

Material permitido:

Calculadora NO programable.Tiempo: **2 horas.**

R

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:

- a) (1p) En el UNIX SVR3 el único caso en que se puede producir un fallo de protección es cuando un proceso intenta acceder a una página cuyos bits de protección no permiten acceder a la página.
- b) (1p) El `nodo-im` permite el acceso rápido a la información del archivo.

2. (1.5 p) Explica qué son los *shell scripts*.

3. (2 p) Explique **razonadamente** qué es la interfaz *nodo-v/sfv* de UNIX y qué ventajas proporciona su uso.

4. (1.5 p) Describa el funcionamiento del siguiente programa escrito en C:

```
        #include <stdio.h>
        #include<unistd.h>
        main()
        {
[1]    if (fork()==0)
            {
[2]        execl("/bin/date", "date", NULL);
            }
[3]        printf("\nFinalizar\n");
        }
```

(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 y mostrar la salida en pantalla asumiendo que la llamada al sistema `fork()` devuelve 0 y 4141.

```
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>

void main(void)
{
    int pid, shmid, par, estado;
    char *cadena;
    key_t llave;
[1]    if (creat("./archivo", 0600) == -1) exit(1);
[2]    llave = ftok("archivo", 'M');
[3]    shmid = shmget(llave, 50 * sizeof(char), IPC_CREAT | 0600);
    if (shmid == -1) exit(1);
[4]    cadena = shmat(shmid, 0, 0);
[5]    if ((pid = fork()) == -1) exit(2);
    if (pid == 0)
    {
        sleep(2); // espero un poco antes de escribir
        strcpy(cadena, "hola");
[6]        shmdt(cadena);
        exit(3);
    }
    else
    {
[7]        par = wait(&estado);
        printf("El proceso %d dice %s \n", pid, cadena);
    }
}
```