

Problema 1 (3,5 puntos).

Has escrito y compilado en tu directorio de trabajo el siguiente programa para UNIX:

```
ej1.c:
#include <fcntl.h>
main()
{
    int fd, fd1, tam = 8;

    fd = open ( "fichero" , O_WRONLY );
    write ( fd , "linea A\n" , tam - 1 );
    fd1 = creat ( "dir1/fichero" , 0777 );
    write( fd1 , "linea B\n" , tam );
    write( fd , "linea C\n" , tam - 1 );
    close ( fd );
    write ( fd , "linea D\n" , tam - 1 );
    dup ( fd1 );
    write ( fd1 , "linea E\n" , tam );
    write ( fd , "linea F\n" , tam - 1 );
}
```

El estado de tu directorio de trabajo lo conoces después de ejecutar el siguiente comando:

```
merlin_$ ls -l -R
```

Obtienes así un listado (ls) largo (-l) y recursivo (-R) del directorio:

```
merlin_$ ls -l -R
total 2
drwxr-x---  2 user1arqcomp   24   Jun  8 17:56  dir1
-rwxr-x---  1 user1arqcomp 24628 Jun  8 18:24  ej1
-rw-r----- 1 user1arqcomp  337   Jun  8 18:24  ej1.c
-rw-r----- 1 user1arqcomp   0     Jun  8 17:55  fichero

./dir1:
total 0
```

Llamamos a éste estado del directorio: **situación1.**

situación2:

Desde tu directorio de trabajo ejecutas ahora el siguiente comando:

```
merlin_$ ln fichero dir1/fichero
```

A) (0,5 puntos)

Cuál sería el estado de tu directorio de trabajo en la situación2?.

B) (1 punto)

Estando en la situación1 escribe cuál sería el contenido de todos los archivos fichero después de ejecutar ej1 (merlin_\$ ej1).

C) (2 puntos)

Estando en la situación2 escribe cuál sería el contenido de todos los archivos fichero después de ejecutar ej1 (merlin_\$ ej1).

Problema 2 (3,5 puntos).

Tenemos el siguiente programa para UNIX que provoca la ejecución de tres procesos:

ej2.c:

```
1 #include "error.h"
2
3 main()
4 {
5     int pid, pid1, estado;
6
7     if ( ( pid = fork( ) ) < 0 )
8         syserr ( "fork" );
9     else if ( pid == 0 )
10    {
11        if ( ( pid = fork( ) ) < 0 )
12            syserr ( "fork" );
13        else if ( pid )
14            {
15                exit ( 0 );
16            }
17        sleep ( 2 );
18        fprintf ( stderr , "Mi padre tiene PID = %d, pid vale = %d \n" ,
19                getppid( ) , pid );
20        exit ( 0 );
21    }
22    while ( ( pid1 = wait ( &estado ) ) > 0 )
23    {
24        fprintf ( stderr , "Wait devuelve %d\n" , pid1 );
25    }
26    exit ( 0 );
27}
```

A) (0,5 puntos)

Dibuja un esquema donde aparezcan TODOS los procesos involucrados en la ejecución de ej2, indicando sus relaciones de paternidad.

B) (2 puntos)

Escribe cuál crees que sería el contenido de la salida estandar de error (stderr) después de ejecutar ej2.

Debes indicar qué proceso produce cada salida.

Supón para tu respuesta los siguientes datos:

El proceso que ejecuta ej2 tiene PID: 25455

El proceso creado en la línea 7 tiene PID: 25456

El proceso creado en la línea 11 tiene PID: 25457

C) (1 punto)

Modifica ej2.c para que en la salida estandar de error (stderr) aparezan los PID's de los dos procesos creados (el de la línea 7 y el de la línea 11).

Nota: En la modificación de C) NO se puede utilizar la llamada al sistema getpid () y el ej2.c modificado NO debe superar las 28 líneas de código.