

Material permitido:

Calculadora NO programable.

Tiempo: **2 horas.**

N2

Aviso 1: Todas las respuestas deben estar razonadas.

Aviso 2: Escriba sus respuestas con una letra **lo más clara posible.**

Aviso 3: No use **Tipp-ex** o similares (atasca el escáner).

ESTE EXAMEN CONSTA DE 5 PREGUNTAS

1. (1p). Señala cuál de las siguientes afirmaciones es verdadera y **razona** la respuesta. El algoritmo de planificación usado en BSD4.3:

- a) Permite que un proceso de baja prioridad quede abandonado mientras existan procesos de mayor prioridad.
- b) Establece la prioridad de un proceso mediante la llamada al sistema `priocntl`.
- c) Permite que cualquier proceso aumente su prioridad mediante el uso de la llamada al sistema `nice`.
- d) Es estrictamente no expropiable lo que hace que pueda aparecer inversión de prioridades.

2. (2p) Explique **razonadamente** si las siguientes afirmaciones son verdaderas o falsas:

i) (1p) Si en el directorio actual existe un archivo `uno.txt` con número de nodo-i 4096 que posee una sola referencia y se invocan consecutivamente los comandos `"ln uno.txt dos.txt"` y `"ln -s uno.txt tres.txt"`; el contador de referencias del archivo `tres.txt` vale 1.

ii) (1p) Una mejora de FFS respecto a s5fs es que Intenta asignar bloques secuenciales de un fichero en posiciones rotacionalmente óptimas.

3. (2p) Conteste a las siguientes cuestiones:

a) (0.5 p) Explique **brevemente** el funcionamiento de un intérprete de comandos de UNIX.

b) (1.5 p) Enumere y explique los tipos de órdenes que de forma general se pueden ejecutar en un intérprete de comandos.

4. (2p) En un sistema UNIX la máscara de modo del fichero ordinario resultados es 6644.

a) (1p) Explique razonadamente el significado de todos los bits de la máscara de modo.

b) (0.5 p) Escriba la máscara de modo simbólica asociada a este fichero.

c) (0.5 p) ¿Qué orden se debe teclear desde la línea de comandos (\$) para que la máscara simbólica del fichero pase a ser `-rwx r-- --T`?

PREGUNTA 5 EN LA SIGUIENTE PÁGINA

5. (3 p) Conteste razonadamente a los siguientes apartados:

a) (1.5 p) Explicar el significado de las sentencias enumeradas ([1]) de este programa.

b) (1.5 p) Explicar el funcionamiento del programa ejecutable `prog` resultante de compilar el código siguiente cuando se invoca desde la ventana de comandos: `./prog SO`.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>
#define MAX 256

[1] void main(int argc, char *args[])
    {
        int tuberia[2];
        int pid;
        char mensaje[MAX];
        if (argc!=2)
            {printf("Argumentos incorrectos\n");
             exit(-1);}

[2] if (pipe(tuberia)==-1)
    {
[3]     perror("Error en pipe");
[4]     exit(-1);
    }

[5] if ((pid=fork())==-1)
    {
        perror("Error en fork");
        exit(-1);
    }
    else if (pid==0)
    {
[6]         if (read(tuberia[0], mensaje, MAX)==0)
            {perror("Error de lectura"); exit(-2);}
        printf("Examen %s ",mensaje);
[7]         if (write(tuberia[1],args[1],strlen(args[1])+1)==0)
            {perror("Error de escritura"); exit(-2);}
[8]         close(tuberia[0]);
            close(tuberia[1]);
            exit(0);
    }
    else
    {
        if (write(tuberia[1],"DyA",4)==0)
            {perror("Error de escritura"); exit(-2);}

[9]         wait();
        if (read(tuberia[0], mensaje,MAX)==0)
            {perror("Error de lectura"); exit(-2);}
        printf("%s\n",mensaje);
        close(tuberia[0]);
        close(tuberia[1]);
        exit(0);
    }
}
```