

PROGRAMACIÓN II

Práctica para realizar
Julio 2012

Fecha entrega: Viernes 29 de Junio

1. Un Sistema Operativo mantiene una cola de procesos que esperan para usar la CPU. Estos procesos están organizados por prioridad en una serie de colas tal y como indica la figura:

prioridad	34	45	114
TCola	4567 (primero cola) 6789 2290 (ultimo cola)	7777 (primero cola) 2323 9999 9897 (ultimo cola)		1122 (primero cola) 2555 1001 1003 (ultimo cola)

El número de prioridad indica qué cola es más prioritaria (en el ejemplo la más prioritaria sería la 114) y los números dentro de la cola son el PID (process identifier) que es un número identificando al proceso que está a la espera del uso de la CPU. Para el ejemplo anterior, el primer proceso en hacer uso de la CPU sería el 1122, luego el 2555 y así sucesivamente.

Tomando como base el siguiente tipo de datos para la cola por prioridad:

```
struct colaindividual
{
    int prioridad;
    TCola cola;
    /* TCola es una cola convencional implementada con cualquier método de los vistos en clase
    Podéis usar, por ejemplo, la implementación de nodos enlazados que figura en el cola.zip adjunto
    */
}

typedef struct colaindividual * TCOLAPRIO;
typedef long PID;
```

Las estructuras colaindividual están organizadas de manera contigua en memoria y supondremos que hay siempre 100 colas y, tal y como indica el ejemplo anterior, los números de prioridad están organizados ascendentemente (pero de manera no consecutiva).

Implementa una biblioteca que tenga las siguientes operaciones:

int EsColaVaciaPrio (TCOLAPRIO tcp) : devuelve 1 o 0 en función de si todas las colas están vacías o no.

*void primeroColaPrio (TCOLAPRIO tcp, PID *proceso)*: devuelve el proceso que es el primero en la cola por prioridad (en el ejemplo sería el 1122).

*void eliminarColaPrio (TCOLAPRIO *tcp)*: elimina el proceso que es el primero de la cola por prioridad (en el ejemplo eliminaría el 1122).

*void añadirColaPrio (TCOLAPRIO *tcp, PID proceso, int prioridad)*: Para esta operación supondremos que la prioridad que nos dan está siempre en la serie de prioridades de la cola por prioridad.

Requisito obligatorio: deberá implementarse el algoritmo de búsqueda binaria para encontrar la prioridad requerida dentro de las prioridades existentes. Tened en cuenta que hay siempre 100 colas en la estructura de datos y que todas las estructuras cola individual figuran consecutivamente en memoria. Nota: El no atender correctamente este requisito conlleva la calificación de suspenso