



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

```
/* Este programa crea 2 hijos (procesos pesados), uno escribe 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
```

```
*/
```

```
//José Manuel Pérez Lobato
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <sys/types.h>
```

```
#include <linux/sem.h>
```

```
void hijo1(int id_conj_sem){
```

```
int i;
```

```
struct sembuf restar = {0,-1,0};
```

```
struct sembuf sumar = {1,1,0};
```

```
for (i=1;i<10;i=i+2){
```

```
    semop (id_conj_sem, &restar, 1);
```

```
    printf ("Hijo 1:%d\n",i);
```

```
    semop (id_conj_sem, &sumar, 1);
```

Tema: ¿????

Este material se comparte con licencia Creative Commons Reconocimiento no Comercial-Compartida-Igual 3.0. España.



```
}  
}  
void hijo2(int id_conj_sem){  
    int i;  
    struct sembuf operacion;  
  
    operacion.sem_flg=0;  
    for (i=2;i<=10;i=i+2){  
        operacion.sem_num=1;  
        operacion.sem_op=-1;  
        semop (id_conj_sem, &operacion, 1);  
        printf ("Hijo 2:%d\n",i);  
        operacion.sem_num=0;  
        operacion.sem_op=1;  
        semop (id_conj_sem, &operacion, 1);  
    }  
}  
  
main (){  
    int pid1, pid2;  
    int id_conj_sem;  
    union semun op_sem;  
  
    if (( id_conj_sem = semget (35, 2, IPC_CREAT|0666))== -1)  
        exit(-1);  
    op_sem.val=1;  
    semctl (id_conj_sem, 0, SETVAL, op_sem);  
    op_sem.val=0;  
    semctl (id_conj_sem, 1, SETVAL, op_sem);  
    if ((pid1=fork())==0)  
        hijo1(id_conj_sem);  
    else  
        if ((pid2=fork())==0)  
            hijo2(id_conj_sem);  
    wait(NULL);  
    wait(NULL);  
    semctl (id_conj_sem, 1, IPC_RMID, op_sem);  
}
```