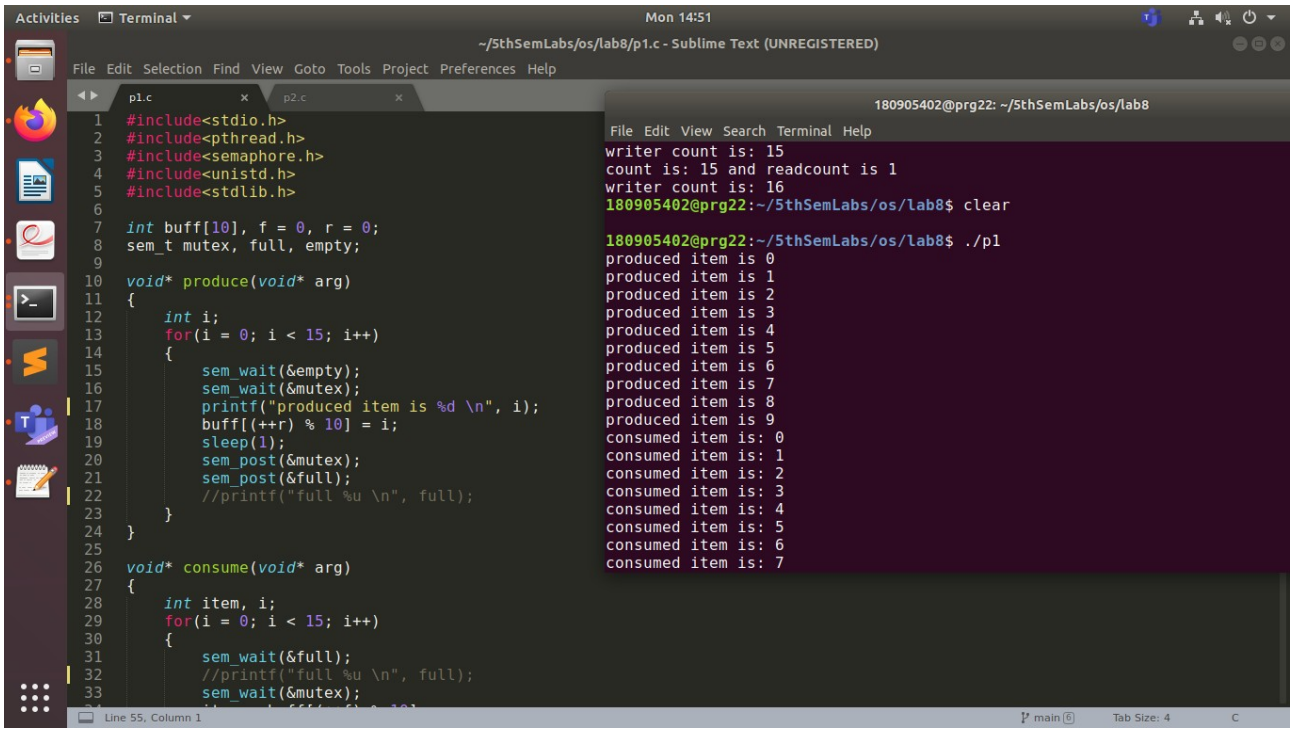
//Lab7

q1)

```
#include<stdio.h>
#include<pthread.h>
#include<semaphore.h>
#include<unistd.h>
#include<stdlib.h>

int buff[10], f = 0, r = 0;
sem_t mutex, full, empty;

void* produce(void* arg)
{
	int i;
	for(i = 0; i < 15; i++)
	{
		sem_wait(&empty);
		sem_wait(&mutex);
		printf("produced item is %d \n", i);
		buff[(++r) % 10] = i;
		sleep(1);
		sem_post(&mutex);
		sem_post(&full);
		//printf("full %u \n", full);
	}
}

void* consume(void* arg)
{
	int item, i;
	for(i = 0; i < 15; i++)
	{
		sem_wait(&full);
		//printf("full %u \n", full);
		sem_wait(&mutex);
		item = buff[(++f) % 10];
		printf("consumed item is: %d \n", item);
		sleep(1);
		sem_post(&mutex);
		sem_post(&empty);
	}
}

int main()
{
	pthread_t tid1, tid2;
	sem_init(&mutex, 0, 1);
	sem_init(&full, 0, 0);
	sem_init(&empty, 0, 10);
	pthread_create(&tid1, NULL, produce, NULL);
```

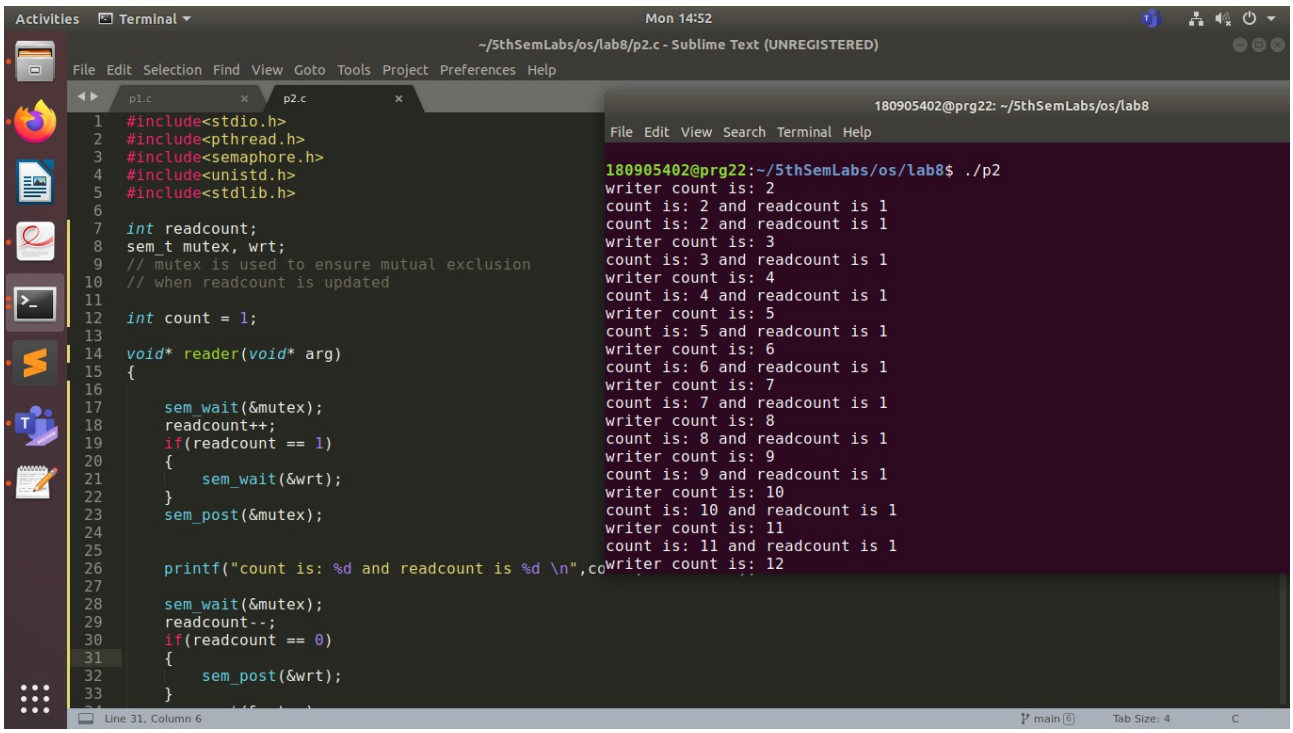
```
        pthread_create(&tid2, NULL, consume, NULL);
        pthread_join(tid1, NULL);
        pthread_join(tid2, NULL);
}
```



q2)

```
#include<stdio.h>
#include<pthread.h>
#include<semaphore.h>
#include<unistd.h>
#include<stdlib.h>

int readcount;
sem_t mutex, wrt;
// mutex is used to ensure mutual exclusion
// when readcount is updated

int count = 1;

void* reader(void* arg)
{

        sem_wait(&mutex);
        readcount++;
        if(readcount == 1)
        {
                sem_wait(&wrt);
        }
        sem_post(&mutex);
```

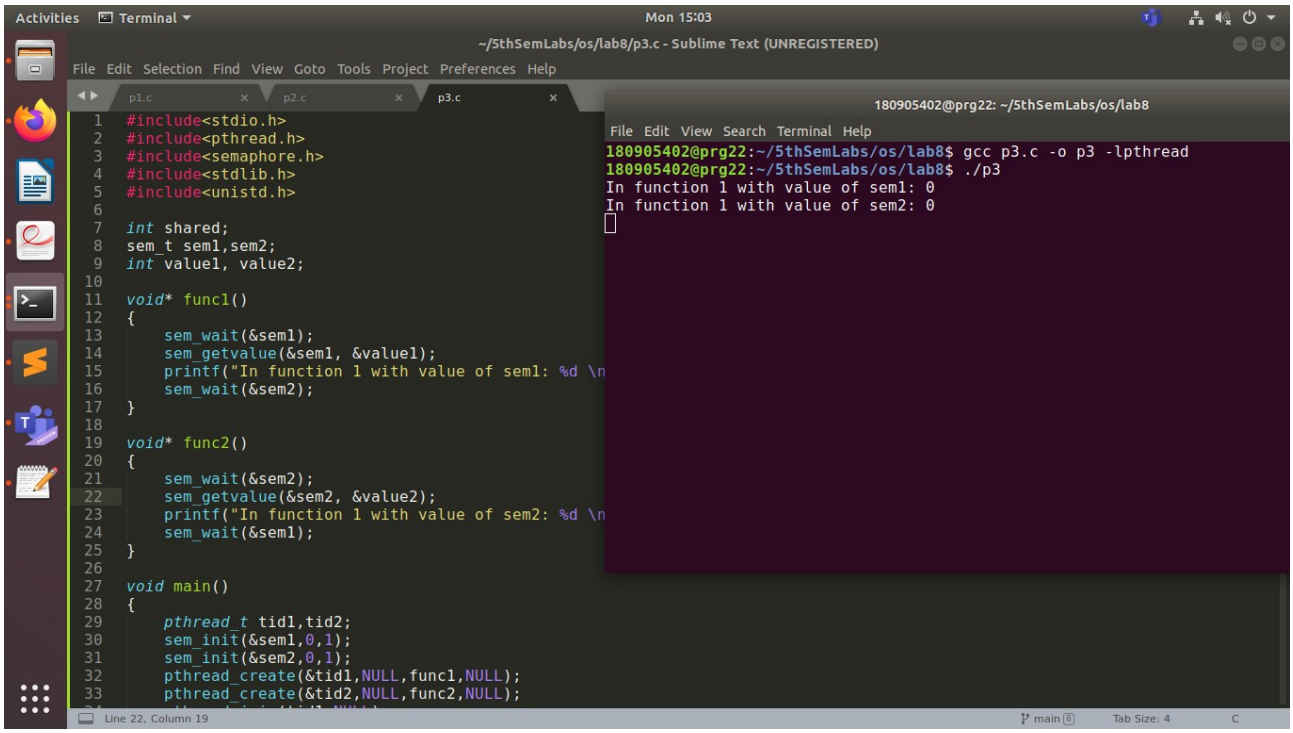
```
	printf("count is: %d and readcount is %d \n",count, readcount);

	sem_wait(&mutex);
	readcount--;
	if(readcount == 0)
	{
		sem_post(&wrt);
	}
	sem_post(&mutex);
}

void* writer(void* arg)
{

	sem_wait(&wrt);
	count++;
	printf("writer count is: %d \n", count);
	sem_post(&wrt);
}

int main()
{
	int readcount = 0;
	pthread_t tid1, tid2;
	sem_init(&mutex, 0, 1);
	sem_init(&wrt, 0, 1);

	for(int i = 0; i < 15; i++)\
	{
		pthread_create(&tid1, NULL, reader, NULL);
		pthread_create(&tid2, NULL, writer, NULL);
		pthread_join(tid1, NULL);
		pthread_join(tid2, NULL);
	}

	sem_destroy(&mutex);
	sem_destroy(&wrt);

}
```



q3)

```
#include<stdio.h>
#include<pthread.h>
#include<semaphore.h>
#include<stdlib.h>
#include<unistd.h>

int shared;
sem_t sem1,sem2;
int value1, value2;

void* func1()
{
        sem_wait(&sem1);
        sem_getvalue(&sem1, &value1);
        printf("In function 1 with value of sem1: %d \n", value1);
        sem_wait(&sem2);
}

void* func2()
{
        sem_wait(&sem2);
        sem_getvalue(&sem2, &value2);
        printf("In function 1 with value of sem2: %d \n", value2);
        sem_wait(&sem1);
}

void main()
{
        pthread_t tid1,tid2;
```

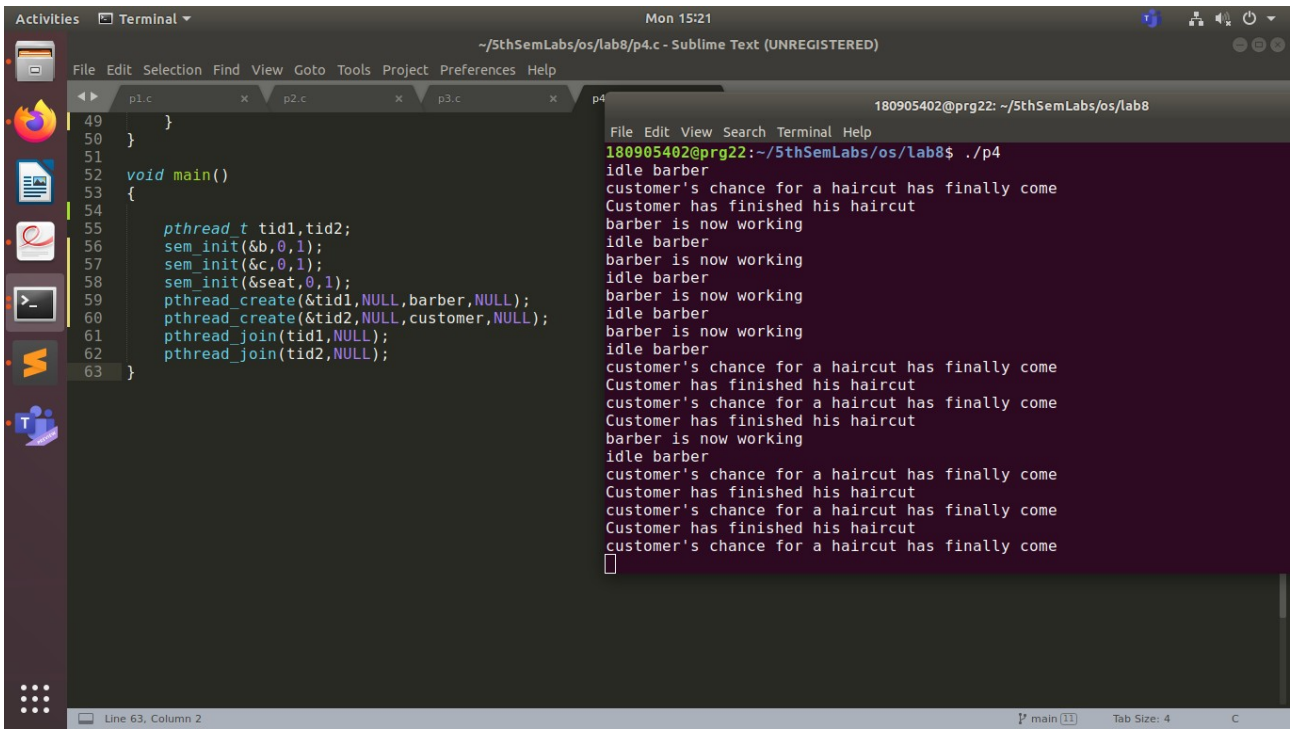
```
        sem_init(&sem1,0,1);
        sem_init(&sem2,0,1);
        pthread_create(&tid1,NULL,func1,NULL);
        pthread_create(&tid2,NULL,func2,NULL);
        pthread_join(tid1,NULL);
        pthread_join(tid2,NULL);
}
```



q4)

```
#include<stdio.h>
#include<pthread.h>
#include<semaphore.h>
#include<stdlib.h>
#include<unistd.h>

int shared;
sem_t b, c;
sem_t seat;
int seats = 5;

void* barber()
{
        while(1)
        {
                printf("idle barber \n");
                sem_wait(&c);
                sem_wait(&seat);
                printf("barber is now working \n");
                seats++;
                sem_post(&seat);
                sem_post(&c);
```

```
	}
}

void* customer()
{
	while(1)
	{
		if(seats > 0)
		{
			seats--;
			sem_post(&c);

			sem_wait(&b);
			sem_wait(&seat);
			printf("customer's chance for a haircut has finally come\n");
			sleep(2);
			seats++;
			sem_post(&seat);
			sem_post(&b);
			printf("Customer has finished his haircut\n");
		}
		else
		{
			printf("shop full\n");
			sleep(2);
		}
	}
}

void main()
{

	pthread_t tid1,tid2;
	sem_init(&b,0,1);
	sem_init(&c,0,1);
	sem_init(&seat,0,1);
	pthread_create(&tid1,NULL,barber,NULL);
	pthread_create(&tid2,NULL,customer,NULL);
	pthread_join(tid1,NULL);
	pthread_join(tid2,NULL);
}
```


Activities
Terminal
Mon 15:21
~/5thSemLabs/os/lab8/p4.c - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
p1.c
p2.c
p3.c
p4
}
}
void main()
{
pthread_t tid1,tid2;
sem_init(&b,0,1);
sem_init(&c,0,1);
sem_init(&seat,0,1);
pthread_create(&tid1,NULL,barber,NULL);
pthread_create(&tid2,NULL,customer,NULL);
pthread_join(tid1,NULL);
pthread_join(tid2,NULL);
}
180905402@prg22: ~/5thSemLabs/os/lab8
File Edit View Search Terminal Help
180905402@prg22:~/5thSemLabs/os/lab8$ ./p4
idle barber
customer's chance for a haircut has finally come
Customer has finished his haircut
barber is now working
idle barber
barber is now working
idle barber
barber is now working
idle barber
barber is now working
idle barber
customer's chance for a haircut has finally come
Customer has finished his haircut
customer's chance for a haircut has finally come
Customer has finished his haircut
barber is now working
idle barber
customer's chance for a haircut has finally come
Customer has finished his haircut
customer's chance for a haircut has finally come
Customer has finished his haircut
customer's chance for a haircut has finally come
Line 63, Column 2
main
Tab Size: 4
C

S