

OJO: LO TIENEN LOS ALUMNOS!!!

Problema 1 (3 puntos).

Hemos escrito escrito y compilado en nuestro directorio de trabajo el siguiente programa para UNIX:

mix.c

```
main ( argc , argv )
int argc;
char **argv;
{
    int fd1, fd2, fd3, tam, x;
    char c;

    fd1 = open ( argv[1] , 0 );
    fd2 = open ( argv[2] , 0 );
    tam = atoi ( argv[3] );
    fd3 = creat ( argv[4] , 0777 );
    x = lseek ( fd2 , tam , 0 );

    while ( read ( fd2 , &c , 1 ) > 0 ){
        while ( x > 0 ) {
            read ( fd1 , &c , 1 );
            write ( fd3 , &c , 1 );
            x--;
        };
        write ( fd3 , &c , 1 );
    }
}
```

- Explica brevemente cómo invocarías al programa ejecutable "mix" y cuál sería el resultado.
- Suponiendo que ninguna línea del código de "mix.c" es errónea, indica si el programa anterior puede fallar alguna vez al invocarlo como has contado en el apartado anterior y si es así, cómo lo haría.

Problema 2 (4 puntos).

Diseñar un programa llamado "thead" cuyo funcionamiento describimos a partir del siguiente ejemplo de utilización:

merlin_\$ P1 | thead n F1 | P2

donde P1, P2 y thead son programas y F1 es el nombre de un fichero (pasado como segundo parámetro a "thead").

Como resultado de la ejecución de la línea anterior:

- P2 recibe en su entrada estandar las n primeras líneas procedentes de la salida estandar de P1 (n es pasado como primer parámetro a "thead" y cuando $n = 0$, P2 recibe en su entrada estandar la salida estandar de P1).
- En F1 queda escrita la salida estandar de P1 (para cualquier valor de n).

Se pide:

Apartado 1. Implementar el programa "thead" en lenguaje C, usando sólo llamadas al sistema UNIX para la entrada/salida.

Apartado 2. Utilizando el programa "thead" del **apartado 1**, escribir las líneas de comandos que ejecuten de la forma más eficiente posible lo que se pide a continuación:

- La salida estandar de P1 quede escrita en F1 y F2 y sus n primeras líneas sean usadas como entrada estandar de P2.
- La salida estandar de P1 quede escrita en F1 y sus n primeras líneas aparezcan en pantalla.
- La salida estandar de P1 quede escrita F1 y en F2.

Problema 3 (3 puntos).

Hemos escrito y compilado en nuestro directorio de trabajo el siguiente programa para UNIX:

programa.c:

```
#include <signal.h>

int una_alarma()
{
    write ( 2 , "UNA_ALARMA\n" , 11 );
}

int otra_alarma()
{
    write ( 2 , "OTRA_ALARMA\n" , 12 );
}

main()
{
    int estado;

    signal ( SIGALRM , una_alarma );

    if( fork() ){
        signal ( SIGALRM , otra_alarma );
        wait ( &estado );
    }
    else{
        alarm ( 3 );
        execl ( "/bin/p1" , "p1" , 0 );
        exit ( 1 );
    }
}
```

- Explica qué hace nuestro programa, porqué y cuál sería su resultado final.

