

Exercise

Write a program that creates two son processes using fork. The first son must print the even numbers (from 2 to 10) and the other one the odd numbers (from 1 to 9). The program must print the numbers on the screen in increasing order. For achieving this, the executions of the two children must alternate. Use semaphores as synchronization mechanism.

Execution example:

```
Son 1:1  
Son 2:2  
Son 1:3  
Son 2:4  
Son 1:5  
Son 2:6  
Son 1:7  
Son 2:8  
Son 1:9  
Son 2:10
```

Solution

```
/* This program creates 2 son processes. The first son prints the even numbers (from 2  
to 10) and the other one the odd numbers (from 1 to 9). The program prints the  
numbers on the screen in increasing order. For achieving this, the executions of the two  
children must alternate.  
*/
```

```
#include <stdio.h>  
#include <stdlib.h>  
#include <sys/types.h>  
#include <linux/sem.h>
```

```
void son1(int id_conj_sem){  
    int i;  
    struct sembuf subtract = {0,-1,0};  
    struct sembuf add = {1,1,0};  
  
    for (i=1;i<10;i=i+2){  
        semop (id_conj_sem, &subtract, 1);  
        printf ("Son 1:%d\n",i);  
        semop (id_conj_sem, &add, 1);  
    }  
}
```

Tema: ¿????

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

```
}  
void son2(int id_conj_sem){  
int i;  
struct sembuf operation;  
  
operation.sem_flg=0;  
for (i=2;i<=10;i=i+2){  
operation.sem_num=1;  
operation.sem_op=-1;  
semop (id_conj_sem, &operation, 1);  
printf ("Son 2:%d\n",i);  
operation.sem_num=0;  
operation.sem_op=1;  
semop (id_conj_sem, &operation, 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)  
son1(id_conj_sem);  
else  
if ((pid2=fork())==0)  
son2(id_conj_sem);  
wait(NULL);  
wait(NULL);  
semctl (id_conj_sem, 1, IPC_RMID, op_sem);  
}
```