**EXPERIMENT – 19**

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

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#define NUM\_THREADS 5

#define NUM\_ITERATIONS 100000

int counter = 0;

pthread\_mutex\_t lock;

void\* incrementCounter(void\* arg) {

for (int i = 0; i < NUM\_ITERATIONS; i++) {

pthread\_mutex\_lock(&lock);

counter++;

pthread\_mutex\_unlock(&lock);

}

return NULL;

}

int main() {

pthread\_t threads[NUM\_THREADS];

if (pthread\_mutex\_init(&lock, NULL) != 0) {

printf("Mutex initialization failed.\n");

return 1;

}

for (int i = 0; i < NUM\_THREADS; i++) {

pthread\_create(&threads[i], NULL, incrementCounter, NULL);

}

for (int i = 0; i < NUM\_THREADS; i++) {

pthread\_join(threads[i], NULL);

}

printf("Final counter value: %d\n", counter); // Should be NUM\_THREADS \* NUM\_ITERATIONS

pthread\_mutex\_destroy(&lock);

return 0;

}

SAMPLE OUTPUT:

Final counter value: 500000