Ejercicios de concurrencia

Ejercicio

Implementar un programa que ejecute 2 threads de forma tal que uno imprime por pantalla los números pares y otro imprime por pantalla los números impares desde 0 a 19. La salida debe estar ordenada y los thread se deben alternar de forma estricta.

Se debe programar usando mutex exclusivamente.

Solución

```
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <semaphore.h>
#include <sys/wait.h>
#include <unistd.h>
#include <sched.h>


int dato_comspartido = 0;
pthread_mutex_t spar, simpar;

void  *par()
{
        int i;
            for (i=0; i<10; i++) {
                pthread_mutex_lock(&spar);
                printf("Thread 1 %d \n", dato_comspartido++);
                pthread_mutex_unlock(&simpar);
            }
}

void  *impar ()
{
        int i;
            for (i=0; i<10; i++) {
                pthread_mutex_lock(&simpar);
                printf("Thread 2 %d \n", dato_comspartido++);
                pthread_mutex_unlock(&spar);
            }
}
```

```
int main(void) {

        pthread_t th1, th2;

        pthread_mutex_init(&spar, NULL);
        pthread_mutex_init(&simpar, NULL);

        pthread_create(&th1, NULL, (void *)par, NULL);
        pthread_create(&th2, NULL, (void *) impar, NULL);

         pthread_join(th1, NULL);
            pthread_join(th2, NULL);

         pthread_mutex_destroy(&spar);
         pthread_mutex_destroy(&simpar);


}
```