

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

- I) (0,75 p) En el planificador del UNIX BSD4.3 la prioridad de planificación de un proceso es un valor entre 0 y 127. Los procesos en modo usuario tienen las prioridades entre 60 y 127.
- II) (0.75 p) Una de las principales desventajas de los enlaces simbólicos es que el análisis de las rutas de acceso es más lento que en el caso de los enlaces duros.

2. (2p) Enumere **brevemente** la información que se almacena en un nodo-i de un sistema de archivos s5fs de UNIX.

3. (2p) Explique **razonadamente** para qué sirven los siguientes comandos de UNIX:

- a) (0.5 p) Comando `ps`
- b) (0.5 p) Comando `top`
- c) (0.5 p) Comando `jobs`
- d) (0.5 p) Comando `ls`

4. (1.5 p) Dibuje un diagrama, **adecuadamente rotulado**, que esquematice las principales acciones que realiza el núcleo durante la ejecución de la rutina asociada a la llamada al sistema `fork()`.

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

- a) (1.5p) Explique el significado de las sentencias enumeradas ([]) del código que se muestra en la página siguiente.
- b) (1.5p) Si el programa se compila y el resultado es el ejecutable `prog`. Explique detalladamente el funcionamiento si se invoca desde el intérprete de comandos (\$) mediante la orden:

```
$ prog DyA_Sistemas_Operativos
```

La pregunta 5 continua en la página siguiente

```

#include <stdlib.h>
#include <stdio.h>
#include <string.h>

[1] main(int b, char *c[])
{
    int pfd[2];
    int a;
    char buf;
    if (b == 2)
    {
[2]         if (pipe(pfd) == -1){
                fprintf(stderr, "E1\n");
                exit(-1);
            }
[3]         a=fork();
            if (a == 0) {
[4]                 close(pfd[1]);
[5]                 while (read(pfd[0], &buf, 1) > 0) write(1, &buf, 1);
                write(1, "\n", 1);
                close(pfd[0]);
            }
            else {
                printf("\nMensaje:\n");
                close(pfd[0]);
                write(pfd[1], c[1], strlen(c[1]));
                close(pfd[1]);

[6]         wait();
            }
        }
        else {
            fprintf(stderr, "E2\n");
            exit(-1);
        }

[7]         exit(0);
    }
}

```