Programiz
C Online Compiler

main.c

Share | Run

```c
#include <stdio.h>
#include <pthread.h>
#include <semaphore.h>
#define MAX 5
sem_t mutex;
int counter = 0;
void* process(void* arg) {
    sem_wait(&mutex);
    counter++;
    printf("Process %d: Counter = %d\n", *(int*)arg, counter);
    sem_post(&mutex);
    return NULL;
}
int main() {
    pthread_t threads[MAX];
    int process_ids[MAX];
    sem_init(&mutex, 0, 1);
    for (int i = 0; i < MAX; i++) {
        process_ids[i] = i + 1;
        pthread_create(&threads[i], NULL, process, (void
            *)&process_ids[i]);
    }
```

Output | Clear

```
Process 3: Counter = 1
Process 2: Counter = 2
Process 4: Counter = 3
Process 1: Counter = 4
Process 5: Counter = 5


=== Code Execution Successful ===
```

Programiz
C Online Compiler

main.c

Share | Run

```c
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>

void* threadFunction(void* arg) {
    printf("Thread is running\n");
    return NULL;
}

int main() {
    pthread_t thread;
    pthread_create(&thread, NULL, threadFunction, NULL);
    pthread_join(thread, NULL);
    return 0;
}
```

Output | Clear

```
Thread is running


=== Code Execution Successful ===
```