

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

a) (1 p). En UNIX; para visualizar pantalla a pantalla el contenido del fichero `host.conf` situado dentro del directorio `/etc`, se debe teclear la orden

```
$ cat /etc/host.conf
```

b) (1 p) Un proceso puede incurrir en un fallo de validez cuando intenta escribir una página cuyo bit *copiar al escribir* esté activado.

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

a) (1 p) ¿Qué es el superbloque?

b) (0.5 p) Explique cómo procede núcleo cuando la lista parcial de nodos-i libres del superbloque se vacía.

3. (1.5 p) Describa qué es un cerrojo con bucle de espera y explique cómo funciona.

4. (2 p) Supóngase que el ladrón de páginas de un cierto sistema UNIX debe transferir a un dispositivo de intercambio 50 páginas del proceso A, 30 páginas del proceso B, 40 páginas del proceso C, y 60 páginas del proceso D; y que en una operación de escritura puede transferir 56 páginas al dispositivo de intercambio. Determinar la secuencia de operaciones de intercambio de operaciones que tendría lugar si el ladrón de páginas examina las páginas de los procesos en el orden A, B, C y D.

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

a) (1.5 p) Explique 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 `prog`. Explique la ejecución del programa su salida si se invoca desde la línea de comandos la orden `./prog DyASO`

La pregunta continua en la página siguiente

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>

void main(int argc, char *argv[])
{
    int i, pid, fd;
    char texto[]="OSAyD 5 oicicrejE";
    int tam=strlen(texto)-1;

    if (argc!=2){printf("Argumentos incorrectos\n"); exit(-1);}
[1] unlink(argv[1]);
[2] if (mknod(argv[1],S_IFIFO|0666,0)==-1){perror("Error mknod"); exit(-2);}
[3] if((fd=open(argv[1],O_RDWR))== -1){perror("Error open"); exit(-3)};
    if((pid=fork())== -1) {perror("error en primer fork"); exit(-4);}
    if(pid==0)
    {
        for(i=0;i<=tam;i++)
        {
[4]         if ((pid=fork())== -1) {perror("error en fork 2"); exit(-3);}

            if(pid==0)
            {
[5]                 if(i==tam) {close(fd);break}
            }

            else
            {
[6]                 wait();
[7]                 sleep(1);
                write(fd,&texto[i],1);
                if (i==0) write(fd,"\n",1);
                close(fd);
                break;
            }
        }
    }
    else
    {
[8]         close(fd);
        execl("/bin/cat", "cat", "argv[1]", NULL);
        printf("Incorrecto\n");
    }
}

```