

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

Ejercicio

Realizar un programa que cree 2 procesos y que permita sincronizarlos usando un pipe. El primero de ellos debe crear el pipe antes de crear el proceso hijo. A continuación cada uno de ellos debe imprimir “Soy el padre” “Soy el hijo” de forma sincronizada y con este orden, leyendo y escribiendo en el pipe como mecanismo de sincronización.

Solución

```
// Ejercicio de las transparencias sobre el uso de las tuberías
// Implementación de una sección crítica con pipes

#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>

int main(void) {

    int fildes[2]; /* pipe para sincronizar */
    char c; /* caracter para sincronizar */

    pipe(fildes);
    write(fildes[1], &c, 1); /* necesario para entrar en la sección crítica la primera vez */
    if (fork() == 0) { /* proceso hijo */
        for(;;) {
            read(fildes[0], &c, 1); /* entrada sección crítica */
            // Sección crítica
            printf("El hijo entra en sección crítica\n");
            sleep(2); // espero para que se vea que el padre no entra
            printf("El hijo sale de la sección crítica\n");
            write(fildes[1], &c, 1); /* salida sección crítica */
            sleep(random()%2); // Espero para que no siempre entre el mismo
        }
    } else { /* proceso padre */
        for(;;) {
            read(fildes[0], &c, 1); /* entrada sección crítica */
            // Sección crítica
            printf("El padre entra en sección crítica\n");
            sleep(2); // espero para que se vea que el hijo no entra
            printf("El padre sale de la sección crítica\n");
            write(fildes[1], &c, 1); /* salida sección crítica */
        }
    }
}
```

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

```
sleep (random()%2); //Espero para que no siempre entre el  
mismo  
}  
}  
}
```