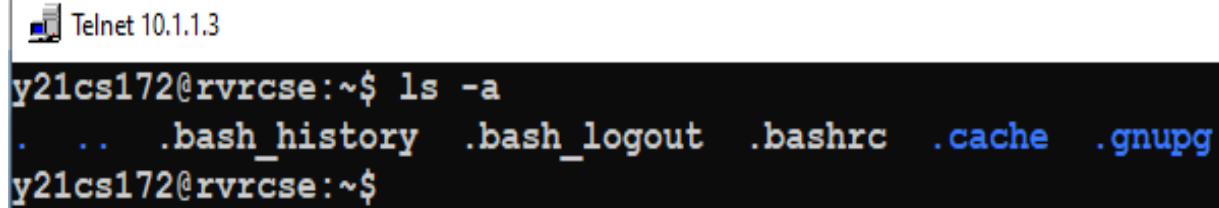
```
Telnet 10.1.1.3
y21cs172@rvrcse:~$ ls
Linux a.out examples.desktop
y21cs172@rvrcse:~$
```

- b. This command is used to print a larger list with all the permissions, users and owner names.



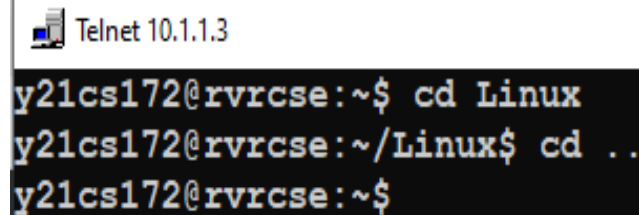
```
Telnet 10.1.1.3
y21cs172@rvrcse:~$ ls -l
total 32
drwxrwxr-x 2 y21cs172 y21cs172 4096 Mar 22 14:49 Linux
-rwxrwxr-x 1 y21cs172 y21cs172 12968 Feb 9 09:05 a.out
-rw-r--r-- 1 y21cs172 y21cs172 8980 Nov 19 10:45 examples.desktop
y21cs172@rvrcse:~$
```

- c. This command is used to print the hidden files that start with '.' in the directory.



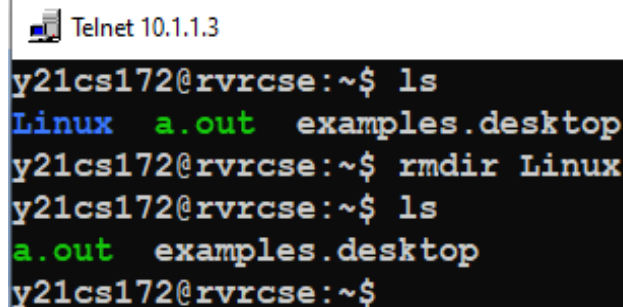
```
Telnet 10.1.1.3  
y21cs172@rvrcse:~$ ls -a  
.  ..  .bash_history  .bash_logout  .bashrc  .cache  .gnupg  
y21cs172@rvrcse:~$
```

4. **cd:** It is used to change the directory to the home directory by default (or) to a specified directory.



```
Telnet 10.1.1.3  
y21cs172@rvrcse:~$ cd Linux  
y21cs172@rvrcse:~/Linux$ cd ..  
y21cs172@rvrcse:~$
```

5. **rmdir:** It is used to remove directory(ies) only if they are empty.



```
Telnet 10.1.1.3  
y21cs172@rvrcse:~$ ls  
Linux  a.out  examples.desktop  
y21cs172@rvrcse:~$ rmdir Linux  
y21cs172@rvrcse:~$ ls  
a.out  examples.desktop  
y21cs172@rvrcse:~$
```

MODULE – 2

FILE HANDLING AND TEXT PROCESSING

1. **cat:** It is used to concatenate files and print on the standard output.

a. **cat > 1.txt:** Creates a new file and write content to it.

```
Telnet 10.1.1.3
y21cs172@rvcse:~/Linux$ ls
y21cs172@rvcse:~/Linux$ cat >1.txt
Linux Programming
^Z
[9]+  Stopped                  cat > 1.txt
y21cs172@rvcse:~/Linux$ ls
1.txt
y21cs172@rvcse:~/Linux$ cat 1.txt
Linux Programming
y21cs172@rvcse:~/Linux$
```

b. **cat 1.txt:** Prints the content of the file on standard output.

c. **cat >> 1.txt:** Used to append extra content to the file.

```
Telnet 10.1.1.3
y21cs172@rvcse:~/Linux$ cat 1.txt
Linux Programming
y21cs172@rvcse:~/Linux$ cat >>1.txt
Unix Programming
^Z
[10]+  Stopped                 cat >> 1.txt
y21cs172@rvcse:~/Linux$ cat 1.txt
Linux Programming
Unix Programming
y21cs172@rvcse:~/Linux$
```

2. **cp:** It is used to copy files or directories from a source to a destination in the file system.

```
Telnet 10.1.1.3
y21cs172@rvcse:~/Linux$ ls
1.txt  head1.txt  more.txt  new.txt  permissions
y21cs172@rvcse:~/Linux$ cat 1.txt
Linux Programming
y21cs172@rvcse:~/Linux$ cp 1.txt 2.txt
y21cs172@rvcse:~/Linux$ ls
1.txt  2.txt  head1.txt  more.txt  new.txt
y21cs172@rvcse:~/Linux$ cat 2.txt
Linux Programming
y21cs172@rvcse:~/Linux$
```

3. **mv:** It is used to move a file from one location to another or to rename a file.

```
Telnet 10.1.1.3
y21cs172@rvcse:~/Linux$ ls
1.txt  head1.txt  more.txt  new.txt  permissions
y21cs172@rvcse:~/Linux$ mv 1.txt 2.txt
y21cs172@rvcse:~/Linux$ ls
2.txt  head1.txt  more.txt  new.txt  permissions
y21cs172@rvcse:~/Linux$
```

4. **rm:** It is used to remove specified files or directories.

```
Telnet 10.1.1.3
y21cs172@rvcse:~/Linux$ ls
1.txt  2.txt
y21cs172@rvcse:~/Linux$ rm 1.txt
y21cs172@rvcse:~/Linux$ ls
2.txt
y21cs172@rvcse:~/Linux$
```

5. **sort:** It is used to sort the content of a given file and prints it on the standard output.

- a. Sorting content based on numbers.

```
Telnet 10.1.1.3
y21cs172@rvcse:~/Linux$ cat >sort1.txt
999
777
111
555
444
^Z
[11]+  Stopped                  cat > sort1.txt
y21cs172@rvcse:~/Linux$ cat sort1.txt
999
777
111
555
444
y21cs172@rvcse:~/Linux$ sort sort1.txt
111
444
555
777
999
y21cs172@rvcse:~/Linux$
```

- b. Sorting content based on characters.

```
Telnet 10.1.1.3
y21cs172@rvcse:~/Linux$ cat 1.txt
tt
aa
yy
yy
aa
bb
y21cs172@rvcse:~/Linux$ sort 1.txt
aa
aa
bb
tt
yy
yy
y21cs172@rvcse:~/Linux$
```

6. **ln -s:** It is used to create symbolic links between two files.

```
Telnet 10.1.1.3
y21cs172@rvcse:~/Linux$ ls
1.txt
y21cs172@rvcse:~/Linux$ cat 1.txt
Linux Programming
y21cs172@rvcse:~/Linux$ ln -s 1.txt 2.txt
y21cs172@rvcse:~/Linux$ ls
1.txt  2.txt
y21cs172@rvcse:~/Linux$ cat 2.txt
Linux Programming
y21cs172@rvcse:~/Linux$ ls -l
total 4
-rw-rw-r-- 1 y21cs172 y21cs172 18 Mar 23 14:27 1.txt
lrwxrwxrwx 1 y21cs172 y21cs172 5 Mar 23 14:30 2.txt -> 1.txt
y21cs172@rvcse:~/Linux$
```

7. **unlink:** It is used to remove the symbolic links between files.

```
y21cs172@rvrcse:~/Linux$ unlink 2.txt
y21cs172@rvrcse:~/Linux$ ls -l
total 44
-rw-rw-r-- 1 y21cs172 y21cs172 18 Mar 24 15:46 1.txt
-rw-rw-r-- 1 y21cs172 y21cs172 40 Mar 24 16:03 head1.txt
-rw-rw-r-- 1 y21cs172 y21cs172 51 Mar 24 16:30 more.txt
-rw-rw-r-- 1 y21cs172 y21cs172 533 Mar 24 16:31 new.txt
```

8. **head:** It is used to print the specified number of lines of content of a file from the starting.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux$ cat head1.txt
aaa
bbb
ccc
ddd
eee
fff
ggg
hhh
iii
jjj
y21cs172@rvrcse:~/Linux$ head -1 head1.txt
aaa
y21cs172@rvrcse:~/Linux$
```

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux$ head -3 head1.txt
aaa
bbb
ccc
y21cs172@rvrcse:~/Linux$
```

9. **tail:** It is used to print the specified number of lines of content of a file from the bottom.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux$ cat head1.txt
aaa
bbb
ccc
ddd
eee
fff
ggg
hhh
iii
jjj
y21cs172@rvrcse:~/Linux$ tail -1 head1.txt
jjj
y21cs172@rvrcse:~/Linux$
```

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux$ tail -5 head1.txt
fff
ggg
hhh
iii
jjj
y21cs172@rvrcse:~/Linux$
```

10. find: It is used to find files or directories in a specified directory and of a specified name expression.

```
Telnet 10.1.1.3
y21cs172@rvcse:~$ find ./Linux -name sed*
./Linux/sed3.txt
./Linux/sed11.txt
./Linux/sed2.txt
./Linux/sed1.txt
./Linux/sed111.txt
y21cs172@rvcse:~$
```

11. more: It is used for paging through text one screen full at a time.

```
Telnet 10.1.1.3
y21cs172@rvcse:~/Linux$ more more.txt
hi
hello
how
are
you

hope
you
are
doing
fine
what
--More-- (54%)

multi
national
company
ok
let's
--More-- (89%)
```

12. w: It is used to display information about the users currently on the machine, and their processes.

```
Telnet 10.1.1.3
y21cs172@rvcse:~/Linux$ w
16:00:25 up 17 days, 7:33, 100 users, load average: 9.15, 9.41, 9.76
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU WHAT
y22cd52   pts/1    10.6.2.4        15:45    9:05   0.18s  0.11s vi lab9.c
y21cs118  pts/3    10.1.5.99       14:38    1:00s  0.35s  0.28s vi lab3_2.c
y22cd49   pts/4    10.6.2.26       15:53    3:13   0.08s  0.06s -bash
y22cd46   pts/6    10.6.2.64       14:45   43:45   0.12s  0.01s vi lab13.c
```

13. nl: It is used to write each file to standard output with line numbers added.

```
Telnet 10.1.1.3
y21cs172@rvcse:~/Linux$ nl head1.txt
 1 aaa
 2 bbb
 3 ccc
 4 ddd
 5 eee
 6 fff
 7 ggg
 8 hhh
 9 iii
10 jjj
y21cs172@rvcse:~/Linux$
```

14. **grep:** It searches for a pattern in the given files.

- a. **-wn** options are used to restricts matching to whole words only with line numbers.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux$ grep -wn the sample.txt
3:To see a man's true face, look to the photos he hasn't posted.
y21cs172@rvrcse:~/Linux$
```

- b. To display only those lines in a file that don't match, use the **-v** option.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux$ grep -wnv the sample.txt
1:My IQ test results came back. They were negative.
2:Why can't you trust an atom? Because they make up literally everything.
3:Why was six afraid of seven? Because seven eight nine.
4:What do you call a hippie's wife? Mississippi.
5:What do you call a bear with no teeth? A gummy bear.
6:What do fish say when they hit a concrete wall? Dam!
7:The claustrophobic astronaut? He just wanted a little more space.
8:If athletes get athlete's foot, what do astronauts get? Missile toe.
10:"Buffet" is a French word that means "get up and get it yourself."
y21cs172@rvrcse:~/Linux$
```

15. **egrep:** It is used to search for extended regular expression patterns.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux$ egrep -E "s.*w" sample.txt
My IQ test results came back. They were negative.
What do you call a hippie's wife? Mississippi.
What do fish say when they hit a concrete wall? Dam!
The claustrophobic astronaut? He just wanted a little more space.
If athletes get athlete's foot, what do astronauts get? Missile toe.
"Buffet" is a French word that means "get up and get it yourself."
y21cs172@rvrcse:~/Linux$
```

16. **fgrep:** It is used to search for fixed strings in the specified files.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux$ fgrep -F -n "ll" sample.txt
2:Why can't you trust an atom? Because they make up literally everything.
4:What do you call a hippie's wife? Mississippi.
5:What do you call a bear with no teeth? A gummy bear.
6:What do fish say when they hit a concrete wall? Dam!
y21cs172@rvrcse:~/Linux$
```

17. **uniq:** The uniq utility displays a file with all of its identical adjacent lines replaced by a single occurrence of the repeated line.

- a. filter out duplicate adjacent lines.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux$ uniq animals
cat snake
monkey snake
dolphin elephant
goat elephant
pig pig
monkey pig
y21cs172@rvrcse:~/Linux$
```

- b. display a count with the lines.

```
y21cs172@rvrcse:~/Linux$ uniq -c animals
1 cat snake
1 monkey snake
2 dolphin elephant
1 goat elephant
2 pig pig
1 monkey pig
y21cs172@rvrcse:~/Linux$
```


- c. ignore first field of each line.

```
y21cs172@rvrcse:~/Linux$ uniq -1 animals
cat snake
dolphin elephant
pig pig
y21cs172@rvrcse:~/Linux$
```

18. chmod: It is used to change user's, group's and owner's read, write and execute permissions of a file or directory.

- a. Changing **user** permissions.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux/permissions$ chmod u-rw 1.txt
y21cs172@rvrcse:~/Linux/permissions$ ls -l
total 4
----rw-rw- 1 y21cs172 y21cs172 3 Apr  5 14:10 1.txt
y21cs172@rvrcse:~/Linux/permissions$ chmod u+rw 1.txt
y21cs172@rvrcse:~/Linux/permissions$ ls -l
total 4
-rwxrw-rw- 1 y21cs172 y21cs172 3 Apr  5 14:10 1.txt
y21cs172@rvrcse:~/Linux/permissions$
```

- b. Changing **group** permissions.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux/permissions$ chmod g-rw 1.txt
y21cs172@rvrcse:~/Linux/permissions$ ls -l
total 4
-rw----rw- 1 y21cs172 y21cs172 3 Apr  5 14:10 1.txt
y21cs172@rvrcse:~/Linux/permissions$ chmod g+rw 1.txt
y21cs172@rvrcse:~/Linux/permissions$ ls -l
total 4
-rw-rwxrw- 1 y21cs172 y21cs172 3 Apr  5 14:10 1.txt
y21cs172@rvrcse:~/Linux/permissions$
```

- c. Changing **owner** permissions.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux/permissions$ chmod o-rw 1.txt
y21cs172@rvrcse:~/Linux/permissions$ ls -l
total 4
-rw-rw---- 1 y21cs172 y21cs172 3 Apr  5 14:10 1.txt
y21cs172@rvrcse:~/Linux/permissions$ chmod o+rw 1.txt
y21cs172@rvrcse:~/Linux/permissions$ ls -l
total 4
-rw-rw-rwx 1 y21cs172 y21cs172 3 Apr  5 14:10 1.txt
y21cs172@rvrcse:~/Linux/permissions$
```

19. paste: It is used to parallel merge or join two files by outputting lines consisting of each line separated by tab delimiter.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux$ cat head1.txt
aaa
bbb
ccc
ddd
eee
y21cs172@rvrcse:~/Linux$ cat tail1.txt
fff
ggg
hhh
iii
jjj
y21cs172@rvrcse:~/Linux$ paste head1.txt tail1.txt
aaa    fff
bbb    ggg
ccc    hhh
ddd    iii
eee    jjj
y21cs172@rvrcse:~/Linux$
```

20. cut: It is used for cutting out the sections from each line and displaying on standard output.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux$ cat animals
cat snake
monkey snake
dolphin elephant
dolphin elephant
goat elephant
pig pig
pig pig
monkey pig
y21cs172@rvrcse:~/Linux$
```

- a. **-b** option to cut specific bytes of each line in a file. We need to mention bytes followed by commands.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux$ cut -b 2,3,5 animals
ats
one
olh
olh
oa
igp
igp
one
y21cs172@rvrcse:~/Linux$
```

- b. **-c** option to specify the columns to print in each line of a file.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux$ cut -c 2,5,7 animals
asa
oe
ohn
ohn
o l
ipg
ipg
oe
y21cs172@rvrcse:~/Linux$
```

- c. **-f** option is used to print the specified field number and the **-d** option specifies the delimiter to separate the fields.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux$ cut -d " " -f 1 animals
cat
monkey
dolphin
dolphin
goat
pig
pig
monkey
y21cs172@rvrcse:~/Linux$
```

21. **join:** It is used to join the lines of two files based on a common field.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux$ cat rollnos
1 y21cs160
2 y21cs162
3 y21cs164
4 y21cs169
5 y21cs172
y21cs172@rvrcse:~/Linux$ cat students
1 Roshan
2 Sufiyan
3 Shreyas
4 Karthik
5 Tayyab
y21cs172@rvrcse:~/Linux$ join students rollnos
1 Roshan y21cs160
2 Sufiyan y21cs162
3 Shreyas y21cs164
4 Karthik y21cs169
5 Tayyab y21cs172
y21cs172@rvrcse:~/Linux$
```

22. **tee:** It is used to read from standard input and write to standard output and files.

a. Without any options.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux$ tee tee1.txt
Hello World
Hello World
This is an example of tee command
This is an example of tee command
y21cs172@rvrcse:~/Linux$ cat tee1.txt
Hello World
This is an example of tee command
```

b. **-a** command it used to append the standard input to a file rather than overwriting.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux$ tee -a tee1.txt
This is an example of appending into
This is an example of appending into
file using tee command
file using tee command
y21cs172@rvrcse:~/Linux$ cat tee1.txt
Hello World
This is an example of tee command
This is an example of appending into
file using tee command
y21cs172@rvrcse:~/Linux$
```

23. cmp: It compares two files byte by byte and returns at which byte the files first differ.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux$ cat file1.txt
Linux Programming
Unix Shell
y21cs172@rvrcse:~/Linux$ cat file2.txt
Linux Programming
Unix Shell
Operating Systems
y21cs172@rvrcse:~/Linux$ cmp file1.txt file2.txt
cmp: EOF on file1.txt after byte 29, line 2
y21cs172@rvrcse:~/Linux$
```

24. diff: It is used to compare two files and display a list of editing changes that would convert the first file into the second file. It displays three kinds of editing changes: 'a' adding lines, 'c' changing lines and 'd' deleting lines.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux$ diff file1.txt file2.txt
2a3
> Operating Systems
y21cs172@rvrcse:~/Linux$
```

MODULE – 3

DISK UTILITIES, BACKUP AND OTHER UTILITIES

1. **who:** It is used to print the information about users who are currently logged in.

```
Select Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux$ who
rwr      :0                2023-03-04 18:39 (:0)
y21cs82  pts/1            2023-03-24 15:48 (10.1.4.7)
y21cs189 pts/2            2023-03-24 15:39 (10.1.4.66)
y21cs13  pts/4            2023-03-24 15:54 (10.1.4.2)
y21cs49  pts/6            2023-03-24 15:42 (10.1.4.14)
y21cs6   pts/5            2023-03-24 15:57 (10.1.4.31)
y21cs162 pts/8            2023-03-24 15:53 (10.1.4.67)
y21cs69  pts/9            2023-03-24 15:45 (10.1.4.160)
y21cs169 pts/7            2023-03-24 15:40 (10.1.4.169)
y21cs164 pts/11           2023-03-24 15:45 (10.1.4.68)
```

2. **du:** It is used to summarize disk usage of the set of files, recursively for directories.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~$ du -h
4.0K      ./cache
4.0K      ./gnupg/private-keys-v1.d
8.0K      ./gnupg
8.0K      ./Linux/permissions
68K       ./Linux
56K       ./systemcalls
4.0K      ./ShellProgramming/sample/test2
4.0K      ./ShellProgramming/sample/test1
12K       ./ShellProgramming/sample
4.0K      ./ShellProgramming/TestDir
92K       ./ShellProgramming
304K      .
y21cs172@rvrcse:~$
```

3. **df:** It displays the amount of disk space available on the file system containing each file name argument.
If not file name is give, the space available on all currently mounted file systems is shown.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~$ df
Filesystem            1K-blocks    Used Available Use% Mounted on
udev                  8117400         0   8117400  0% /dev
tmpfs                 1630440      3292   1627148  1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv 205314024 43739964 151071916 23% /
tmpfs                 8152196         0   8152196  0% /dev/shm
tmpfs                  5120           4     5116  1% /run/lock
tmpfs                 8152196         0   8152196  0% /sys/fs/cgroup
```

4. **sed:** The stream editor utility sed scans one or more files and performs an editing action on all of the lines that match a particular condition.
- a. **Substituting text:** Substituting first character with a space in each line.

```
Telnet 10.1.1.3
y21cs172@rvcse:~/Linux$ cat sed1.txt
People just like me,
Are all around the world,
Waiting for the loved ones that they need.
And with my heart,
I make a simple wish,
Plain enough for everyone to see.
y21cs172@rvcse:~/Linux$ sed 's/^/ /' sed1.txt
 People just like me,
 Are all around the world,
 Waiting for the loved ones that they need.
 And with my heart,
 I make a simple wish,
 Plain enough for everyone to see.
y21cs172@rvcse:~/Linux$
```

- b. **Deleting text:** Deleting only those lines that contain the word 'a'.

```
Telnet 10.1.1.3
y21cs172@rvcse:~/Linux$ sed '/\<a\>/d' sed1.txt
People just like me,
Are all around the world,
Waiting for the loved ones that they need.
And with my heart,
Plain enough for everyone to see.
y21cs172@rvcse:~/Linux$
```

- c. **Inserting text:**

```
Telnet 10.1.1.3
y21cs172@rvcse:~/Linux$ cat sed11.txt
1i\
How are you\
y21cs172@rvcse:~/Linux$ sed -f sed11.txt sed1.txt
How are you

People just like me,
Are all around the world,
Waiting for the loved ones that they need.
And with my heart,
I make a simple wish,
Plain enough for everyone to see.
y21cs172@rvcse:~/Linux$
```

- d. **Replacing text:**

```
Telnet 10.1.1.3
y21cs172@rvcse:~/Linux$ cat sed2.txt
1,3c\
How are you\
y21cs172@rvcse:~/Linux$ sed -f sed2.txt sed1.txt
How are you

And with my heart,
I make a simple wish,
Plain enough for everyone to see.
y21cs172@rvcse:~/Linux$
```

MODULE – 4

PROGRAMMABLE TEXT PROCESSING

awk is a programmable text-processing utility that scans the lines of its input and performs actions on every line that matches a particular criterion

1. Accessing individual files:

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux/awk$ cat float
Wish I was floating in blue across the sky,
My imagination is strong,
And I often visit the days
When everything seemed so clear.
Now I wonder what I'm doing here at all...
y21cs172@rvrcse:~/Linux/awk$ awk '{print NF, $0}' float
9 Wish I was floating in blue across the sky,
4 My imagination is strong,
6 And I often visit the days
5 When everything seemed so clear.
9 Now I wonder what I'm doing here at all...
y21cs172@rvrcse:~/Linux/awk$
```

2. Begin and End:

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux/awk$ cat awk2
BEGIN { print "Start of file:", FILENAME }
{ print $1 $3 $NF }
END { print "End of file" }
y21cs172@rvrcse:~/Linux/awk$ awk -f awk2 float
Start of file:
Wishwassky,
Mysisstrong,
Andoftendays
Whenseemedclear.
Nowwonderall...
End of file
y21cs172@rvrcse:~/Linux/awk$
```

3. Operators:

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux/awk$ cat awk3
NR > 1 && NR < 4 { print NR, $1, $3, $NF }
y21cs172@rvrcse:~/Linux/awk$ awk -f awk3 float
2 My is strong,
3 And often days
y21cs172@rvrcse:~/Linux/awk$
```

4. Variables:

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux/awk$ cat awk4
BEGIN { print "Scanning file" }
{
printf "line %d: %s\n", NR, $0;
lineCount++;
wordCount += NF;
}
END { printf "lines = %d, words = %d\n", lineCount, wordCount }
y21cs172@rvrcse:~/Linux/awk$ awk -f awk4 float
Scanning file
line 1: Wish I was floating in blue across the sky,
line 2: My imagination is strong,
line 3: And I often visit the days
line 4: When everything seemed so clear.
line 5: Now I wonder what I'm doing here at all...
lines = 5, words = 33
y21cs172@rvrcse:~/Linux/awk$
```

5. Control Structures:

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux/awk$ cat awk5
{
for (i = NF; i >= 1; i--)
printf "%s ", $i;
printf "\n";
}
y21cs172@rvrcse:~/Linux/awk$ awk -f awk5 float
sky, the across blue in floating was I Wish
strong, is imagination My
days the visit often I And
clear. so seemed everything When
all... at here doing I'm what wonder I Now
y21cs172@rvrcse:~/Linux/awk$
```

6. Extended regular expressions:

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux/awk$ cat awk6
/t.*e/ { print $0 }
y21cs172@rvrcse:~/Linux/awk$ awk -f awk6 float
Wish I was floating in blue across the sky,
And I often visit the days
When everything seemed so clear.
Now I wonder what I'm doing here at all...
y21cs172@rvrcse:~/Linux/awk$
```


7. Condition Ranges:

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux/awk$ cat awk7
/strong/ , /clear/ { print $0 }
y21cs172@rvrcse:~/Linux/awk$ awk -f awk7 float
My imagination is strong,
And I often visit the days
When everything seemed so clear.
y21cs172@rvrcse:~/Linux/awk$
```

8. Field Separators:

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux/awk$ cat float2
Wish:I:was:floating:in:blue:across:the:sky,
My:imagination:is:strong,
And:I:often:visit:the:days
When:everything:seemed:so:clear.
Now:I:wonder:what:I'm:doing:here:at:all...
y21cs172@rvrcse:~/Linux/awk$ cat awk3
NR > 1 && NR < 4 { print NR, $1, $3, $NF }
y21cs172@rvrcse:~/Linux/awk$ awk -F: -f awk3 float2
2 My is strong,
3 And often days
y21cs172@rvrcse:~/Linux/awk$
```

9. Built-In functions:

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/Linux/awk$ cat test
1.1 a
2.2 at
3.3 eat
4.4 beat
y21cs172@rvrcse:~/Linux/awk$ cat awk9
{
printf "$1 = %g ", $1;
printf "exp = %.2g ", exp ($1);
printf "log = %.2g ", log ($1);
printf "sqrt = %.2g ", sqrt ($1);
printf "int = %d ", int ($1);
printf "substr (%s, 1, 2) = %s\n", $2, substr($2, 1, 2);
}
y21cs172@rvrcse:~/Linux/awk$ awk -f awk9 test
$1 = 1.1 exp = 3 log = 0.095 sqrt = 1 int = 1 substr (a, 1, 2) = a
$1 = 2.2 exp = 9 log = 0.79 sqrt = 1.5 int = 2 substr (at, 1, 2) = at
$1 = 3.3 exp = 27 log = 1.2 sqrt = 1.8 int = 3 substr (eat, 1, 2) = ea
$1 = 4.4 exp = 81 log = 1.5 sqrt = 2.1 int = 4 substr (beat, 1, 2) = be
y21cs172@rvrcse:~/Linux/awk$
```

MODULE - 5

SHELL SCRIPTING

Write a shell script program for the following:

1. To create a directory and list all the directory files in a directory.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/ShellProgramming$ cat Lab1.sh
echo 'Enter the Directory name to create:'
read Dir_name

if [ -e $Dir_name ]
then
    echo "$Dir_name directory already exists!"
else
    mkdir $Dir_name
    echo "$Dir_name directory created."
fi

echo 'Enter a Directory name to Display Files: '
read Dir_name

cd $Dir_name

echo 'List of Directory files in the directory: '$Dir_name
for i in */
do
    echo "$i"
done
```

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/ShellProgramming$ sh Lab1.sh
Enter the Directory name to create:
TestDir
TestDir directory created.
Enter a Directory name to Display Files:
sample
List of Directory files in the directory: sample
test1/
test2/
y21cs172@rvrcse:~/ShellProgramming$
```

2. To display a list of all the files in the current directory.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/ShellProgramming$ cat Lab2.sh
echo "Current directory:"
pwd
echo "Files in current directory:"
ls -a

Telnet 10.1.1.3
y21cs172@rvrcse:~/ShellProgramming$ sh Lab2.sh
Current directory:
/users/y21cs/y21cs172/ShellProgramming
Files in current directory:
. Lab1.sh Lab6.sh TestDir name.sh printFileContent.sh
.. Lab2.sh Lab9.sh if1.sh operators.sh sample
y21cs172@rvrcse:~/ShellProgramming$
```

3. To count no of lines, words, and characters of an input file.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/ShellProgramming$ cat Lab3.sh
read -p "Enter the file name: " filename

if [ ! -f "$filename" ]; then
    echo "File not found!"
    exit 1
fi

line_count=$(wc -l < "$filename")
word_count=$(wc -w < "$filename")
char_count=$(wc -m < "$filename")

echo "Number of lines: $line_count"
echo "Number of words: $word_count"
echo "Number of characters: $char_count"
y21cs172@rvrcse:~/ShellProgramming$
```

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/ShellProgramming$ sh Lab3.sh
Enter the file name: Lab6.sh
Number of lines: 10
Number of words: 24
Number of characters: 126
y21cs172@rvrcse:~/ShellProgramming$
```

4. To accept a file name starting and ending line numbers as arguments and display all the lines between given line numbers.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/ShellProgramming$ cat Lab4.sh
read -p "Enter file name: " filename

if [ ! -f "$filename" ]; then
    echo "File not found!"
    exit 1
fi

read -p "Enter starting line:" start_line
read -p "Enter ending line:" end_line

if [ "$start_line" -gt "$end_line" ]; then
    echo "Start line cannot be greater than end line!"
    exit 1
fi

sed -n $start_line,$end_line\p $filename
y21cs172@rvrcse:~/ShellProgramming$
```

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/ShellProgramming$ sh Lab4.sh
Enter file name: test.txt
Enter starting line:2
Enter ending line:5
Why can't you trust an atom? Because they make up literally everything.
Why was six afraid of seven? Because seven eight nine.
What do you call a hippie's wife? Mississippi.
What do you call a bear with no teeth? A gummy bear.
y21cs172@rvrcse:~/ShellProgramming$
```

5. To deletes all lines containing the specified word in one or more files supplied as arguments to it.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/ShellProgramming$ cat Lab5.sh
read -p "Enter a word to delete: " word
for i in $@
do
if [ -e $i ]
then
        echo "The file content after deleting $w:"
        sed '/'$word'/d' $i
else
        echo "File doesn't exist!"
fi
done
y21cs172@rvrcse:~/ShellProgramming$
```

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/ShellProgramming$ sh Lab5.sh test1.txt
Enter a word to delete: the
The file content after deleting :
Shell Programming
hi hello
y21cs172@rvrcse:~/ShellProgramming$
```

6. To test whether the given file is existing or not.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/ShellProgramming$ cat Lab6.sh
read -p "Enter filename: " filename

if [ -f $filename ]
then
        echo "$filename exists!"
else
        echo "$filename doesn't exist!"
fi
y21cs172@rvrcse:~/ShellProgramming$
```

Telnet 10.1.1.3

```
y21cs172@rvrcse:~/ShellProgramming$ sh Lab6.sh
Enter filename: Lab1.sh
Lab1.sh exists!
y21cs172@rvrcse:~/ShellProgramming$
```

7. To read, delete and append a file.

Telnet 10.1.1.3

```
y21cs172@rvrcse:~/ShellProgramming$ cat Lab7.sh
read -p "Enter the file name: " fname
if [ -f $fname ]
then
    echo "Contents of file: "
    cat $fname
else
    echo "File does not exist"
fi
read -p "Enter the file name to delete: " fname

if [ -f $fname ]
then
    rm $fname
    echo "File deleted"
else
    echo "File does not exist"
fi
read -p "Enter the file name to append: " fname

if [ -f $fname ]
then
    echo "Enter text to append: "
    cat >> $fname
else
    echo "File does not exist"
fi
```

Telnet 10.1.1.3

```
y21cs172@rvrcse:~/ShellProgramming$ sh Lab7.sh
Enter the file name: sample.txt
Contents of file:
Hi, My name is Tayyab.
Enter the file name to delete: test2.txt
File deleted
Enter the file name to append: test1.txt
Enter text to append:
Shell Programming
```

8. To store all command line arguments to an array and print.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/ShellProgramming$ cat Lab8.sh
arr=$@

echo "The arguments are: "
for i in $arr
do
    echo $i
done
```

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/ShellProgramming$ sh Lab8.sh Hi, My name is Tayyab.
The arguments are:
Hi,
My
name
is
Tayyab.
y21cs172@rvrcse:~/ShellProgramming$
```

9. To print the calendar month by default

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/ShellProgramming$ cat Lab9.sh
month=$1
year=$2
echo "Calender of the given month and year:"
cal $month $year
```

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/ShellProgramming$ sh Lab9.sh April 2004
Calender of the given month and year:
April 2004
Su Mo Tu We Th Fr Sa
          1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30
```

MODULE-6

FILE MANAGEMENT SYSTEM CALLS

1. Write a program on File management System Calls: open (), read (), write (), close ().

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/systemcalls$ cat Lab1.c
#include<stdio.h>
#include<fcntl.h>
#include<stdlib.h>
#include<string.h>
void main()
{
    int fd1,fd2;
    fd1 = open("sample.txt", O_RDONLY);
    fd2 = open("test.txt", O_CREAT | O_RDWR, 0700);
    printf("fd1 = %d\n",fd1);
    printf("fd2 = %d\n",fd2);
    char *c = (char*)malloc(20*sizeof(char));
    int s = read(fd1,c,10);
    c[s] = '\0';
    printf("Contents of first %d bytes of fd1: %s\n",s,c);
    write(fd2, "RVR & JC", 8);
    close(fd1);
    close(fd2);
}
```

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/systemcalls$ ./a.out
fd1 = 3
fd2 = 4
Contents of first 10 bytes of fd1: My IQ test
y21cs172@rvrcse:~/systemcalls$
```

2. Write a program on File handling system call: perror ().

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/systemcalls$ cat Lab2.c
#include<stdio.h>
#include<errno.h>
#include<sys/file.h>
void main()
{
    int fd1, fd2;
    fd1 = open("nonexist1.txt",O_RDONLY);
    if(fd1 == -1)
    {
        printf("errno = %d\n",errno);
        perror("Could not open the file to read");
    }
    fd2 = open("nonexist2.txt",O_WRONLY);
    if(fd2 == -1)
    {
        printf("errno = %d\n",errno);
        perror("Could not open the file to write");
    }
}
```

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/systemcalls$ ./a.out
errno = 2
Could not open the file to read: No such file or directory
errno = 2
Could not open the file to write: No such file or directory
y21cs172@rvrcse:~/systemcalls$
```

3. Write a program for demonstrating dup () and dup2() system calls.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/systemcalls$ cat Lab3.c
#include<stdio.h>
#include<unistd.h>
#include<fcntl.h>
void main()
{
    int old_fd, new_fd1, new_fd2;
    old_fd = open("test.txt", O_RDWR);
    printf("The Old File Descriptor is: %d\n",old_fd);
    new_fd1 = dup(old_fd);
    printf("The First New File Descriptor is: %d\n",new_fd1);
    new_fd2 = dup2(old_fd, 7);
    printf("The Second New File Descriptor is: %d\n",new_fd2);
    close(old_fd);
}
```

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/systemcalls$ ./a.out
The Old File Descriptor is: 3
The First New File Descriptor is: 4
The Second New File Descriptor is: 7
y21cs172@rvrcse:~/systemcalls$
```


MODULE-7

PROCESS MANAGEMENT SYSTEM CALLS

1. Write a program to create two processes, to run a loop in which one process adds all even numbers and other process adds all odd numbers (use fork () system call).

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/systemcalls$ cat Lab4.c
#include<stdio.h>
#include<sys/types.h>
#define Max_count 10
void EvenSum();
void OddSum();
void main()
{
    pid_t pid = fork();
    if(pid==0)
        EvenSum();
    else
        OddSum();
}
void OddSum()
{
    int sum=0;
    for(int i=1;i<=Max_count;i++)
        if(i%2 != 0)
            sum += i;
    printf("Sum of Odd Numbers upto %d: %d\n",Max_count,sum);
}
void EvenSum()
{
    int sum=0;
    for(int i=1;i<=Max_count;i++)
        if(i%2 == 0)
            sum += i;
    printf("Sum of Even Numbers upto %d: %d\n",Max_count,sum);
}
```

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/systemcalls$ ./a.out
Sum of Odd Numbers upto 10: 25
y21cs172@rvrcse:~/systemcalls$ Sum of Even Numbers upto 10: 30
```

2. Write a Program to create orphan process.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/systemcalls$ cat Lab5.c
#include<stdio.h>
#include<sys/types.h>
#include<unistd.h>
void main()
{
    int pid = fork();
    if(pid>0)
        printf("In Parent Process");
    else if(pid==0)
    {
        sleep(5);
        printf("In Cild Process");
    }
}
```

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/systemcalls$ ./a.out
In Parent Processy21cs172@rvrcse:~/systemcalls$
```

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/systemcalls$ ./a.out
In Parent Processy21cs172@rvrcse:~/systemcalls$ In Cild Process
```

3. Write a Program to create a zombie process and how to avoid Zombie using wait ().

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/systemcalls$ cat Lab6.c
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/wait.h>
#include<sys/types.h>
void main()
{
    pid_t child = fork();
    if(child>0)
    {
        printf("Parent process start\n");
        wait(&child);
        printf("Parent process end\n");
    }
    else if(child==0)
    {
        printf("Child process start\n");
        sleep(10);
        printf("Child process end\n");
        exit(0);
    }
}
```

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/systemcalls$ ./a.out
Parent process start
Child process start
```

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/systemcalls$ ./a.out
Parent process start
Child process start
Child process end
Parent process end
y21cs172@rvrcse:~/systemcalls$
```

MODULE – 8

OPERATIONS ON SIGNALS

1. Write a program for Requesting an alarm signal to execute user defined alarm handler.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/systemcalls$ cat Lab7.c
#include<stdio.h>
#include<signal.h>
int alarm_flag = 0;
void alarm_handler();
void main()
{
    signal(SIGALRM, alarm_handler);
    alarm(5);
    printf("Looping...\n");
    while(!alarm_flag)
    {
        printf("Inside the Loop\n");
        pause();
    }
    printf("Loop ends due to alarm signal\n");
}
void alarm_handler()
{
    printf("An alarm clock signal was recieved\n");
    alarm_flag = 1;
}
```

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/systemcalls$ ./a.out
Looping...
Inside the Loop
```

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/systemcalls$ ./a.out
Looping...
Inside the Loop
An alarm clock signal was recieved
Loop ends due to alarm signal
y21cs172@rvrcse:~/systemcalls$
```

2. Write a program to demonstrate Suspending and Resuming Processes.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/systemcalls$ cat Lab8.c
#include<stdio.h>
#include<signal.h>
void main()
{
    int pid1, pid2;
    pid1 = fork();
    if(pid1==0)
    {
        while(1)
        {
            printf("Process 1 is alive\n");
            sleep(1);
        }
    }
    pid2 = fork();
    if(pid2==0)
    {
        while(1)
        {
            printf("Process 2 is alive\n");
            sleep(1);
        }
    }
    sleep(3);
    kill(pid1, SIGSTOP);
    sleep(3);
    kill(pid1, SIGCONT);
    sleep(3);
    kill(pid1, SIGINT);
    kill(pid2, SIGINT);
}
```

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/systemcalls$ ./a.out
Process 1 is alive
Process 2 is alive
Process 1 is alive
Process 2 is alive
Process 1 is alive
Process 2 is alive
Process 2 is alive
Process 2 is alive
Process 2 is alive
Process 1 is alive
Process 2 is alive
Process 1 is alive
Process 2 is alive
Process 1 is alive
Process 2 is alive
```

MODULE – 9
INTER PROCESS COMMUNICATION

1. Write a program to implement the concept of pipes.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/systemcalls$ cat piping.c
#include<stdio.h>
#include<string.h>
#define READ 0
#define WRITE 1
char *phrase = "This is a piped text";
void main()
{
    int fd[2], bytesRead;
    char message[100];
    pipe(fd);
    if(fork()==0)
    {
        close(fd[READ]);
        write(fd[WRITE],phrase,strlen(phrase)+1);
        close(fd[WRITE]);
    }
    else
    {
        close(fd[WRITE]);
        bytesRead = read(fd[READ],message,100);
        printf("Read %d bytes: %s\n",bytesRead,message);
        close(fd[READ]);
    }
}
y21cs172@rvrcse:~/systemcalls$ ./a.out
Read 21 bytes: This is a piped text
y21cs172@rvrcse:~/systemcalls$
```

2. Write a program to implement the concept of named pipes.

```
Telnet 10.1.1.3
y21cs172@rvrcse:~/systemcalls$ mknod myPipe p
y21cs172@rvrcse:~/systemcalls$ ls -l myPipe
prw-rw-r-- 1 y21cs172 y21cs172 0 Jul  5 11:01 myPipe
y21cs172@rvrcse:~/systemcalls$ chmod ug-rw myPipe
y21cs172@rvrcse:~/systemcalls$ ls -l myPipe
p-----r-- 1 y21cs172 y21cs172 0 Jul  5 11:01 myPipe
y21cs172@rvrcse:~/systemcalls$
```