



# Software para sistemas empotrados y dispositivos móviles

## Control POSIX

### Grado Ing. Software

**Importante:** El código a entregar de cada ejercicio no puede tener errores de compilación.

1.- (5 puntos) Se desea simular un sistema de control de temperatura del agua de un depósito, que funcione adecuadamente en términos de confortabilidad tanto en verano como en invierno. Para ello, se deben desarrollar tres tareas que comparten el estado del sistema mediante variables compartidas protegidas con acceso en exclusión mutua.

- La tarea que simula la evolución de la temperatura actual del agua tiene un periodo de **1 segundo, prioridad 12**. Su comportamiento es el siguiente:
  - El valor inicial de la *temperatura actual* es 30 grados. El nuevo valor de la temperatura actual se calcula a partir de su valor anterior según las siguientes condiciones:
    - Si el valor de la *temperatura actual* es inferior al valor de la *temperatura natural*, entonces el valor de la *temperatura actual* se incrementa un grado.
    - Si el valor de la *temperatura actual* es superior al valor de la *temperatura natural*, entonces el valor de la *temperatura actual* se decrementa un grado.
    - El valor de la *temperatura natural* se genera de forma aleatoria en un rango entre [25,35].
    - Si el *chorro de agua fría o caliente* están activos (véase abajo), la temperatura se incrementará o se decrementará un grado respectivamente independientemente del valor natural.
- La tarea que simula el control de los actuadores de chorros de agua fría/caliente tiene un periodo de **3 segundos, prioridad 11**. Su comportamiento es el siguiente:
  - Si el valor de la *temperatura actual* es inferior a 25 grados, entonces el estado del *chorro de agua caliente* debe ser *Activo*, y en otro caso *Inactivo*.
  - Si el valor de la *temperatura actual* es superior a 35 grados, entonces el estado del *chorro de agua fría* debe ser *Activo*, y en otro caso *Inactivo*.
- La tarea que simula la monitorización del estado del sistema tiene un periodo de **4 segundos, prioridad 10**. Su comportamiento es el siguiente:
  - Muestra de forma periódica el estado actual del sistema: la *temperatura actual*, *temperatura natural*, *estado del chorro de agua fría*, *estado del chorro de agua caliente*

**Nota 1:** El comportamiento periódico de las hebras deberá diseñarse con ``clock_gettime()`` y ``clock_nanosleep()``.

**Nota 2.-** Se puede generar un número aleatorio con la función `rand()`. Es necesario inicializar el generador de números aleatorios al principio del programa mediante `srand(time(NULL))`.

2.- (5 puntos) Diseñar e implementar un programa para controlar el nivel de agua de un tanque. Existen una entrada de agua y una salida que son independientes y cada una está controlada por una válvula, la cual está preparada para dejar salir/ entrar un litro de agua por segundo, en el caso de la entrada y cada dos segundos en el caso de la salida.

El sistema será simulado por tres tareas (hebras), de las cuales las dos primeras están encargadas del control de la válvula en entrada y salida respectivamente, y la tercera tarea (hebra) será la encargada de gestionar y monitorizar el nivel del agua del tanque.

La tarea de control de entrada de agua simulará, mediante una tarea periódica con periodo de 1 sg, el llenado del agua en el tanque, y enviará una **señal**, con una probabilidad del 60% a la tarea de gestión del tanque.

La tarea de control de salida de agua simulará, mediante una tarea periódica con periodo de 2 sg, la salida de agua del tanque, y enviará una **señal**, con una probabilidad del 40%, a la tarea de gestión del tanque.

La tarea de gestión y monitorización del nivel de agua del tanque es una tarea que se encuentra, de forma permanente, a la espera de recibir notificaciones (señales) de las tareas de control de E/S de agua, y actualiza el nivel de agua según las señales recibidas. Además, también mostrará el valor del nivel del agua en el tanque.

Diseñar la solución con los mecanismos de temporizadores y hebras de POSIX. La gestión de las tareas periódicas de control de tornos se realizará mediante **temporizadores** (`create_timer`, etc).

Para gestionar la probabilidad en la generación de señales de entrada y salida, se puede generar un número aleatorio entre 0 y 100 (`rand()%101`) y comprobar si el número aleatorio supera un determinado umbral. Es necesario inicializar el generador de números aleatorios al principio del programa mediante `srand(time(NULL))`.