

Activities Text Editor Sun 21:55

Open page.c Save

```
#include<stdio.h>
void main()
{
    int p,d,add;
    int f[8];
    printf("Enter logical address Page No(0-3)");
    scanf("%d",&p);
    printf("Enter logical address Offset(0-3)");
    scanf("%d",&d);
    int pt[]={2,5,4,7};
    add = pt[p]*4 + d;
    printf("Physical Address is:%d",add);
}
```

C Tab Width: 8 Ln 13, Col 2 INS

Activities Terminal Sun 21:57

tushar@tushar-VirtualBox: ~

File Edit View Search Terminal Help

```
tushar@tushar-VirtualBox:~$ gedit page.c
tushar@tushar-VirtualBox:~$ gcc page.c
tushar@tushar-VirtualBox:~$ ./a.out
Enter logical address Page No(0-3)2
Enter logical address Offset(0-3)3
Physical Address is:19tushar@tushar-VirtualBox:~$
```

```
ProducerConsumer - Notepad
File Edit Format View Help
public class ProducerConsumer
{
    public static void main(String[] args)
    {
        Shop c = new Shop();
        Producer p1 = new Producer(c, 1);
        Consumer c1 = new Consumer(c, 1);
        p1.start();
        c1.start();
    }
}
class Shop
{
    private int materials;
    private boolean available = false;
    public synchronized int get()
    {
        while (available == false)
        {
            try
            {
                wait();
            }
            catch (InterruptedException ie)
            {
            }
        }
        available = false;
        notifyAll();
        return materials;
    }
    public synchronized void put(int value)
    {
        while (available == true)
        {
            try
            {
                wait();
            }
            catch (InterruptedException ie)
            {
                ie.printStackTrace();
            }
        }
        materials = value;
        available = true;
        notifyAll();
    }
}
class Consumer extends Thread
{
    private Shop Shop;
    private int number;
    public Consumer(Shop c, int number)
    {
        Shop = c;
        this.number = number;
    }
}
```

```
ProducerConsumer - Notepad
File Edit Format View Help
        ie.printStackTrace();
    }
    materials = value;
    available = true;
    notifyAll();
}
class Consumer extends Thread
{
    private Shop Shop;
    private int number;
    public Consumer(Shop c, int number)
    {
        Shop = c;
        this.number = number;
    }
    public void run()
    {
        int value = 0;
        for (int i = 0; i < 10; i++)
        {
            value = Shop.get();
            System.out.println("Consumed value " + this.number+ " got: " + value);
        }
    }
}
class Producer extends Thread
{
    private Shop Shop;
    private int number;
    public Producer(Shop c, int number)
    {
        Shop = c;
        this.number = number;
    }
    public void run()
    {
        for (int i = 0; i < 10; i++)
        {
            Shop.put(i);
            System.out.println("Produced value " + this.number+ " put: " + i);
            try
            {
                sleep((int)(Math.random() * 100));
            }
            catch (InterruptedException ie)
            {
                ie.printStackTrace();
            }
        }
    }
}
}
```

```
Command Prompt
Microsoft Windows [Version 10.0.18362.476]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\TUSHAR>cd desktop
C:\Users\TUSHAR\Desktop>javac ProducerConsumer.java
C:\Users\TUSHAR\Desktop>java ProducerConsumer
Consumed value 1 got: 0
Produced value 1 put: 0
Produced value 1 put: 1
Consumed value 1 got: 1
Produced value 1 put: 2
Consumed value 1 got: 2
Produced value 1 put: 3
Consumed value 1 got: 3
Produced value 1 put: 4
Consumed value 1 got: 4
Produced value 1 put: 5
Consumed value 1 got: 5
Produced value 1 put: 6
Consumed value 1 got: 6
Produced value 1 put: 7
Consumed value 1 got: 7
Produced value 1 put: 8
Consumed value 1 got: 8
Produced value 1 put: 9
Consumed value 1 got: 9

C:\Users\TUSHAR\Desktop>
```

Activities Text Editor Mon 00:18

seg.c

```
#include<stdio.h>
void main()
{
int b[10]={219,2300,90,1327,1952}, |
l[10]={600,14,100,580,96}, s, d, p;
printf("Enter segment number");
scanf("%d",&s);
printf("Enter offset");
scanf("%d",&d);
if(d<l[s])
{
p = b[s] + d;
printf("\nPhysical Address is %d",p);
}
else
{
printf("Trap to OS");
}
}
```

C Tab Width: 8 Ln 4, Col 36 INS

Activities Terminal Mon 00:17

tushar@tushar-VirtualBox: ~

File Edit View Search Terminal Help

```
tushar@tushar-VirtualBox:~$ gedit seg.c
tushar@tushar-VirtualBox:~$ gcc seg.c
tushar@tushar-VirtualBox:~$ ./a.out
Enter segment number3
Enter offset400

Physical Address is 1727tushar@tushar-VirtualBox:~$
```

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <pthread.h>
4 #include <semaphore.h>
5
6 typedef struct {
7     int position;
8     int count;
9     sem_t *forks;
10    sem_t *lock;
11 } params_t;
12
13 void initialize_semaphores(sem_t *lock, sem_t *forks, int num_forks);
14 void run_all_threads(pthread_t *threads, sem_t *forks, sem_t *lock, int num_philosophers);
15
16 void *philosopher(void *params);
17 void think(int position);
18 void eat(int position);
19
20 int main(int argc, char *args[])
21 {
22     int num_philosophers = 5;
23
24     sem_t lock;
25     sem_t forks[num_philosophers];
26     pthread_t philosophers[num_philosophers];
27
28     initialize_semaphores(&lock, forks, num_philosophers);
29     run_all_threads(philosophers, forks, &lock, num_philosophers);
30     pthread_exit(NULL);
31 }
32
33 void initialize_semaphores(sem_t *lock, sem_t *forks, int num_forks)
34 {
35     int i;
36     for(i = 0; i < num_forks; i++) {
37         sem_init(&forks[i], 0, 1);
38     }
39
40     sem_init(lock, 0, num_forks - 1);
41 }
42
43 void run_all_threads(pthread_t *threads, sem_t *forks, sem_t *lock, int num_philosophers)
44 {
45     int i;
46     for(i = 0; i < num_philosophers; i++) {
```

Command line arguments:

```
47 void run_all_threads(pthread_t *threads, sem_t *forks, sem_t *lock, int num_philosophers)
48 {
49     int i;
50     for(i = 0; i < num_philosophers; i++) {
51         params_t *arg = malloc(sizeof(params_t));
52         arg->position = i;
53         arg->count = num_philosophers;
54         arg->lock = lock;
55         arg->forks = forks;
56
57         pthread_create(&threads[i], NULL, philosopher, (void *)arg);
58     }
59 }
60
61 void *philosopher(void *params)
62 {
63     int i;
64     params_t self = *(params_t *)params;
65
66     for(i = 0; i < 3; i++) {
67         think(self.position);
68
69         sem_wait(self.lock);
70         sem_wait(&self.forks[self.position]);
71         sem_wait(&self.forks[(self.position + 1) % self.count]);
72         eat(self.position);
73         sem_post(&self.forks[self.position]);
74         sem_post(&self.forks[(self.position + 1) % self.count]);
75         sem_post(self.lock);
76     }
77
78     think(self.position);
79     pthread_exit(NULL);
80 }
81
82 void think(int position)
83 {
84     printf("Philosopher %d thinking...\n", position);
85 }
86
87 void eat(int position)
88 {
89     printf("Philosopher %d eating...\n", position);
90 }
91 }
```

Command line arguments:

```
Philosopher 1 thinking...
Philosopher 1 eating...
Philosopher 1 thinking...
Philosopher 1 eating...
Philosopher 1 thinking...
Philosopher 1 eating...
Philosopher 1 thinking...
Philosopher 2 thinking...
Philosopher 2 eating...
Philosopher 2 thinking...
Philosopher 2 eating...
Philosopher 2 thinking...
Philosopher 2 eating...
Philosopher 2 thinking...
Philosopher 3 thinking...
Philosopher 3 eating...
Philosopher 3 thinking...
Philosopher 3 eating...
Philosopher 3 thinking...
Philosopher 3 eating...
Philosopher 3 thinking...
Philosopher 4 thinking...
Philosopher 4 eating...
Philosopher 4 thinking...
Philosopher 4 eating...
Philosopher 4 thinking...
Philosopher 4 eating...
Philosopher 4 thinking...
Philosopher 0 thinking...
Philosopher 0 eating...
Philosopher 0 thinking...
Philosopher 0 eating...
Philosopher 0 thinking...
Philosopher 0 eating...
Philosopher 0 thinking...
```

...Program finished with exit code 0  
Press ENTER to exit console.