4.(a) Shell script to check whether a number is odd or prime

read -p "Enter a numbe " number

**if** [ $((number%2)) -eq 0 ]

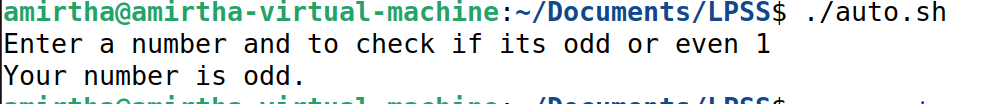
**then**

echo "Your number is even"

**else**

echo "Your number is odd."

Fi



(b) To find factorial

#!/bin/bash

echo Enter the number you want to get factorial

read mynumber

factorial=1

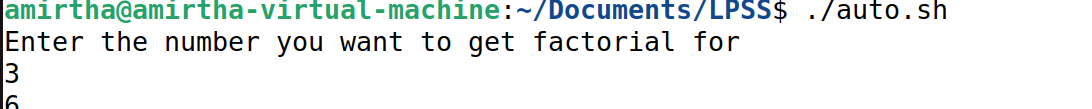
for ((i=1;i<=mynumber;i++))

do

factorial=$(($factorial\*$i))

done

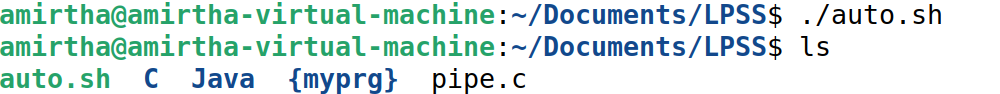
echo $factorial



4 C, to create multiple directory in shell script

#!/bin/bash

mkdir -p {Java,C}



d)

**#! /bin/bash**

**echo Enter the filename**

**read file**

**c=`cat $file | wc -c`**

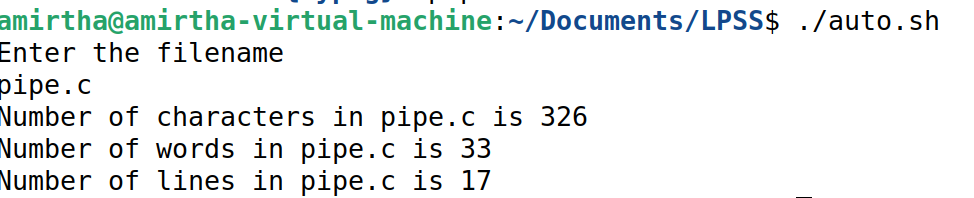
**w=`cat $file | wc -w`**

**l=`grep -c "." $file`**

**echo Number of characters in $file is $c**

**echo Number of words in $file is $w**

**echo Number of lines in $file is $l**



5 a) pipe using fork()

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/wait.h>

int main(void) {

int pipefds[2];

char \*data;

char buffer[10];

if(pipe(pipefds) == -1) {

perror("pipe");

return -1;

}

pid\_t pid = fork();

if(pid == 0) { // in child process

data = "abcdefgh\0";

close(pipefds[0]);

write(pipefds[1], data, 10);

printf("Generating data in child and sending to parent...\n");

sleep(2); // intentional delay

exit(EXIT\_SUCCESS);

}

if(pid > 0) { // in main process

wait(NULL); // wait for child process to finish

close(pipefds[1]);

read(pipefds[0], buffer, 10);

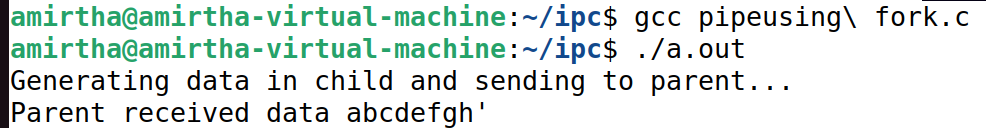
close(pipefds[0]); // close read fd

printf("Parent received data %s'\n", buffer);

}

return 0;

}



5 b) FIFO

1.Create a named pipe using mkfifo()

#include<stdio.h>

#include<sys/types.h>

#include<sys/stat.h>

int main()

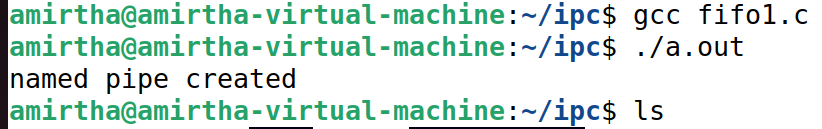
{

int res;

res = mkfifo("fifochannel",0777); //creates a named pipe with the name fifo

printf("named pipe created\n");

}



is shown

2.create server process.

#include<unistd.h>

#include<stdio.h>

#include<fcntl.h>

int main()

{

int res,n;

char buffer[50];

res=open("fifochannel",O\_WRONLY);

write(res,"INDIA is world's largest democracy",35);

printf("Sender Process %d sent the data\n",getpid());

}



It will be blank for sometime

3)Create a program fifoclient.c and execute in terminal 2.

#include<unistd.h>

#include<stdio.h>

#include<fcntl.h>

int main()

{

int res,n;

char buffer[100];

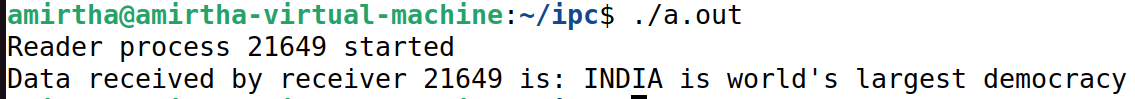
res=open("fifochannel",O\_RDONLY);

n=read(res,buffer,100);

printf("Reader process %d started\n",getpid());

printf("Data received by receiver %d is: %s\n",getpid(), buffer);

}



After executing fifoclient.c,check the status of fifoserver.c terminal