

Material permitido:

Calculadora NO programable.

Tiempo: **2 horas.**

N1

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. Responda a las siguientes cuestiones:

a) (0.5 p) Explique el funcionamiento del siguiente comando:

```
grep patrón fichero1 fichero2 ... ficheroN
```

b) (0.5 p) Explique el funcionamiento de la siguiente llamada al sistema y el significado de sus parámetros:

```
resultado = mount(dispositivo, dir, flags);
```

2. (2 p) Describa las acciones que realiza el núcleo cuando un proceso realiza la llamada al sistema `shmget`.

3. (2 p) Dibuje un diagrama, **adecuadamente rotulado**, que esquematice las principales acciones que realiza el núcleo durante la ejecución del algoritmo `exec()`.

4. (2p) Escriba un programa en C que realice la creación de un conjunto de 3 semáforos asociados a la llave creada a partir del fichero “esllave” y la clave ‘J’. El conjunto de semáforos se crea con permisos de lectura y modificación para el usuario.

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

a) (1.5 p) Explique el significado de las sentencias enumeradas ([]) del programa que se muestra en la página siguiente.

b) (1.5 p) Explique el funcionamiento del programa explicando detalladamente su salida asumiendo que no hubiese otros programas que compitan con él por el uso de CPU.

La pregunta continua en la página siguiente

```
#include <signal.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <time.h>
#include <sys/times.h>
```

```
void despertar(int sig);
void continuar(int sig);
void acabar(int sig);
```

```
void main(void)
{
```

```
    int pid;
[1]    signal(SIGCHLD,despertar);
    signal(SIGCONT,continuar);
    signal(SIGTERM,acabar);
[2]    if ((pid=fork())==-1){
        perror("Error en fork:");
        exit(1);
    }
    if (pid!=0) {
[3]        pause();
        printf("del examen DyASO ");
[4]        while(1);
        printf("erroneo.\n");
[5]        exit(0);
    }
    else{
[6]        sleep(1);
        printf("Ejercicio 5 ");
[7]        kill(getppid(),SIGCONT);
        sleep(1);
    }
}
```

```
void despertar(int sig)
```

```
{
    struct tms T;
    time_t cpu;
[8]    times(&T);
    printf("correcto");
    cpu=T.tms_utime + T.tms_stime + T.tms_cutime + T.tms_cstime;
    printf(" tras %ld ms de ejecución activa.\n",
        (1000*cpu)/sysconf(_SC_CLK_TCK));
[9]    raise(SIGTERM);
}
```

```
void continuar(int sig) {}
void acabar(int sig) {exit(0);}
```