19.shared memory

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <semaphore.h>

#define SHARED\_MEMORY\_INITIAL\_VALUE 7

int shared\_memory = SHARED\_MEMORY\_INITIAL\_VALUE;

sem\_t semaphore;

void \*double\_thread(void \*arg) {

int local\_memory;

sem\_wait(&semaphore);

local\_memory = shared\_memory;

printf("Doubled value: %d\n", local\_memory \* 2);

sem\_post(&semaphore);

return NULL;

}

void \*five\_times\_thread(void \*arg) {

int local\_memory;

sem\_wait(&semaphore);

local\_memory = shared\_memory;

printf("Five times value: %d\n", local\_memory \* 5);

sem\_post(&semaphore);

return NULL;

}

int main() {

pthread\_t thread1, thread2;

sem\_init(&semaphore, 0, 1); // Initialize semaphore to 1

pthread\_create(&thread1, NULL, double\_thread, NULL);

pthread\_create(&thread2, NULL, five\_times\_thread, NULL);

pthread\_join(thread1, NULL);

pthread\_join(thread2, NULL);

sem\_destroy(&semaphore); // Destroy semaphore

return 0;

}

Output

