

	EXAMEN FINAL SISTEMAS OPERATIVOS	3 de septiembre de 2013
	Nombre y Apellidos: _____ NIF: _____.	

Ejercicio 1 (3 pts): Una empresa de reparto recibe distintos tipos de paquetes que debe distribuir a través de repartidores especializados, cada repartidor es capaz de repartir sólo un único tipo de paquete.

1. (1pt) Implemente un hilo **Recepcionista()** encargado de recibir los paquetes y avisar a los hilos **Repartidor(int tipo)** según el siguiente esquema y utilizando como mecanismo de sincronización **ÚNICAMENTE Mutex y Variables de Condición**:

```
//Maximo num. de paquetes a la espera de ser repartidos
#define MAX_PAQUETES    100
#define TIPO1           1
#define TIPO2           2
int nTipo1=0, nTipo2=0;
//Resto de variables necesarias y su valor inicial:
```

```
Recepcionista () {
    int tipo;
    while(1){
        //Si el almacén está lleno, espera a que haya hueco
        tipo=atenderCliente(); //¡Tiempo de espera largo!
        switch(tipo){
            case TIPO1:
                nTipo1++;
                //Avisar a repartidores tipo 1
                break;
            case TIPO2:
                nTipo2++;
                //Avisar a repartidores tipo 2
                break;
        }
    }
}
```

2. (1pt) Implemente un hilo genérico **Repartidor(int tipo)** encargado de coger paquetes del almacén, sólo del tipo indicado durante el proceso de creación, y repartirlos (el proceso se continuará de manera indefinida). Este hilo deberá de ser capaz de funcionar adecuadamente con el **Recepcionista**.
3. (1pt) ¿Cuántos hilos **Repartidor** de cada tipo puede haber activos en el sistema? ¿Y recepcionista? Si alguna de las respuestas es 1 indique las modificaciones que habría que hacer para poder tener activos un número indeterminado de ellos. Razone sus respuestas.

Ejercicio 2 (2 pts): Supongamos un disco con 40 cilindros, 4 cabezas, y 20 sectores de 512 bytes cada uno. Supongamos también que el posicionamiento tarda 6ms por cilindro, que gira a 3000rpm y que la latencia rotacional promedio es de media vuelta. Si no se intenta colocar los bloques de un fichero cercanos unos a otros, dos bloques lógicamente consecutivos (es decir, que se siguen el uno al otro dentro del fichero) se encontrarán separados 13 cilindros por término medio. Sin embargo, si se intenta agrupar los bloques relacionados, la distancia media entre bloques puede reducirse a 2 cilindros.

1. ¿Cuánto tiempo llevaría leer un fichero de 51200 bytes en ambos casos?
2. ¿Cuánto tiempo que costaría leer un fichero de 51200 bloque en el caso mejor? ¿y en el caso peor?

Ejercicio 3 (2 pts): Un sistema de paginación pura tiene un tamaño de página de 1024B, una memoria virtual de 128 páginas numeradas de 0 a 127, y una memoria física de 10 marcos de páginas numerados de 0 a 9. El contenido actual de la memoria física es el siguiente:

Dir. Inicio	Contenido
0	Página 44

2048	Página 0

5120	Página 1
6144	Página 6

9216	TP

Conteste de forma razonada a las siguientes preguntas:

1. ¿Cuál es el contenido actual de la Tabla de Páginas?
2. Después de cargar la página 94 en la posición 3 y de sustituir la página 6 por la página 21 ¿Cuál es el nuevo contenido de la TP?
3. ¿Qué direcciones físicas referencian las direcciones virtuales 0x0 y 0x410?
4. ¿Qué ocurre cuando se referencia la dirección virtual 0x812?

Ejercicio 4 (3 pts): Un programador poco avezado pretende sacar partido de un sistema multiproceso paralelizando la copia de un fichero de la siguiente manera:

```
#define BLOCK 1024
char buf[BLOCK] = "xxxxxxxx...xxxxx";
void main() {
    pid_t pid;
    int fdo, fdd, size;
    fdo = open("Origen", O_RDONLY);
    fdd = open("Destino", O_RDWR | O_CREAT | O_TRUNC, 0666);
    for (int i=0; i < 4; i++) {
        lseek(fdo, i*BLOCK, SEEK_SET); //Posicionamiento absoluto
        lseek(fdd, i*BLOCK, SEEK_SET); //Posicionamiento absoluto
        pid = fork();
        if (pid==0){
            size=read(fdo, buf, BLOCK);
            write(fdd, buf, size);
            exit(0);
        }
    }
    while (wait(NULL) != -1) { };
    read(fdd, buf, BLOCK);
    lseek(fdd, 3*BLOCK, SEEK_SET);
    read(fdd, buf, BLOCK);
}
```

Contenido de Origen

Región	0 -> 1K-1	1K -> 2K-1	2K -> 3K-1	3K -> 4K-1
Datos	000...000	111...111	222...222	333...333

Responda razonadamente a las siguientes preguntas, suponiendo que la prioridad es para el proceso padre y que los hijos se ejecutan de manera secuencial:

1. ¿Cuál es el contenido final del fichero 'Destino'? Indica la evolución del contenido del array 'buf' para el proceso padre.
2. Sea sistema de ficheros de tipo Linux (nodos-i), con 4 punteros directos y un indirecto simple, tamaño de bloque de 512B y 4bytes por puntero. Describa el contenido del nodo-i y de los bloques de datos del fichero 'Origen'. Asuma que, en caso de necesitar bloques de datos, están asignados a partir del 100 y en orden ascendente
3. Escribe una versión correcta de la copia paralela.