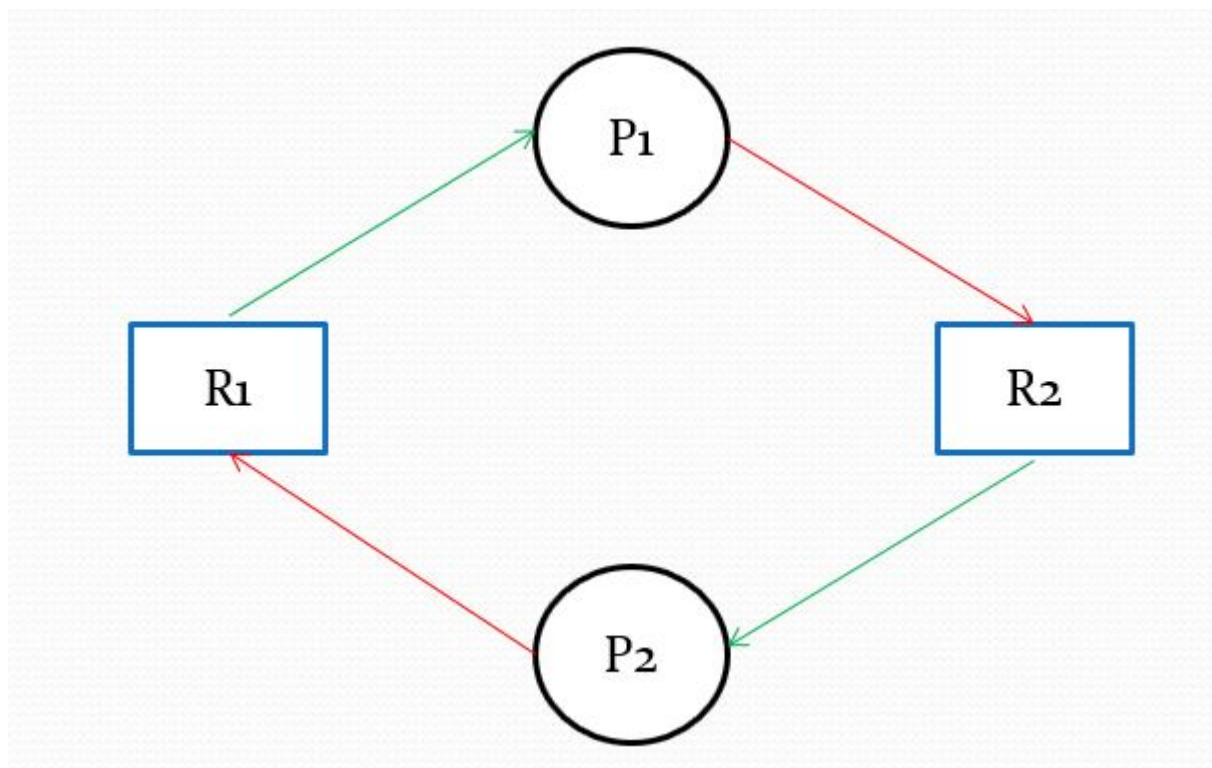


Program to create Deadlock Using C in Linux

[Leave a Comment](#) / [Programs](#) / By [Baljit Singh Saini](#)


Deadlock in operating system is a situation which occurs when a process or thread enters a waiting state because a resource requested is being held by another waiting process, which in turn is waiting for another resource held by another waiting process. In a deadlock state a process is unable to change its state(waiting) indefinitely because the resources requested by it are being used by another waiting process.

Setup



To simulate deadlock in the system we will create the above shown situation.

✓ P1 and P2 will be represented by two thread **one** and **two**.



Join 30 Million U

Enhance your writing

second_mutex while the second thread, thread **two** will try to acquire lock **first_mutex**. Since the resources are already occupied by the other thread, both the threads will get into a deadlock.

Note: You must know how to [create Threads](#) to understand this program

Program to create Deadlock Using C in Linux using Mutex Locks and threads

```
#include<stdio.h>
#include<pthread.h>
#include<unistd.h>
void *function1();
void *function2();
pthread_mutex_t first_mutex; //mutex lock
pthread_mutex_t second_mutex;

int main() {
    pthread_mutex_init(&first_mutex,NULL); //initialize the lock
    pthread_mutex_init(&second_mutex,NULL);
    pthread_t one, two;
    pthread_create(&one, NULL, function1, NULL); // create thread
    pthread_create(&two, NULL, function2, NULL);
    pthread_join(one, NULL);
    pthread_join(two, NULL);
    printf("Thread joined\n");
}

void *function1( ) {
    pthread_mutex_lock(&first_mutex); // to acquire the
resource/mutex lock
    printf("Thread ONE acquired first_mutex\n");
    sleep(1);
    pthread_mutex_lock(&second_mutex);
    printf("Thread ONE acquired second_mutex\n");
    pthread_mutex_unlock(&second_mutex); // to release the
resource
    printf("Thread ONE released second_mutex\n");
    pthread_mutex_unlock(&first_mutex);
    printf("Thread ONE released first_mutex\n");
}
```



```
sleep(1);  
pthread_mutex_lock(&first_mutex);  
printf("Thread TWO acquired first_mutex\n");  
pthread_mutex_unlock(&first_mutex);  
printf("Thread TWO released first_mutex\n");  
pthread_mutex_unlock(&second_mutex);  
printf("Thread TWO released second_mutex\n");  
  
}
```

Practice Programs on Program to create Deadlock Using C in Linux

Q1. Write a program to simulate deadlock between three threads.

Q2. Write a program to create 4 threads (thread1, thread2, thread3 and thread4). Create a deadlock situation between thread2 and thread4.

Viva Questions on Program to create Deadlock Using C in Linux

Q1. What is deadlock?

Q2. What is the minimum number of threads/process required for deadlock to occur?

Q3. What is the significance of pthread_mutex_lock() function?

Q4. Why the sleep() function is used in the program?

Video Link



Deadlock in OS || Program to simulate Deadlock || C Program

[← Previous Post](#)[Next Post →](#)

Leave a Comment

You must be [logged in](#) to post a comment.



OS Lab

Program to create Threads in



Linux Essentials Series

Using ls command in Linux !! I



00:00

13:14

OS Theory

Uniprogramming vs Multiprog



00:00

13:30





TATA MOTORS
Connecting Aspirations

THE 2023 TIAGO

Now Smarter. Efficient. Safer.

Now with **3 Years / 1 Lakh kms Warranty** | **Improved Mileage**

PRICE STARTS AT ₹5.54 LAKH*

*Ex-Showroom Price of Tiago XE Petrol, New Delhi

Categories

[Database](#)

[Entity-Relationship\(ER\)](#)

[Introduction to Database](#)

[Linux](#)

[commands](#)



[Disk Management](#)

[File Management](#)

[Introduction](#)

[IPC](#)

[Memory Management](#)

[Practice Problems](#)

[Process](#)

[Process Synchronization](#)

[System calls](#)

[Thread](#)

[Programs](#)

[Question Hub](#)

[UGC-NET MCQ's for OS](#)

[Uncategorized](#)

Copyright © 2023 Dextutor | Powered by [Astra WordPress Theme](#)

