## Sincronización de Procesos con Semáforos

### 1. Procesos no sincronizados

Programa practica08-sin-semaforos.c

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
int main (int argc, char *argv[]){
     int i=0, j, pid;
     pid=fork();
     if (pid==-1){
          perror("fork");
          exit(-1);
     if (pid==0){
          for (j=0;j<100;j++){
               write(STDOUT_FILENO, "o", 1);
               sleep(1);
          }
     }
     else {
          for (j=0; j<100; j++){
               write(STDOUT_FILENO,"+",1);
               sleep(1);
          }
     }
```

### ⊗ □ □ walter@walter-desktop: ~/waltersanchez/practica08

Figura 1. Debido a una falta de sincronización entre el proceso padre y su proceso hijo la salida de los procesos se mezclan entre si

El siguiente programa practica08-con-semaforos.c es la version sincronizada del proceso padre y su proceso hijo del programa practica08-sin-semaforos.c, en esta nueva version ninguno de los procesos accede a la sección critica del otro proceso y viceversa, ver figura 2.

### Programa practica08-con-semaforos.c

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>
int main (int argc, char *argv[]){
     struct sembuf semvar;
     int semid;
     int i=0,j,pid;
     semid = semget(IPC PRIVATE,1,IPC CREAT|0600);
     semvar.sem num=0;
     semvar.sem flg=0;
     semctl(semid,0,SETVAL,1);
     pid=fork();
     if (pid==-1){
           perror("fork");
           exit(-1);
     if (pid==0){
           semvar.sem op=-1;
           semop(semid,&semvar,1);
           for (j=0;j<100;j++)
                write(STDOUT FILENO, "o", 1);
                sleep(1);
           }
           semvar.sem op=1;
           semop(semid,&semvar,1);
     else {
           semvar.sem op=-1;
           semop(semid,&semvar,1);
           for (j=0;j<100;j++){
                write(STDOUT FILENO,"+",1);
                sleep(1);
           }
           semvar.sem op=1;
           semop(semid,&semvar,1);
     semctl(semid,0,IPC RMID,0);
}
```

# 

Figura 2. Proceso padre y proceso hijo sincronizados con semaforos.