**OS LABORTORY**

**END TERM REPORT**

***By***

**NAME OF THE CANDIDATE:**

**Shubhanshoo Dwivedi**

**K18NZ**

**011**



**Department of Intelligent Systems**

**School of Computer Science Engineering**

**Lovely Professional University, Jalandhar**

**APRIL-2020**

**Student Declaration**

**This is to declare that this report has been written by me/us. No part of the report is copied from other sources. All information included from other sources have been duly acknowledged. I/We aver that if any part of the report is found to be copied, I/we are shall take full responsibility for it.**

**Shubhanshoo dwivedi**

**Roll number: 11**

**Place: Jalandhar**

**Date: 17-04-2020**

**GITHUB LINK:** **https://github.com/shubhanshoo123/OS-LAB.git**

**ANS-1:**

#include<stdio.h>

#include <pthread.h>

#include <semaphore.h>

#include<stdlib.h>

#include<unistd.h>

#define n 5 //number of philosophers

#define THINKING 1

#define HUNGRY 2

#define EATING 0

#define LEFT (phili + 4) % n

#define RIGHT (phili + 1) % n

int state[n],i=0;

int phil[n];

int c=0;

sem\_t mutex;

sem\_t S[n];

void test(int phili)

{

if (state[phili] == HUNGRY

&& state[LEFT] != EATING

&& state[RIGHT] != EATING) {

// state that eating

state[phili] = EATING;

sleep(2);

printf("Philosopher %d -> takes chops (%d,%d)\n",

phili + 1, LEFT + 1, phili + 1);

printf("==>Philosopher %d is Eating\n", phili + 1);

sem\_post(&S[phili]);

}

}

void take\_fork(int phili)

{

sem\_wait(&mutex);

state[phili] = HUNGRY;

printf("\tPhilosopher %d is Hungry\n", phili + 1);

test(phili);

sem\_post(&mutex);

sem\_wait(&S[phili]);

sleep(1);

}

void put\_fork(int phili)

{

sem\_wait(&mutex);

state[phili] = THINKING;

printf("Philosopher %d -> puts down chops (%d,%d)\n",

phili + 1, LEFT + 1, phili + 1);

c= c+1;

if (c==6)

{

exit(0);

}

printf("Philosopher %d is thinking\n", phili + 1);

test(LEFT);

test(RIGHT);

sem\_post(&mutex);

}

void\* philospher(void\* num)

{

while (1) {

int\* i = num;

sleep(1);

take\_fork(\*i);

sleep(0);

put\_fork(\*i);

}

}

int main()

{

while(i<n){

phil[i]=i;

i++;

}

int i;

pthread\_t thread\_id[n];

sem\_init(&mutex, 0, 1);

for (i = 0; i < n; i++)

{ sem\_init(&S[i], 0, 0); }

for (i = 0; i < n; i++) {

pthread\_create(&thread\_id[i], NULL,

philospher, &phil[i]);

printf("Philosopher %d is thinking\n", i+1);

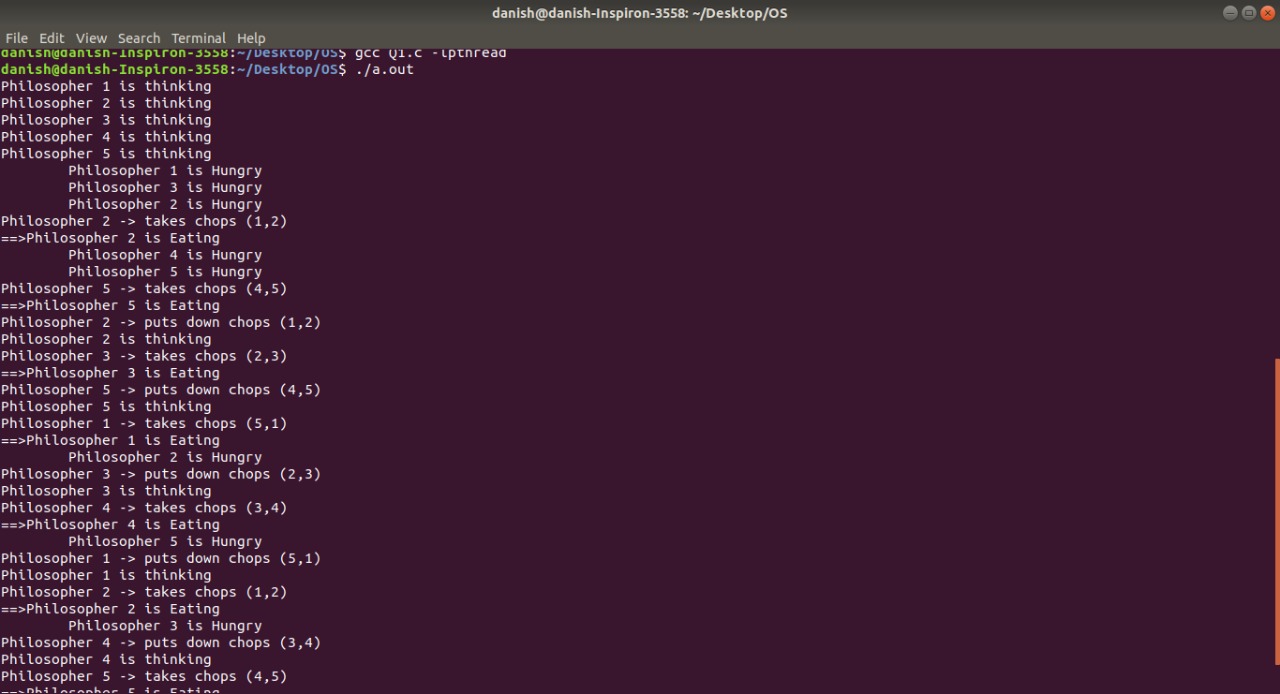
}

for (i = 0; i < n; i++)

pthread\_join(thread\_id[i], NULL);

}

Output:



**Ans-2:**

#include<stdio.h>

#include<unistd.h>

int main() {

int p1[2], p2[2];

int r1, r2;

int pid;

char a[20] = "Hello";

char b[20] = "Message Received";

char c[20];

pipe(p1);

pipe(p2);

pid = fork();

if (pid != 0){

close(p1[0]);

close(p2[1]);

printf("Parent Message: %s\n",a);

write(p1[1],a, sizeof(a));

read(p2[0],c, sizeof(c));

printf("Acknowledge Received : %s\n",c);

} else {

close(p1[1]);

close(p2[0]);

read(p1[0],c, sizeof(c));

printf("Parent message received: %s\n",c);

printf("Acknowlegment by Child: %s\n",b);

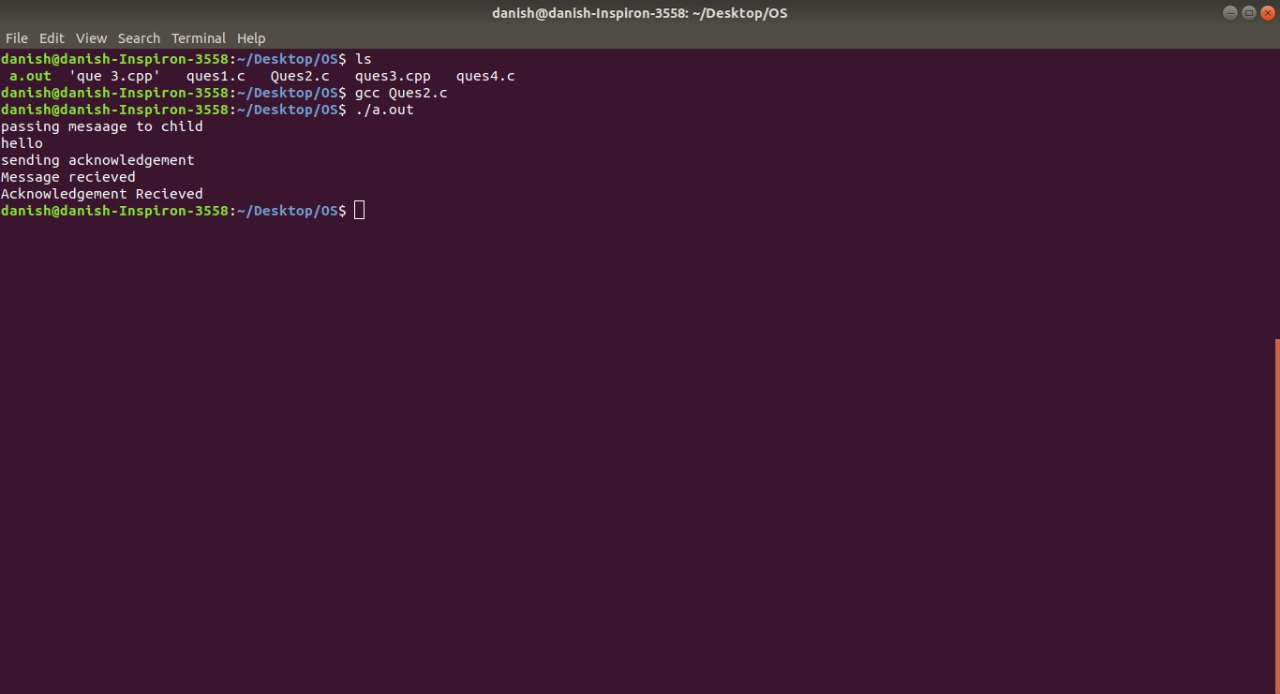
write(p2[1],b, sizeof(b));

}

return 0;

}

Output:



**Ans-3**

#include<stdio.h>

#include<stlib.h>

#include<unistd.h>

int main()

{

int fd[2],n;

char buffer[100];

pid\_t p;

pipe(fd);

p=fork();

if>(p>0)

{

Close(fd[0]);

Print(“passing value to child\n”)

Write(fd[1],”end term practical\n”20);

Sleep(1);

}

Else

{

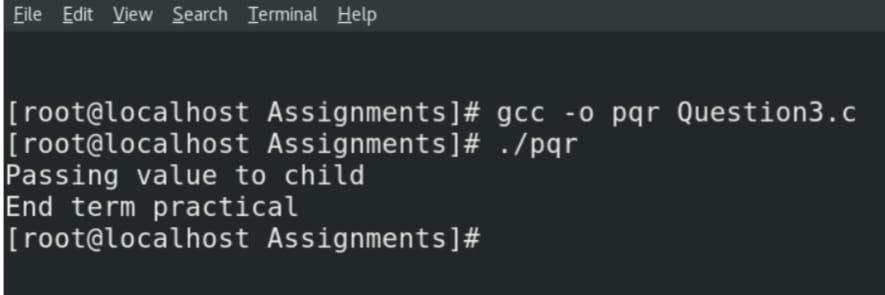
Close(fd[1]);

n=read(fd[0],buffer,100);

write(1,buffer,n);

}

}

**OUTPUT:** 

**Ans-4**

#include<stdio.h>

#include<stlib.h>

#include<unistd.h>

int main()

{

File\*fp;

Char ch;

int num;

long length;

print(“Enter the value of num”);

scnaf(“%d,&num);

fb=fopen(“test.os”,r);

if (fp==NULL)

{

Puts(“cantot open this file”);

Fsseek(fp,0,SEEK\_END)

Length=ftell(fp);

Fseek(fp,(length-num),SEEK\_SET);

do

{

Ch=fgetc(fp);

}

Putchar(ch);

}

While(ch!=EOF);

Fclose(fp);

return(0);

}

OUTPUT:

LAST 10 CHAR OF FILE ON MY SYTSEM