

**19. Design a C program to implement process synchronization using mutex locks.**

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

pthread_mutex_t lock; // mutex lock

void* process(void* arg) {
    pthread_mutex_lock(&lock); // enter critical section
    printf("Process %d is entering critical section\n", *(int*)arg);
    sleep(1); // simulate some work
    printf("Process %d is leaving critical section\n", *(int*)arg);
    pthread_mutex_unlock(&lock); // exit critical section
    return NULL;
}

int main() {
    pthread_t t1, t2, t3;
    int a = 1, b = 2, c = 3;
    pthread_mutex_init(&lock, NULL); // initialize mutex
    pthread_create(&t1, NULL, process, &a);
    pthread_create(&t2, NULL, process, &b);
    pthread_create(&t3, NULL, process, &c);
    pthread_join(t1, NULL);
    pthread_join(t2, NULL);
    pthread_join(t3, NULL);
    pthread_mutex_destroy(&lock); // destroy mutex
    return 0;
}
```

## OUTPUT-

```
#include <unistd.h>
pthread_mutex_t lock; // mutex lock
void* process(void* arg) {
    pthread_mutex_lock(&lock); // enter critical section
    printf("Process %d is entering critical section\n", *(int*)arg);
    sleep(1); // simulate some work
    printf("Process %d is leaving critical section\n", *(int*)arg);
    pthread_mutex_unlock(&lock); // exit critical section
    return NULL;
}
int main() {
    pthread_t t1, t2, t3;
    int a = 1, b = 2, c = 3;
    pthread_mutex_init(&lock, NULL); // initialize mutex
    pthread_create(&t1, NULL, process, &a);
    pthread_create(&t2, NULL, process, &b);
    pthread_create(&t3, NULL, process, &c);
    pthread_join(t1, NULL);
    pthread_join(t2, NULL);
    pthread_join(t3, NULL);
    pthread_mutex_destroy(&lock); // destroy mutex
    return 0;
}
```

```
Process 1 is entering critical section
Process 1 is leaving critical section
Process 2 is entering critical section
Process 2 is leaving critical section
Process 3 is entering critical section
Process 3 is leaving critical section
-----
Process exited after 3.133 seconds with return value 0
Press any key to continue . . . |
```