



Universidad de Alcalá

**DEPARTAMENTO DE AUTOMÁTICA
ARQUITECTURA Y TECNOLOGÍA DE COMPUTADORES**

**Grado en Ingeniería de Computadores
COMPUTACIÓN DE ALTAS PRESTACIONES**

Práctica 2-1

Implementar alarmas mediante hebras

OBJETIVO

Implementar el programa alarm que se entrega en versión secuencial mediante el paradigma de hebras. La idea es que nuestra versión con hebras tenga básicamente la misma funcionalidad.

DESARROLLO DE LA PRÁCTICA

Presentarse en la máquina

En primer lugar, es necesario presentarse en cualquiera de las máquinas del laboratorio y abrir un terminal alfanumérico, como ya se ha hecho en la práctica anterior.

Descarga del código

En la página web de la asignatura encontramos los recursos necesarios para la realización de esta práctica. Descargamos [alarm.zip](#) donde encontramos el código del programa `alarm.c` junto con los ficheros `errors.h`, `Makefile.al` y otros.

Fase de paralelización: parte I

Al observar al código del programa `alarm.c`, nos damos cuenta de que el usuario se queda suspendido mientras se ejecuta la espera que solicitó. El cambio que hemos de hacer, utilizando el paradigma de hebras, consiste en generar una hebra por cada pedido del usuario, de modo que sea la hebra la que ejecute la espera y, al finalizar ésta, emita por pantalla el mensaje asociado. Entretanto, el usuario puede seguir tecleando pedidos.

Fase de paralelización: parte II

En la parte II, se pide implementar una lista de pedidos que sea alimentada por la hebra principal (con la que el usuario interactúa para generar sus pedidos) y de donde otra hebra extrae los pedidos y los procesa.

El código para manejar listas está en los ficheros `listas.c` y `listas.h`, y un ejemplo está en el fichero `ejem_listas.c`.

Fase de paralelización: parte III

En la parte III, se pide que la hebra que procesa los pedidos no esté constantemente examinando la lista para ver si existe un pedido nuevo: ahora debemos usar señales y variables de condición, de modo que la hebra esté en estado suspendido mientras no existan elementos en la lista esperando para ser procesados.