Universidad Carlos III de Madrid	**Departamento de Informática** **Grado en Ingeniería Informática** **Sistemas Operativos** **Examen Ordinario – Tipos A y B** **23 de enero de 2010**	ARCOS

ATENCIÓN:

- Lea atentamente todo el enunciado antes de comenzar a contestar.
- Dispone de **120 minutos** para realizar la prueba.
- No se podrán utilizar libros ni apuntes, ni calculadoras de ningún tipo.
- Los teléfonos móviles deberán permanecer desconectados durante la prueba (apagados, no silenciados).
- Solamente se corregirán los ejercicios contestados con bolígrafo. Por favor no utilice lápiz.

APELLIDOS:

NOMBRE:

NIA:

Ejercicio 1 [0,5 puntos]: Defina el concepto de exclusión mutua.

Mecanismo que evita el acceso simultaneo a los recursos compartidos

Ejercicio 2 [0,5 puntos]: ¿Qué utilidad tiene el número mágico en el formato ELF?

Identifica al ejecutable. Así por ejemplo, en el formato ELF, el primer byte del archivo ejecutable debe contener el valor hexadecimal 7f y los tres siguientes los caracteres 'E', 'L' y 'F'.

Ejercicio 3 [1 punto]: Considere un sistema de archivos tipo UNIX en el que el tamaño de bloque es de 1 KB y las direcciones de bloques se representan como valores de 2 bytes. Determine la longitud del fichero más grande que se puede representar.

i-nodo	**Tamaño**
enlaces directos [10]	1KB x 10 = 10KB
enlace indirecto simple	(1KB/2bytes) x 1KB = 512 KB
enlace indirecto doble	(1KB/2bytes) x (1KB/2bytes) x 1KB = 256 MB
enlace indirecto triple	(1KB/2bytes) x (1KB/2bytes) x (1KB/2bytes) x 1KB = 128 GB
TOTAL	**10 KB + 512 KB + 256 MB + 128 GB**

Ejercicio 4 [1 puntos]: Explique cuál es el mayor problema que plantea la caché de bloques del sistema de ficheros.

Mantener la coherencia de los datos entre la caché y los datos disponibles en el sistema de almacenamiento.

Ejercicio 5 [2,5 puntos]: Una aplicación de gestión de ascensores tiene el siguiente código:

```
void * ascensor(void * numeroAscensor) {
 /* Código para un hilo que gestiona un ascensor */
 int * p;
 int n;
 p = (int*)numeroAscensor;
 n = *p;
 /* n contiene el número de ascensor */
 /* Resto de código de la función ascensor() */
}

void inicia_hilos(pthread_h * hilos, /* array de hilos */
                  int n) /* número de hilos */
{
 int i;
 for (i=0;i<n;i++) {
  pthread_create(&th[i], NULL, ascensor, (void*)&i);
 }
 /* Resto de función */
}
```

a) Si la función inicia_hilos se llama con un valor de 1 para el parámetro n ¿Puede darse algún problema? ¿Cuál?
   No
b) Si la función inicia_hilos se llama con un valor de 2 para el parámetro n ¿Puede darse algún problema? ¿Cuál?
   Si, debido a que el parámetro de los hilos se pasa por referencia, lo que puede provocar que ambos hilos compartan el valor.
c) Soluciones el problema de concurrencia que haya identificado en los apartados anteriores.
   Hay dos soluciones:
   1 Si se pasa por parámetro un valor entero:
   pthread_create(&th[i], NULL, ascensor, (void*)i);
   y en el proceso ligero cambiar:
   int p = (int)numeroAscensor;
   n = *p;

2 Solución válida para cualquier tipo de parámetro:

```
int ocupado = 1;
pthread_cond_t cond;
pthread_mutex_t mutex;

void * ascensor(void * numeroAscensor) {
 /* Código para un hilo que gestiona un ascensor */
 int * p;
 int n;

    pthread_mutex_lock(&mutex);
    p = (int*)numeroAscensor;
    n = *p;
    pthread_cond_signal(&cond);
    ocupado=0;
    pthread_mutex_unlock(&mutex);

 /* n contiene el número de ascensor */
 /* Resto de código de la función ascensor() */
}

void inicia_hilos(pthread_h * hilos, /* array de hilos */
                  int n) /* número de hilos */
{
 int i;
 for (i=0;i<n;i++) {
            pthread_mutex_lock(&mutex);
            pthread_create(&th[i], NULL, ascensor, (void*)&i);
            while(ocupado){
                  pthread_cond_wait(&cond,&mutex);
            }
            ocupado=1;
            pthread_mutex_unlock(&mutex);
 }
 /* Resto de función */
}
```

Ejercicio 6 [3 puntos]: Sea un sistema de ficheros que inicialmente está vacío. Es decir, inicialmente solamente contiene el directorio raíz.

Se pide:

a) Diagrama de i-nodos y bloques para el sistema de ficheros en el momento inicial.

i-nodos	0	1	2	3	4	5	6
	DIR CE=2 BD=100						

datos	100		101	102	103	104	105	106
	. ..	0 0						

b) Diagrama de i-nodos y bloques tras ejecutar el mandato mkdir d1; mkdir d2; cd d2; mkdir d3; cd ..; mkdir d4.

i-nodos	0	1	2	3	4		
	DIR CE=5 BD=100	DIR CE=2 BD=101	DIR CE=3 BD=102	DIR CE=2 BD=103	DIR CE=2 BD=104		

datos	100		101		102		103		104		105	106
	. .. d1 d2 d4	0 0 1 2 4	. ..	1 0	. .. d3	2 0 3	. ..	3 2	. ..	4 0		

c) Diagrama de i-nodos y bloques tras ejecutar el mandato cd /; ls > f1; cp f1 d2/f1

i-nodos	0	1	2	3	4	5	6
	DIR CE=5 BD=100	DIR CE=2 BD=101	DIR CE=3 BD=102	DIR CE=2 BD=103	DIR CE=2 BD=104	FILE CE=1 BD=105	FILE CE=1 BD=106

datos	100		101		102		103		104		105	106
	. .. d1 d2 d4 f1	0 0 1 2 4 5	. ..	1 0	. .. d3 f1	2 0 3 6	. ..	3 2	. ..	4 0	. .. d1 d2 d4	. .. d1 d2 d4

d) Diagrama de i-nodos y bloques tras ejecutar el mandato cd /; ln -s /d2/f1 /d1/ss

i-nodos	0	1	2	3	4	5	6	7
	DIR CE=5 BD=100	DIR CE=2 BD=101	DIR CE=3 BD=102	DIR CE=2 BD=103	DIR CE=2 BD=104	FILE CE=1 BD=105	FILE CE=1 BD=106	SYMLINK CE=1 BD=107

datos	100		101		102		103		104		105	106	107
	. .. d1 d2 d4 f1	0 0 1 2 4 5	. .. ss	1 0 7	. .. d3 f1	2 0 3 6	. ..	3 2	. ..	4 0	. .. d1 d2 d4	. .. d1 d2 d4	/d2/f1

e) Diagrama de i-nodos y bloques tras ejecutar el mandato cd/; ln /d2/f1 /d1/tt

i-nodos	0	1	2	3	4	5	6	7
	DIR CE=5 BD=100	DIR CE=2 BD=101	DIR CE=3 BD=102	DIR CE=2 BD=103	DIR CE=2 BD=104	FILE CE=1 BD=105	FILE CE=2 BD=106	SYMLINK CE=1 BD=107

datos	100		101		102		103		104		105	106	107
	. .. d1 d2 d4 f1	0 0 1 2 4 5	. .. ss tt	1 0 7 6	. .. d3 f1	2 0 3 6	. ..	3 2	. ..	4 0	. .. d1 d2 d4	. .. d1 d2 d4	/d2/f1

Ejercicio 7 [1,5 puntos]: Escriba un programa en C que cifre un fichero basado en un fichero de claves. El programa tomará tres parámetros en la línea de mandatos:

cifra datos cifrado clave

El parámetro datos, se corresponde con el nombre del fichero de datos que se desea cifrar. El parámetro cifrado se corresponde con el nombre del fichero cifrado que se desea generar. El parámetro clave se corresponde con el nombre del fichero que contiene la tabla de claves a usar.

El programa le byte a byte el fichero de datos y para cada byte determina su valor cifrado usando el fichero de claves. Para realizar el cifrado se realizar un XOR (or exclusivo) entre el i-ésimo byte del fichero de datos y el i-ésimo byte del fichero de claves. El fichero de claves se recorre de forma circular, de modo que cuando se llega al final se recorre otra vez desde el principio.

Escriba el programa usando exclusivamente archivos proyectados en memoria.