

	EXAMEN FINAL SISTEMAS OPERATIVOS	4 de septiembre de 2012
	Nombre y Apellidos: _____ NIF: _____.	

Ejercicio 1 (3 pts): Tras finalizar el diseño de nuestro sistema operativo, nos damos cuenta de que no proporcionamos soporte para semáforos generales. Por fortuna, sí disponemos de **mutex** y **variables condicionales**. Complete cada uno de los siguientes apartados para crear una biblioteca de rutinas que implemente semáforos generales a partir de mutex y variables condicionales.

- a) (1pt) Describa en primer lugar la estructura de datos que representará al semáforo general, completando/modificando la propuesta a continuación:

```
typedef struct sem_t {
    mutex_t *cerrojo;           // mutex
    cond_t *cond;              // variable condicional
    .....
} semaforo;
```

- b) (1pt) Implemente la función de creación e inicialización del semáforo. ¿Qué parámetro(s) deberá recibir dicha función?

```
semaforo *crear_semaforo (.....) {
    .....
    semaforo *sem = malloc(sizeof(semaforo));
    pthread_mutex_init(sem->cerrojo, NULL);
    pthread_cond_init(sem->cond, NULL);
    .....
    return sem;
}
```

- c) (1pt) Finalmente, implemente las funciones `wait()` y `post()`. Ambas recibirán como parámetro un semáforo (tipo **semaforo** creado en el apartado a) y devuelto por la función **crear_semaforo**).

Ejercicio 2 (1.5 pts): Considerar la siguiente secuencia de referencias a direcciones de memoria virtual generadas por un sólo programa en un sistema con paginación pura:

0x10, 0x1A, 0x1F4, 0x17C, 0x7C, 0x3B9, 0x185, 0x2FF, 0x24C, 0x434, 0x458, 0x36D

- a) (0.5pts) Deducir la cadena de referencias (secuencia de números de página generadas por el programa), suponiendo un tamaño de página de 256 Bytes.
- b) (1pt) Determinar razonadamente el número de fallos de página para cada una de las siguientes estrategias de sustitución, suponiendo que hay dos marcos de página disponibles para el programa: **Óptima, FIFO, LRU, Reloj**.

Ejercicio 3 (1.5 pt): Dado el siguiente código (suponga que la asignación de PIDs por parte del SO es consecutiva y comienza en 0):

```
main() {
    for (int i=0; i<4; i++){
        if (i%2 == 0) {
            fork();
            wait(NULL);
        }
    }
    printf("PID: %d, (%d)\n", getpid(), getppid());
}
```

- a) (0.5pts) Escriba la salida por pantalla del programa.
- b) (0.5pts) ¿Qué esquema jerárquico de procesos genera este programa?
- c) (0.5pts) ¿Cuántos procesos de esta jerarquía estarán presentes simultáneamente en memoria como máximo?

Ejercicio 4 (1.5 pt): Supóngase un disco de 256 cilindros, 4 cabezas, 100 sectores por pista y 2KB por sector. El disco gira a 6000rpm y el tiempo de desplazamiento o posicionamiento del brazo de cabezas de lectura/escritura es de 0.5ms por cilindro. El disco gira en dirección ascendente desde el sector 0 al 99 e inicialmente se encuentra en CPS (Cilindro, Pista y Sector)= (25, 3, 12).

- a) (0.5pts) Determinar el tiempo de posicionamiento del brazo de disco hasta alcanzar CPS=(15,2,15).
- b) (0.5pts) Calcular el tiempo necesario para la lectura de 900KB físicamente consecutivos a partir de la posición CPS del apartado anterior
- c) (0.5pts) y la posición CPS de la cabeza tras la lectura.

Ejercicio 5 (2.5 pts): Un sistema de ficheros basado en i-nodos y mapa de bits contiene la siguiente información:

Mapa de bits: 1 0 0 1 0 1 1 1 0 0 0 0 1 0 0 1 0 0 0

i-nodo 2		i-nodo 3		i-nodo 4		i-nodo 5		i-nodo 9	
Tamaño	1	Tamaño	2	Tamaño	1	Tamaño		Tamaño	1
#Enlaces	NA	#Enlaces	1	#Enlaces		#Enlaces	NA	#Enlaces	NA
Tipo F/D	D	Tipo F/D	F	Tipo F/D	F	Tipo F/D	D	Tipo F/D	D
Directo	3	Directo	6	Directo	12	Directo	0	Directo	
Indirecto	Null	Indirecto	7	Indirecto	Null	Indirecto	Null	Indirecto	Null

Bloque 0		Bloque 3		Bloque 5		Bloque 6		Bloque 7		Bloque 12		Bloque 15	
.	5	.	2	.	9	Datos sin formato						Datos sin formato	
..	2	5								
C	9	A	3										
D	4	B	5										
		E	4										

- a) (1.5pts) Rellene los huecos para que el sistema sea consistente. Asuma para ello que el tamaño se expresa en bloques.
- b) (1pt) Dibuje el árbol del directorio empleando óvalos para los directorios, rectángulos para los ficheros y triángulos para los datos.