



NOMBRE Y APELLIDOS: _____ NIA: _____

**Examen de Sistemas Operativos
11 de Septiembre de 2010**

NOTAS:

- * Para la realización del presente examen se dispondrá de **2 horas**.
 - * **No** se pueden utilizar libros **ni** apuntes, ni usar móvil (o similar).
 - * Responda cada pregunta en hojas distintas.
-

Teoría . (2,5 ptos)

a.- ¿Qué es la MMU ? Indica brevemente que función realiza

b.- Indica 3 de las responsabilidades del SO respecto a la E/S y el almacenamiento secundario

c.-. En que consiste la semántica UNIX de coutilización de archivos



NOMBRE Y APELLIDOS: _____ NIA: _____

**Examen de Sistemas Operativos
11 de Septiembre de 2010**

Ejercicio 2. (2,5 ptos)

Realizar una aplicación en lenguaje C que conste de 1 padre y 2 hijos.

- * El padre crea los 2 hijos y después lee números de teclado hasta que lea el 0
- * Los números pares se los pasa al hijo 0 y los impares al hijo 1 mediante pipes.
- * Cuando lea el 0 lo enviará a ambos hijos y esperará la finalización de éstos antes de terminar la ejecución.
- * Los hijos sumarán los números recibidos del padre y cuando lean el 0 mostrarán la suma total y finalizarán su ejecución

Suponga que las librerías necesarias ya se han importado mediante los *include* necesarios

Ejercicio 3. (2,5 ptos)

Escribe en lenguaje de programación C un programa que :

- Cree 20 threads que simulan los vehículos que desean entrar a un parking de 2 plantas. En cada planta caben 5 vehículos.
- Los coches sólo pueden entrar en el parking si hay, al menos, 1 plaza libre en alguna de las plantas, sino deben esperar a que salga un coche.
- Cada vez que un coche entra en el parking mostrará un mensaje indicándolo. En el mensaje se indicará su identificador de thread y la planta en la que ha entrado.
- Si hay plazas en la planta 1 se elegirá ésta, sino se elegirá la planta 2.
- Una vez dentro esperará un tiempo aleatorio entre 1 y 10 segundos y abandonará el parking indicándolo con un mensaje similar al de la entrada.
- El programa principal debe esperar a que todos los vehículos salgan del parking antes de finalizar

Suponga que las librerías necesarias ya se han importado mediante los *include* necesarios

Ejercicio 4. (2,5 ptos)

a) Se tiene un disco de 20 GB con sistema de ficheros ext2 con las siguientes características:

- Tamaño del bloque de 8 KBytes.
- Direcciones de los bloques: 4 bytes.
- Estructura del i-nodo:
 - 10 punteros directos.
 - 1 puntero indirecto simple.
 - 1 puntero indirecto doble.
 - 1 puntero indirecto triple.



NOMBRE Y APELLIDOS: _____ NIA: _____

**Examen de Sistemas Operativos
11 de Septiembre de 2010**

Indica cuál es el número de bloques que ocupa un fichero de 100 MBytes, incluyendo tanto los bloques de datos como los de direcciones.

- b) Rellena la siguiente tabla de i-nodos y del contenido de los bloques de datos para que reflejen la situación de un disco en el que sólo hay un directorio DIR que contiene 1 fichero con 2 nombres f1 y f2 y , un enlace simbólico f3 al fichero f1

El orden en el que se han creado los diferentes elementos es :

- 1°. El directorio DIR
- 2°. f1
- 3°. f2
- 4°. f3

Tabla de I-nodos:

Nº Inodo	1	2	3	4	5
Tipo					
Contador					
Enlaces Fis.					
Dirección					
Bloque Datos					
.....					

Bloques de datos:

Nº Bloque	1	2	3	4	5
Contenido					



NOMBRE Y APELLIDOS: _____ NIA: _____

**Examen de Sistemas Operativos
11 de Septiembre de 2010**

SOLUCIÓN

Teoría 1.

- La MMU (memory management unit) traduce las direcciones virtuales en físicas
- La MMU produce un fallo de página (trap) cuando la dirección no está en memoria principal

Teoría 2.

El SO tiene la responsabilidad de gestionar los siguientes aspectos de la E/S y el almacenamiento secundario:

- Traducir peticiones a formato de manejador.
- Copiar memoria de/a memoria a/de controlador.
- Controlar operaciones por DMA.
- Controlar dispositivos de E/S serie: teclado, ratón, etc.
- Asignación y liberación de espacio.
- Planificación de accesos a los dispositivos.

Teoría 3.

- Semántica de coutilización UNIX
 - Las escrituras son inmediatamente visibles para todos los procesos con el archivo abierto.
 - Los procesos pueden compartir archivos. Si existe relación de parentesco pueden compartir el puntero.
 - La coutilización afecta también a los metadatos.

Ejercicio 2:

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

```
void hijo (int n,int fd[]) {
int suma=0,num;
```

```
do {
    read (fd[0],&num, sizeof(int));
    suma+=num;
} while (num != 0);
```



NOMBRE Y APELLIDOS: _____ NIA: _____

**Examen de Sistemas Operativos
11 de Septiembre de 2010**

```
if (n==0 )
    printf ("suma de los pares=%d\n", suma);
else
    printf ("suma de los impares=%d\n", suma);
exit(0);
}
main (){
int fd0[2], fd1[2]; //Tuberías para el hijo 0 y para el hijo 1
int pidh0,pidh1;
int num,cont=0,pidhm;

pipe (fd0);
if ( (pidh0= fork() ) == 0){
    close (fd0[1]);
    hijo(0,fd0);
}else
    if (pidh0 == -1) exit (1);

close (fd0[0]);

pipe (fd1);
if ( (pidh1= fork() ) == 0){
    close (fd1[1]);
    hijo(1,fd1);
}else
    if (pidh1 == -1) exit (2);

close (fd1[0]);

do {
    printf ("Dar número :");
    scanf ("%d", &num);
    if (num %2==0)
        write (fd0[1], &num, sizeof(int) );
    else
        write (fd1[1], &num, sizeof(int) );
} while (num != 0);
write (fd1[1], &num, sizeof(int) ); //escribo el 0 también para los impares

do {
    pidhm=wait(NULL);
    if (pidhm== pidh1 || pidhm== pidh0)
        cont++;
} while (cont == 2);
printf ("Fin de la aplicación\n");
```



NOMBRE Y APELLIDOS: _____ NIA: _____

**Examen de Sistemas Operativos
11 de Septiembre de 2010**

}

Ejercicio 3:

```
/* José Manuel Pérez Lobato */
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#define MAX    20    /* Numero máximo*/
#define TRUE    1
#define FALSE    0

pthread_mutex_t mutex;    /* mutex para controlar el acceso compartido */
pthread_cond_t esperaplaza;    /* controla la espera */
int plazaslibresP1=5;
int plazaslibresP2=5;

int espera=1;

void *coche(void *p) {
int plantaelegida=0;
srandom(pthread_self());
pthread_mutex_lock(&mutex);
while (plazaslibresP1 ==0 && plazaslibresP2 ==0)
pthread_cond_wait(&esperaplaza, &mutex);
if (plazaslibresP1 >0 ) {
plazaslibresP1--;
plantaelegida=1;
}
else {
plazaslibresP2--;
plantaelegida=2;
}
printf ("Entra %u en la planta %d\n", pthread_self(),plantaelegida);
pthread_mutex_unlock(&mutex);
sleep (5+ random()% 10);
pthread_mutex_lock(&mutex);
if(plantaelegida==1)
plazaslibresP1++;
else
plazaslibresP2++;
printf ("Sale %u de la planta %d\n", pthread_self(),plantaelegida);
pthread_cond_signal(&esperaplaza);
pthread_mutex_unlock(&mutex);
pthread_exit(0);
}
```



NOMBRE Y APELLIDOS: _____ NIA: _____

**Examen de Sistemas Operativos
11 de Septiembre de 2010**

```
main(int argc, char *argv[]){
    pthread_t th[MAX];
    pthread_attr_t attr;
    int i, afila;

    pthread_mutex_init(&mutex, NULL);
    pthread_cond_init(&esperaplaza, NULL);
    pthread_attr_init(&attr);
    for (i=0; i<MAX; i++){
        pthread_create(&th[i], &attr, coche, NULL);
    }
    for (i=0; i<MAX; i++)
        pthread_join(th[i], NULL);

    pthread_mutex_destroy(&mutex);
    pthread_cond_destroy(&esperaplaza);

    exit(0);
}
```

Ejercicio 4:**a)**

En cada bloque de direcciones caben 2048 direcciones de bloque: 8KBytes /4 bytes =2048 posiciones.

Los 10 apuntadores simples apuntarán a los 80 primeros Kbyte del fichero.

Cada bloque de direcciones que se necesite apuntará a 2048 apuntadores*8Kbytes= 16 Mbyte del fichero

Por tanto para llegar a 100 Mbyte necesitamos 7 bloques de direcciones.

El primero lo obtenemos del apuntador indirecto simple, que apunta directamente a uno de ellos. Para los 6 restantes necesitamos del apuntador indirecto doble que apuntará a 1 bloque de direcciones que apuntará a esos 6 bloques de direcciones necesarios.

Luego se necesitan 7 + 1= 8 bloques de direcciones

El tamaño en bloques de datos del fichero es de : 100 Mbyte / 8Kbytes =102400 / 8 = 12800 bloques

Luego el tamaño total son 12808 bloques más el inodo.

b)**Tabla de I-nodos:**

Nº Inodo	1	2	3	4	
Tipo	Directorio	Directorio	Fichero	Enlace Simbólico	
Contador Enlaces Fis.	3	2	2	1	



NOMBRE Y APELLIDOS: _____ NIA: _____

**Examen de Sistemas Operativos
11 de Septiembre de 2010**

Dirección Bloque Datos	11	12	13	14	
.....					

Bloques de datos:

Nº Bloque	11	12	13	14	
Contenido	. 1	. 2	Datos del Fichero f1	f1	
	.. 1	.. 1			
	DIR 2	f1 3			
		f2 3			
		f3 4			