

Universidad Carlos III de Madrid
Departamento de Informática
Curso de Sistemas Operativos

Ejercicio

Realizar un programa que cree 2 threads. El primero de ellos debe escribir los números impares y el otro los pares, desde el 1 al 20 respectivamente.

En pantalla deben aparecer los números ordenados por lo que las ejecuciones deben ser alternas.

Se utilizarán **semáforos como mecanismo de sincronización** entre los threads.

Ejemplo de ejecución:

Hijo 1:1

Hijo 2:2

Hijo 1:3

Hijo 2:4

Hijo 1:5

Hijo 2:6

Solución

```
#include <stdio.h>
#include <pthread.h>
#include <semaphore.h>
#include <stdlib.h>
#include <unistd.h>
```

```
int dato_compartido = 0;
sem_t sem1, sem2;
```

```
void uno(void)
{
    int i;
    for (i=0; i<10; i++) {
        sem_wait(&sem1);
        printf("Thread 1 %d \n", dato_compartido++);
        sem_post(&sem2);
    }
}
```

```
void dos(void)
{
    int i;
    for (i=0; i<10; i++) {
        sem_wait(&sem2);
        printf("Thread 2 %d \n", dato_compartido++);
        sem_post(&sem1);
    }
}
```

Universidad Carlos III de Madrid
Departamento de Informática
Curso de Sistemas Operativos

```
int main(void) {  
    pthread_t th1, th2;  
    sem_init(&sem1,0,1);  
    sem_init(&sem2,0,0);  
    pthread_create(&th1, NULL, (void *)uno, NULL);  
    pthread_create(&th2, NULL, (void *)dos, NULL);  
    pthread_join(th1, NULL);  
    pthread_join(th2, NULL);  
    sem_destroy(&sem1);  
    sem_destroy(&sem2);  
}
```

```
Sem_t sem1, sem2;  
int compartilhada;
```

```
int main {
```

```
sem_init (&sem1, 0, 1) ← El 1º esector  
do ponto desbloquear  
sem_init (&sem2, 0, 0) ← espera.
```

```
pthread_t th1, th2;
```

```
pthread_create (&th1, null, (void*) __, null)  
" ( &th2, null, (void*) __, null).
```

```
}
```