EXPERIMENT 19

#include<pthread.h>

#include<stdio.h>

#include<unistd.h>

void \*fun1();

void \*fun2();

int shared=1;

pthread\_mutex\_t l;

int main()

{

pthread\_mutex\_init(&l, NULL);

pthread\_t thread1, thread2;

pthread\_create(&thread1, NULL, fun1, NULL);

pthread\_create(&thread2, NULL, fun2, NULL);

pthread\_join(thread1, NULL);

pthread\_join(thread2,NULL);

printf("Final value of shared is %d\n",shared);

}

void \*fun1()

{

int x;

printf("Thread1 trying to acquire lock\n");

pthread\_mutex\_lock(&l);

printf("Thread1 acquired lock\n");

x=shared;

printf("Thread1 reads the value of shared variable as %d\n",x);

x++;

printf("Local updation by Thread1: %d\n",x);

sleep(1);

shared=x;

printf("Value of shared variable updated by Thread1 is: %d\n",shared);

pthread\_mutex\_unlock(&l);

printf("Thread1 released the lock\n");

}

void \*fun2()

{

int y;

printf("Thread2 trying to acquire lock\n");

pthread\_mutex\_lock(&l);

printf("Thread2 acquired lock\n");

y=shared;

printf("Thread2 reads the value as %d\n",y);

y--;

printf("Local updation by Thread2: %d\n",y);

sleep(1);

shared=y;

printf("Value of shared variable updated by Thread2 is: %d\n",shared);

pthread\_mutex\_unlock(&l);

printf("Thread2 released the lock\n");

}