**THREAD**

#include <stdio.h>

#include <pthread.h>

#include <semaphore.h>

#include <unistd.h>

// Global variables

int x = 0;

sem\_t s1, s2;

void\* thread1\_func(void\* arg) {

sem\_wait(&s1); // wait(s1)

x = x + 1;

printf("T1 prints: %d\n", x);

sem\_wait(&s2); // wait(s2)

sem\_post(&s1); // signal(s1)

pthread\_exit(NULL);

}

void\* thread2\_func(void\* arg) {

sem\_wait(&s1); // wait(s1)

x = x + 1;

printf("T2 prints: %d\n", x);

sem\_post(&s2); // signal(s2)

sem\_post(&s1); // signal(s1)

pthread\_exit(NULL);

}

int main() {

pthread\_t t1, t2;

// Initialize semaphores

sem\_init(&s1, 0, 1); // s1 = 1

sem\_init(&s2, 0, 0); // s2 = 0

// Create threads

pthread\_create(&t1, NULL, thread1\_func, NULL);

pthread\_create(&t2, NULL, thread2\_func, NULL);

// Wait for threads to finish

pthread\_join(t1, NULL);

pthread\_join(t2, NULL);

// Destroy semaphores

sem\_destroy(&s1);

sem\_destroy(&s2);

return 0;

}

Run as

File name = threads\_example.c

gcc threads\_example.c -o threads\_example -lpthread

./threads\_example