

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

- (1 p) El núcleo puede emplear sus mecanismos de protección para proteger unas hebras de usuario de otras.
- (1 p) El uso de las colas de mensajes en la transferencia de grandes cantidades de datos no afecta al rendimiento del sistema.

2 (1.5 p) Supóngase que el fichero `/etc/group` contiene, entre otras, las siguientes líneas:

```
root*:0:
estudiantes*:10: TUTOR, ESTUDIANTE1
profesores:x:50:DIRECTOR
```

- (0.25 p) ¿Cuántos miembros tiene el grupo `root`?
- (0.25 p) ¿Cuál es el GID del grupo `profesores`?
- (0.25 p) ¿Qué usuarios tienen acceso en el grupo `estudiantes`?
- (0.75 p) ¿Tiene contraseña de entrada el grupo `profesores`? ¿Y el grupo `estudiantes`? En caso afirmativo, ¿dónde está almacenada?

3. (1.5 p) Responda razonadamente a las siguientes cuestiones:

- (0.75 p) ¿Qué son los *callouts*? Relaciona algunos usos de los *callouts*.
- (0.75 p) Describe las formas de implementar la lista de *callouts*.

4. (2 p) Supóngase que la lista parcial de i-nodos libres del superbloque está vacía, su i-nodo recordado es 25 y su variable índice puede tomar como máximo el valor 4. Además, existen los siguientes i-nodos libres en la tabla de i-nodos: 95, 45, 88, 37, 32, 30 y 73. Dibuje la lista parcial de i-nodos libres del superbloque una vez que ha sido rellenada por el núcleo. ¿Cuál sería ahora el i-nodo recordado?

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

- (2 p) Explicar el significado de las sentencias enumeradas ([1]) del programa que se muestra en la página siguiente.
- (1 p) Explicar el funcionamiento del programa si se compila en el ejecutable programa, se invoca con la orden `./programa DyASO` y no se produce ningún error.

La pregunta continua en la página siguiente

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <string.h>
#include <sys/wait.h>
```

```
[1]  int main(int argc, char *argv[])
    {
        int fd, pid, salida, status;

        if (argc!=2) {printf("Argumentos incorrectos\n"); exit(1);}
[2]  fd=open("fichero1.sh", O_RDWR|O_CREAT|O_TRUNC, 0666);
[3]  if (fd==-1) {perror("Error open"); exit(2);}
    write(fd,"#!/bin/bash\nnecho Ejercicio 5 ",29);
[4]  write(fd,argv[1],strlen(argv[1]));
[5]  close(fd);
[6]  chmod("fichero1.sh",0700);

[7]  unlink("fifo");
[8]  if (mknod("fifo",S_IFIFO|0666,0)==-1)
    {perror("Error mknod"); exit(3);}

[9]  if ( (pid=fork())==-1 ) {printf("Error fork\n"); exit(4);}
    if ( pid==0 )
    {
[10]     salida=system("./fichero1.sh > fifo");
        if (salida==0) printf(";Correcto!\n");
        unlink("fifo");
        unlink("fichero1.sh");
    }
    else
    {
[11]     execl("/bin/cat","cat","fifo",NULL);
        perror("Error execl");
        printf(";Incorecto!\n");
        exit(5);
    }
    exit(0);
}
```