

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. (2p). Explique razonadamente si las siguientes afirmaciones son VERDADERAS o FALSAS

- i. (1p) La actuación de los descriptores de fichero reduce el rendimiento del sistema.
- ii. (1p) Cuando se produce una interrupción que no esté bloqueada o enmascarada, el núcleo añade una capa de contexto en la pila de capas de contexto de un proceso.

2. (2p). Enumere y describa las características comunes de los mecanismos IPC del System V.

3. (1p). Describe el resultado de la siguiente operación:

```
mutsem=semget(2471, 3, IPC_CREAT | 0666);
```

4. (2p). El ladrón de páginas de un cierto sistema UNIX debe transferir a un dispositivo de intercambio 25 páginas del proceso A, 100 páginas del proceso B, 50 páginas del proceso C y 75 páginas del proceso D. En una operación de escritura puede transferir 30 páginas al dispositivo de intercambio.

- a) (1.5 p). Determinar la secuencia de operaciones de intercambio de páginas que tendría lugar si el ladrón de páginas examina las páginas de los procesos en el orden C, B, D y A.
- b) (0.5 p). ¿Cuál es el último proceso que transfiere completamente todas sus páginas al dispositivo de intercambio?

(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 suponiendo que al primer proceso creado al ejecutarlo se le asigna el pid 1000, los pids se asignan en secuencia, no se produce ningún error y los identificadores textuales de las señales de usuario son "User defined signal 1" y "User defined signal 2" respectivamente.

```
#include <signal.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
void fun1(int sig)
{
    printf("%d recibe: %s \n",getpid(),strsignal(sig));
}
void main()
{
    int pid;
    signal(SIGUSR1,fun1);
    signal(SIGUSR2,fun1);
[1]    if ((pid=fork())==-1) { perror("Error en fork():"); exit(2);}
    if (pid==0)
    {
[2]        printf("Primera ronda\n");
[3]        signal(SIGUSR1,SIG_IGN);
[4]        kill(getppid(),SIGUSR1);
        sleep(1);

        printf("Segunda ronda\n");
[5]        sigblock(sigmask(SIGUSR2));
        signal(SIGUSR1,fun1);
        kill(getppid(),SIGUSR1);
        sleep(1);

        printf("Tercera ronda\n");
[6]        sigsetmask(0);
        sleep(1);
        exit(3);
    }
    else
    {
        pause();
        kill(pid,SIGUSR1);
        kill(pid,SIGUSR2);
[7]        pause();
        kill(pid,SIGUSR1);
        kill(pid,SIGUSR2);
[8]        wait();
        exit(3);
    }
}
```