

Material permitido:

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

N

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). Señale **razonadamente** si las siguientes afirmaciones son verdaderas o falsas:

- I. (1p) La rutina `splttty()` invocada cuando se produce una interrupción por parte de un terminal, disminuye el `npi` al valor de `npi` anteriormente almacenado.
- II. (1 p) En UNIX los atributos de un fichero se encuentran almacenados en la entrada del directorio al que pertenece.

2. (1.5 p). Conteste **razonadamente** a las siguientes cuestiones:

- a) (0.5 p) ¿Qué es la tabla de regiones por proceso?
- b) (1 p) Señale al menos cinco operaciones con regiones implementadas por el núcleo

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

- a) (0.5 p) ¿Qué parámetros de entrada requiere el algoritmo `sleep()`?
- b) (1 p) Dibuje un diagrama con las principales acciones que realiza el núcleo durante la ejecución del algoritmo `sleep()`.

4. (2 p) Considérese un sistema GNU/Linux, con la tabla de montaje ubicada en `/etc/fstab`. El contenido de este archivo es el que se muestra a continuación:

#	device	directory	type	options
	/dev/hda1	/	ext2	defaults
	/dev/hda2	/usr	ext2	defaults
	/dev/hda3	none	swap	sw
	/dev/sda1	/dosc	msdos	defaults
	/proc	/proc	proc	none

Describa la información que se obtiene a partir de esta contenido.

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

- a) (1.5 p) Explique brevemente el significado de las sentencias enumeradas ([]) del programa mostrado en la página siguiente.
- b) (1.5 p) El programa es compilado produciendo el fichero ejecutable `Ejercicio`. Explique la ejecución del programa y su salida si se invoca desde la línea de comandos la orden `./Ejercicio cinco`, y no se produce ningún error.

Nota: `fflush` sirve para vaciar el buffer de salida estándar haciendo que la salida de `printf` se imprima inmediatamente.

La pregunta continua en la página siguiente

```

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <wait.h>
#include <sys/msg.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
void main(int argc, char* argv[]){
    int pid, msqid, shmid, par, estado;
    int tam=(strlen(argv[1])+1)*sizeof(char);
    char* texto;
    key_t llave;
[1]    llave=ftok(argv[0], 'X');
[2]    if (argc!=2) {printf("Argumentos incorrectos\n"); exit(1);}

[3]    shmid=shmget(llave,100*sizeof(char),IPC_CREAT | 0600);
    if (shmid==-1){perror("Error en shmget");exit(2);}
[4]    texto=shmat(shmid,0,0);

[5]    msqid=msgget(llave, IPC_CREAT | 0600);
    if (msqid==-1){perror("Error en msgget");exit(3);}
    struct{
        long tipo;
        char cadena[tam];
    } mensaje;
    int longitud=sizeof(mensaje)-sizeof(mensaje.tipo);

[6]    if ((pid=fork())==-1) {perror("Error en Fork"); exit(4);}
    if (pid==0){
[7]        msgrcv(msqid, &mensaje,longitud,1,0);
        printf("%s ",mensaje.cadena);
        fflush(stdout);
        strcpy(texto,"correcto.");
[8]        shmdt(texto);
    }
    else {
[9]        sleep(1);
        strcpy(texto,"mal resuelto.");
        printf("%s ",argv[0]);
        fflush(stdout);
        mensaje.tipo=1;
        strcpy(mensaje.cadena,argv[1]);
[10]    msgsnd(msqid, &mensaje,longitud,0);
[11]    par=wait(&estado);
        printf("%s\n",texto);
[12]    msgctl(msqid, IPC_RMID,0);
        shmdt(texto);
    }
}

```