

	EXAMEN FINAL SISTEMAS OPERATIVOS	2 de septiembre de 2014
	Apellidos, Nombre: _____ NIF: _____.	

1. (3pts): Un almacén de reparto dispone de un *Recepcionista* y múltiples *Repartidores*. Cuando llega un nuevo paquete al centro de reparto, el *Recepcionista* lo guarda en el almacén si hay hueco, esperando si fuese necesario. Los *Repartidores* están constantemente cogiendo paquetes del almacén y repartiéndolos. Tenga en cuenta lo siguiente:

- a. El *Recepcionista* recoge nuevos paquetes que pesarán entre 1 y 5Kgrs, valor de retorno de *recibirPaquete()*.
- b. Los *Repartidores* son capaces de cargar en su vehículo un máximo de 10Kgrs. Si el siguiente paquete a repartir no cabe en el vehículo, o el almacén está vacío y hemos cargado algo, se procederá al reparto, invocando a la función *repartir()*.
- c. La capacidad del almacén está definida en Kgrs, *KG_ALMACEN*. Asuma que se dispone de una cola FIFO de tamaño *KG_ALMACEN* para almacenar los paquetes y que para facilitar el manejo se dispone de las siguientes funciones:
 - i. *int insertarCola(int kgrPaq)*: inserta un elemento en la cola (retorna -1 si no hay hueco).
 - ii. *int primerElemento()*: devuelve el valor del primer elemento sin extraerlo (-1 si está vacía).
 - iii. *void extraerPrimerElemento()*: extrae el primer elemento.

Implemente el sistema basándose en *threads* y empleando **Mutex** y **Variables de Condición** como únicos mecanismos de sincronización. Siga el siguiente esquema:

```
#define KG_ALMACEN 1000 //Capacidad del almacén
int kgrsAlmacen = 0; //Peso total de los paquetes en el almacén

//Resto de variables necesarias y su valor inicial
...
```

<pre>thread_Recepcionista() { while(1) { int nuevoPaq; nuevoPaq=recibirPaquete() //Espero a que haya hueco insertarCola(nuevoPaq); //Avisar los repartidores } }</pre>	<pre>thread_Repartidor() { int cargaKgr; while(1) { cargaKgr=0; //Cargar vehículo (extrayendo de la cola) repartir() } }</pre>
--	---

NOTAS: Las llamadas a las funciones *recibirPaquete()* y *repartir()* no deben ser parte de la sección crítica, ya que pueden tardar en ejecutarse un tiempo considerable. Las llamadas a las funciones de la librería FIFO no son *thread-safe* (no disponen de mecanismos de protección internos).

2. (1.5pts): Considerar un fichero que consta de 6 bloques. ¿Cuántas lecturas y escrituras de bloques de datos y consultas/modificaciones de estructuras auxiliares del sistema de ficheros son necesarias para cada una de las tres estrategias de asignación (contigua, enlazada e indexada) al realizar las siguientes operaciones?:

- a) Añadir un bloque de información al comienzo, a la mitad y al final.
- b) Suprimir un bloque del principio, de la mitad y del final.

3. (2pts): Dato el siguiente código:

```

01: #define M 10
02: int myPipe[2];
03: void * newThread(void *num);
04: int main() {
05:     int pid,n;
06:     pthread_t tid;
07:     pipe(myPipe);
08:     if( (pid=fork()) == 0){
09:         close(myPipe[0]);
10:         for(n=0; n<M; n++){
11:             if( (n%2) == 0 ){
12:                 pthread_create(&tid,NULL,
13:                               newThread,&n);
14:                 pthread_join(tid,NULL);
15:             }
16:         }
17:     } else{
18:         wait(NULL);
19:         close(myPipe[1]);
20:         while( read(myPipe[0],&n,
21:                   sizeof(int)) !=0 ){
22:             printf("Leo %d\n",n);
23:         }
24:         close(myPipe[0]);
25:         return(0);
26:     }
27: void * newThread(void *num) {
28:     printf("Escribo %d\n", *(int*)num);
29:     write(myPipe[1],num,sizeof(int));
30:     return(NULL);
31: }
    
```

1. Indique **razonadamente** el resultado por pantalla de la ejecución del código, asumiendo que las llamadas al sistema no fallarán. Indique si puede haber variaciones y ponga un ejemplo en caso afirmativo.
2. ¿Qué ocurriría si intercambiáramos las líneas 07 con la 08?

4. (2pts): Un sistema de paginación pura tiene un tamaño de página de 32 bytes, una memoria virtual de 8 páginas y 6 marcos asignados (del 0 al 5). El contenido actual de la tabla de páginas es el siguiente:

#Pag	Marcos	Validez	Protección
0	4	1	RW-
1	1	1	-X-
2	0	1	-X-
3	3	0	RW-
4	2	1	RW-
5	3	1	---
6	0	0	RW-
7	3	0	---

Calcule las direcciones físicas correspondientes de las siguientes direcciones de memoria:

- a. Lectura de 0x14 y 0xF0
- b. Ejecución de 0x68 y 0x20
- c. Escritura de 0x92 y 0xC6

En caso de no ser posible la traducción, o se produzca algún error, indique por qué. En caso de fallo de página asigne el marco libre 5.

5. (1.5pts): Suponga que un manejador de disco tiene la siguiente cola de peticiones de pistas pendiente:

3 - 34 - 45 - 22 - 99 - 101 - 2 - 30

Supóngase además que la última tanda de peticiones atendidas por el controlador fue 42 y 44 y que el disco tiene 120 pistas. Indique el orden de servicio y el desplazamiento de la cabeza lectora siguiendo las políticas: **FIFO, SSF, SCAN y C-SCAN.**