

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. (1p) Señala qué máscara representa un fichero regular donde únicamente el propietario del fichero puede leer, escribir y ejecutar el fichero; y además el bit `S_ISUID` está activado y los bits `S_ISGID` y `S_ISVTX` están activados.

2. (2p) Explique **razonadamente** si las siguientes afirmaciones son verdaderas o falsas:

i) (1p) Una tubería sirve para transmitir datos a múltiples procesos de forma simultánea.

ii) (1p) Se tiene un computador con una memoria principal de capacidad $C_{MP} = 16$ Mbytes y un tamaño de página $S_P = 1$ Kbyte. El número total de marcos de página es 8192.

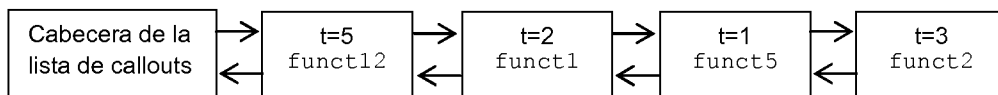
3. (2 p) Conteste **razonadamente** a las siguientes preguntas:

a) (0.5 p) ¿Qué es una región de un proceso?

b) (0.5 p) ¿Cuáles son las principales regiones en que se descompone un proceso?

c) (1 p) ¿Cuál es el contenido y las características de las regiones en que se descompone un proceso?

4. (2p) En la Figura se muestra la lista de *callouts* del núcleo del UNIX BSD en un cierto instante de tiempo.



Se pide:

a) (0.5 p) Explicar brevemente que es un *callout*.

b) (0.75 p) Determinar el tiempo de disparo (en tics) de `func1`, `func2`, `func5` y `func12`.

c) (0.75 p) Supuesto que ha transcurrido un tic, dibujar la lista de *callout*.

(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 describir su salida asumiendo que es ejecutado por el administrador del sistema, en el directorio actual existe un directorio *dir1* y dentro de él un único archivo *prueba.txt*, y además que la raíz del sistema de archivos es de tipo *ext4* y se encuentra alojada en el dispositivo *sda1*.

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/mount.h>
#include <unistd.h>

void main(void)
{
    int res,res2,pid;
[1]    if ((res=mount("/dev/sda1","./dir1","ext4",0,NULL))==0)
        {
[2]            if ((pid=fork())==0)
                {
[3]                printf("\nPrimer ls:\n");
[4]                execl("/bin/ls","ls","dir1",NULL);
                perror("fallo en execl");}
            else
                {
[5]                wait();
[6]                sleep(1);
[7]                if (umount("./dir1")==1) perror("fallo en umount");
                printf("\nSegundo ls:\n");
[8]                system("ls dir1");
                printf("Fin.\n");
                }
        }
    else
    {
        perror("Error en mount");
    }
}
```