



UNIVERSIDAD
DE MÁLAGA

Dpto. Lenguajes y
Ciencias de la Computación

Programación de Sistemas y Concurrencia

Ordinaria Junio 2014

APELLIDOS _____ NOMBRE _____

DNI _____ ORDENADOR _____ GRUPO _____

Ejercicio 1. Lenguaje C (2 puntos)

Se desea implementar una cola de procesos ordenada por prioridad. Esta cola inserta los procesos ordenados por el valor de su prioridad pero, en el caso de misma prioridad, se almacenan por orden de llegada. Se dispone de un fichero **binario** con el siguiente formato:

```
<numprocesos><idproceso><prioridad><idproceso><prioridad><idproc  
eso><prioridad><idproceso><prioridad>...
```

Donde:

<numprocesos> es el número de procesos de la cola (**int**)

<idproceso> es el identificador del proceso (**int**)

<prioridad> es la prioridad del proceso (**int**)

Implementar las siguientes procedimientos/funciones:

`void Crear_Cola(char *nombre, TColaPrio *cp).` Crear una cola de prioridad a partir del fichero con el nombre *nombre*. *cp* es un puntero al primer proceso de la cola. El **fichero de entrada** contiene 8 procesos almacenados en el siguiente orden:

0:4, 1:1, 2:4, 3:1, 4:2, 5:2, 6:2, 7:3

`void Mostrar (TColaPrio cp).` Muestra el contenido de la cola desde el primer elemento al último de la cola.

`void Destruir (TColaPrio *cp).` Elimina todos los elementos de la lista liberando su memoria.

`void Ejecutar_Max_Prio(TColaPrio *cp).` Ejecuta el proceso de mayor prioridad, eliminándolo de la lista y liberando su memoria. Si hubiera varios procesos de máxima prioridad se deben eliminar todos.

`void Ejecutar(TColaPrio *cp, int prio).` Ejecuta los procesos de prioridad *prio*. Elimina de la lista y libera la memoria de todos los nodos con la prioridad *prio*.

Ejercicios 2. Semáforos (2.5 puntos)

Implementar una solución con **semáforos binarios** al siguiente problema:

En una isla volcánica viven 60 personas que es necesario evacuar debido a la inminente erupción prevista por los Servicios de Previsión Telúrica. Se sabe a ciencia cierta que el volcán hará erupción dentro de 120 minutos y, para evacuar a los habitantes, hay un helipuerto en el centro de la isla y un helicóptero con capacidad para 6 personas máximo. Cada recorrido de evacuación que hace el helicóptero le lleva 12 minutos (6 de ida a la costa más cercana y otros 6 de vuelta). Cuando el helicóptero llega a la isla recoge a las personas que hayan llegado hasta ese momento al helipuerto (hasta un máximo de 6) y las evacua (no espera a nadie más). El primer viaje que hace el helicóptero puede ser en el minuto 0 o en el minuto 12. Por otro lado, cada habitante debe llegar al helipuerto por sus propios medios y, dado que se encuentran desperdigados por la isla, tarda un tiempo aleatorio entre 0 y 120 minutos en llegar al helipuerto.

Se pide simular la evacuación asumiendo que cada habitante es un hilo y que el helicóptero también es otro hilo. Aparentemente, todos los habitantes deberían poder ser evacuados pero, en la mayoría de las ejecuciones de la solución a este problema, hay habitantes que se quedan en la isla y perecen porque les pilla la erupción; se pide justificar también este comportamiento.

Ejercicios 3 y 4. Monitores (2.5 puntos). Paso de mensajes (2 puntos)

Supón que existen **I** (por ejemplo, $I=2$) **impresoras** que pueden ser utilizadas por **N usuarios** ($N>I$), y un **GestorImpresoras** que gestiona las peticiones de los usuarios. Las **impresoras** son hebras que están a la espera de que les envíen trabajos a imprimir. Una vez que el gestor les envía un trabajo, lo imprimen y, cuando la impresión termina, informan al gestor. Los **usuarios** son hebras que, de vez en cuando, necesitan imprimir un trabajo. Cuando un usuario quiere imprimir, lo solicita al gestor, *y espera hasta que el gestor le informa* de que la impresión del trabajo se ha realizado. El **gestor** acepta peticiones de los usuarios que quieren imprimir un trabajo. Cuando hay impresoras libres, le pasan el trabajo a la impresora. Asimismo, el **gestor** acepta mensajes por parte de las impresoras cuando han terminado de imprimir el trabajo que el gestor les ha asignado. En este caso, informan al usuario correspondiente de que la impresión de su trabajo ha terminado. Mientras que el trabajo de un usuario está siendo impreso, el gestor debe poder seguir aceptando peticiones de otros usuarios (si hay impresoras libres), o mensajes de otras impresoras que han acabado su impresión.

a) Implementar este sistema, suponiendo que las impresoras y los usuarios se sincronizan a través de un **monitor** GestorImpresoras, que puede utilizar métodos sincronizados o locks.

En el campus virtual, se puede encontrar el esqueleto de este ejercicio

b) Implementar este sistema, suponiendo que las impresoras y los usuarios se comunican a través de **canales síncronos**.