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

Ejercicio Realizar un programa que cree 2 hijos utilizando fork (procesos pesados). El primero de ellos debe escribir los números pares (del 2 al 10) y el otro los impares (del 1 al 9).
En pantalla deben aparecer los números ordenados por lo que las ejecuciones deben ser alternas.
Se utilizarán **semáforos como mecanismo de sincronización** entre los procesos.
Ejemplo de ejecución:
Hijo 1:1 Hijo 2:2 Hijo 1:3 Hijo 2:4 Hijo 1:5 Hijo 2:6 Hijo 1:7 Hijo 2:8 Hijo 1:9 Hijo 2:10

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_compartido = 0;
sem_t *sem1, *sem2;

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

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

```
int main(void) {
            int status;

          sem1 = sem_open("mysem1", O_CREAT, O_RDWR, 1);
          if (sem1 == SEM_FAILED) {
                    perror("Failed to open semphore for sem1");
                    exit(-1);
          }
          sem2 = sem_open("mysem2", O_CREAT, O_RDWR , 0);
          if (sem2 == SEM_FAILED) {
                    perror("Failed to open semphore for sem2");
                    exit(-1);
          }

           if (fork() == 0) {
              uno (sem1, sem2);
           } else {
             if (fork() == 0) {
                dos (sem1, sem2);
             } else {
               wait(&status);
              wait(&status);
              sem_close(sem1);
              sem_close(sem2);
             }
          }
}
```