

Presentació

Aquesta pràctica planteja una sèrie d'activitats amb l'objectiu que l'estudiant pugui aplicar sobre un sistema UNIX alguns dels conceptes introduïts en els primers mòduls de l'assignatura.

L'estudiant haurà de realitzar una sèrie d'experiments i respondre a les preguntes plantejades. També haurà d'escriure un programa en llenguatge C.

Competències

Transversals:

- Capacitat per a adaptar-se a les tecnologies i als futurs entorns actualitzant les competències professionals

Específiques:

- Capacitat d'analitzar un problema en el nivell d'abstracció adequat a cada situació i aplicar les habilitats i coneixements adquirits per a abordar-lo i resoldre'l
- Capacitat de dissenyar i construir aplicacions informàtiques mitjançant tècniques de desenvolupament, integració i reutilització

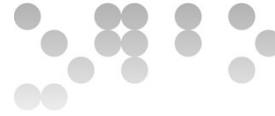
Enunciat

Per a realitzar aquesta pràctica us facilitem el fitxer pr2so.zip amb fitxers font. Descomprimir-ho amb l'ordre `unzip pr2so.zip`. Per a compilar un programa, per exemple, `prog.c`, heu d'executar l'ordre `gcc -o prog prog.c`

1. (4 punts) Donats els següents codis, indiqueu justificadament quin és el resultat d'executar-los. Indiqueu quins paràmetres esperen, quina és la jerarquia de processos creada, quins missatges (i en quina ordre) es mostren pels dispositius d'entrada/sortida, bloquejos...

S'aconsella que en primer terme intenteu determinar el seu comportament sense executar-los en terminal. Executeu-los per a verificar la vostra resposta. Podeu assumir que cap crida retornarà error.

No heu de corregir els codis. Només es demana que expliqueu el seu comportament.



(a) Codi 1 (1 punt):

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

int main(int argc, char *argv[])
{
    int num;
    pid_t t;

    for (num = 0; num < 2; num++) {
        t = fork(); /* 1 */
        if (t == 0)
            break;
    }
    t = fork(); /* 2 */
    printf("Soc el process %d y meu pare es %d\n", getpid(), getppid());

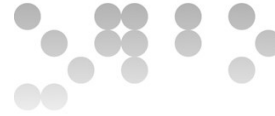
    return 0;
}
```

(a) Codi 2 (1 punt):

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

int main(int argc, char *argv[])
{
    int num;
    pid_t t;

    for (num = 1; num <= 3; num++) {
        t = fork();
        if (num == 3 && t == 0)
            execlp ("ls", "ls", "-l", NULL);
    }
    return 0;
}
```

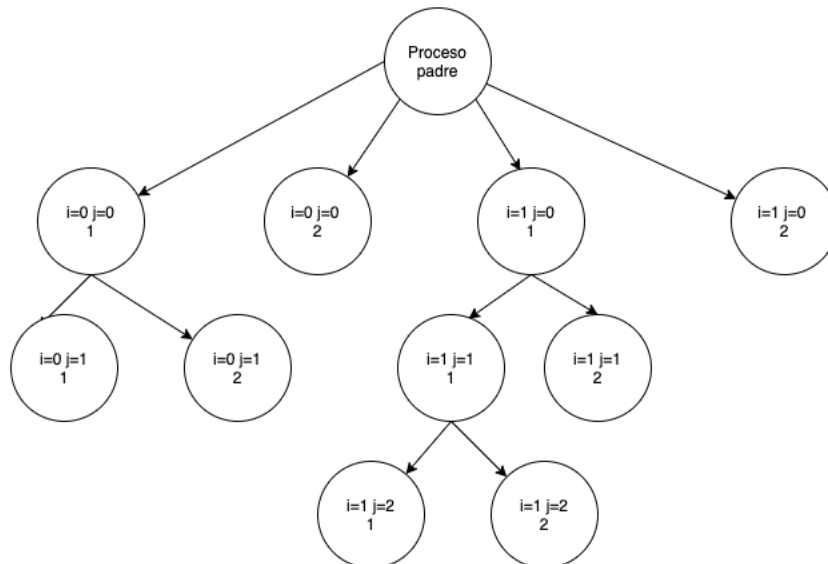


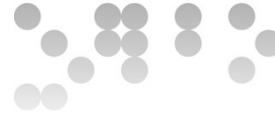
(b) Codi 3 (2 punts):

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

int main(int argc, char *argv[])
{
    int pid, i, j;
    pid_t t;

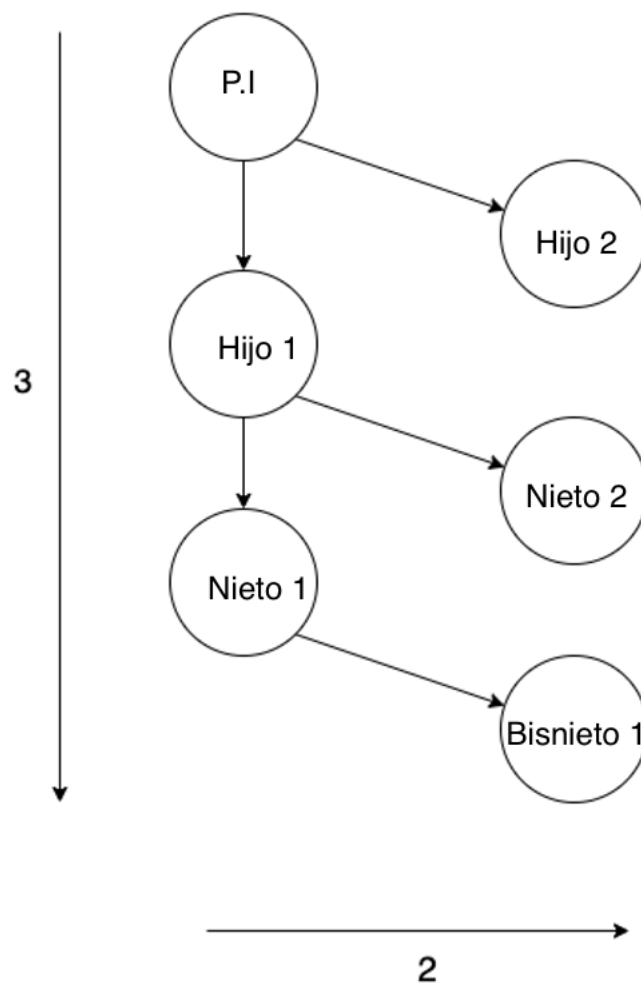
    for (i = 0; i < 2; i++) {
        pid = getpid();
        for (j = 0; j < i+2; j++) {
            t = fork(); /* 1 */
            if (t != 0) {
                t = fork(); /* 2 */
                break;
            }
        }
        if (pid != getpid())
            break;
    }
    return 0;
}
```

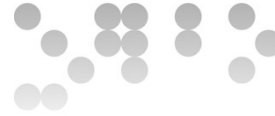




2. (6 punts)

2.1. (3 punts) Completar el codi arbol.c proporcionat l'el zip perquè generi un arbre de processos similar a la següent figura. Els valors de de heights i width de l'arbre seran donats com a paràmetres d'entrada per l'usuari en cada execució. En la figura es considera que el valor de heights equival a 3 i el valor de width 2. En total tindrem un procés inicial, dos processos fills, un dels fills crea una altra vegada dos processos, i , finalment un dels néts crea l'últim procés. Podeu afegir els variables que cosíssi necessari per a implementar el codi.





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

int main(int argc, char *argv[])
{
    int i, height, width;

    if (argc != 3) exit(0);

    height = atoi(argv[1]); /* height */
    width = atoi(argv[2]); /* width */

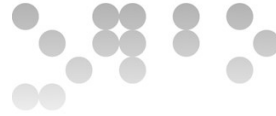
    /* tu codigo aqui */

    printf("Soc el process %d y meu pare es %d\n", getpid(), getppid());
    sleep(1);
    return 0;
}
```

2.2(3 punts)

Escriuiu un programa en C que demani per teclat dues cadenes de caràcters i després escrigui cada cadena per pantalla caràcter a caràcter. L'escriptura de cada cadena haurà de fer-la un procés diferent. S'hauria d'usar read/*write per a llegir l'input i imprimir els caràcters per la pantalla. El procés pare haurà d'esperar que acabi el procés fill.

```
> ./test
Cadena 1:
asd
Cadena 2:
dsa
asddsa
```



Recursos

- Mòduls 1, 2, 3, 4, 5 i 6 de l'assignatura.
- Documento "Introducció a la programació d'Unix" (disponible a l'aula) o qualsevol altre manual similar.
- L'aula "Laboratori de Sistemes Operatius" (podeu plantejar els vostres dubtes relatius a l'entorn UNIX, programació,...).

Criteris d'avaluació

- Es valorarà la justificació de les respostes presentades.
- El pes de cada pregunta aquesta indicat en l'enunciat.

Format i data d'entrega

Es crearà un fitxer zip que contindrà tots els fitxers del lliurament (un pdf amb la resposta a les preguntes i fitxers font).

Data límit de lliurament: 09/12/21