

# Sistemas Operativos – Grado en Ingeniería del Software

## Práctica obligatoria 1

# Programación en C

### Descripción

En esta práctica se abordará el problema de implementar una biblioteca (también llamada librería) en el lenguaje de programación C. El grupo de prácticas deberá realizar las siguientes actividades:

- Escribir una biblioteca en lenguaje C que muestre N líneas por la salida estándar, con tres funcionalidades diferentes según la función invocada.
- Escribir un programa de prueba que sirva para comprobar el funcionamiento de las funciones de biblioteca.

Las funciones principales que se implementarán en la biblioteca serán las siguientes:

#### **Función *int head(int N)*:**

Se comporta de la misma forma que *head(1)* leyendo de la entrada estándar, es decir, muestra las N primeras líneas en la salida estándar recibidas por la entrada estándar.

#### **Función *int tail(int N)*:**

Se comporta de la misma forma que *tail(1)* leyendo de la entrada estándar, es decir, muestra las N últimas líneas en la salida estándar recibidas por la entrada estándar.

#### **Función *int longlines(int N)*:**

Muestra las N líneas más largas recibidas de mayor a menor, o todas ellas si hay menos de N líneas, por la entrada estándar de forma ordenada en la salida estándar.

El grupo de prácticas deberá realizar tres ficheros:

- *libreria.h*: fichero con las cabeceras de las funciones de biblioteca.
- *libreria.c*: fichero con el código de las funciones pedidas.
- *test.c*: programa de prueba de las funciones implementadas.

El código de los 3 ficheros estará debidamente comentado.

## **Objetivos parciales**

Los objetivos parciales perseguidos en esta práctica son los siguientes:

- Ser capaz de reconocer argumentos en el programa principal que llame a la biblioteca **(1.5 puntos)**.
- Realizar un control adecuado de errores **(1.5 puntos)**
- Ejecución correcta de la funcionalidad de la biblioteca. **(5 puntos)**
- Utilización de memoria dinámica **(2 puntos)**

**Nota:** las puntuaciones para cada objetivo parcial son las puntuaciones máximas que se pueden obtener si se cumplen esos objetivos.

**Nota:** no se debe hacer un programa separado para cada objetivo, sino un único programa genérico que cumpla con todos los objetivos simultáneamente.

## **Entrega de prácticas**

La entrega de prácticas se hará a través del Campus Virtual en las fechas anunciadas en el mismo. Se debe entregar tres archivos: uno de cabecera *libreria.h*, otro con el código de la librería *libreria.c* y un último fichero *test.c* con un programa de ejemplo que haga uso de la librería construida. Dichos ficheros deben estar debidamente comentados. Además de los mismos se debe entregar una memoria de la práctica en formato **pdf**. La memoria debe incluir:

- Índice de contenidos
- Autores
- Descripción del código: incluyendo descripción de las principales funciones implementadas.
- Comentarios personales: incluyendo problemas encontrados, críticas constructivas, propuesta de mejoras y evaluación del tiempo dedicado.
- No incluir código fuente
- NO DESCUIDE LA CALIDAD DE LA MEMORIA DE SU PRÁCTICA. Aprobar la memoria es tan imprescindible para aprobar la práctica, como el correcto funcionamiento del código C desarrollado. Si al evaluar la memoria se considera que no alcanza el mínimo admisible, la práctica se considerará SUSPENSA.

**IMPORTANTE:** Todos los archivos solicitados (fuentes y memoria) deberán empaquetarse en un único archivo comprimido en formato ZIP llamado **practica1.zip** y entregarse a través del Campus Virtual pinchando en el enlace correspondiente.

## **Evaluación de la práctica**

La práctica se evaluará comprobando el correcto funcionamiento de los distintos objetivos, y valorando la simplicidad del código, los comentarios, la óptima gestión de recursos, la gestión

de errores y la calidad de la memoria. El profesor podrá solicitar una defensa oral de la práctica si lo considerase necesario.

A la hora de codificar las funciones pedidas en ambos apartados, se deberán respetar una serie de normas de estilo:

- Las variables locales de las funciones se declararán inmediatamente después de la llave de comienzo del cuerpo de la misma. Se penalizará la declaración de variables entre sentencias ejecutables de las funciones.
- Al declarar una variable, no se admitirá que se le asigne un valor inicial que dependa del valor de otras variables. El valor asignado en estos casos siempre deberá ser conocido en tiempo de compilación.
- Cuando se declare una variable de tipo array, su tamaño deberá ser conocido en tiempo de compilación. Si se quiere utilizar un array de tamaño variable, deberá crearse en memoria dinámica mediante las funciones correspondientes (malloc, calloc o realloc). La memoria dinámica solicitada deberá liberarse (free) antes de salir del programa.
- Las operaciones sobre strings (copia, comparación, duplicación, concatenación, etc) se realizarán en lo posible mediante las funciones indicadas en string.h (ver referencias a la biblioteca de C en el Campus Virtual).
- En general, se penalizará el uso de construcciones propias de C++.
- Al compilar el código fuente, deberá producirse el menor número posible de avisos o warnings (mejor que no se produzca ninguno).

El incumplimiento de estas normas de estilo, así como de otras normas que puedan ser anunciadas por el profesor a través del Campus Virtual, conllevará una penalización en la nota obtenida a partir del baremo indicado en el apartado de objetivos parciales.

### **Autoría de la práctica**

La práctica se debe realizar en grupos de 2 personas como máximo.

El hecho de detectar copia en las prácticas expondrá a los alumnos a la posibilidad de una apertura de expediente disciplinario y expulsión. En caso de detectar copia, los alumnos afectados serán suspendidos en la TOTALIDAD de la asignatura. Una práctica será considerada copia en caso de que contenga la totalidad o una parte de la práctica de otro alumno. Se considerará copia en caso de:

- Archivos que contengan la totalidad o fragmentos de código de otro alumno
- Memorias con la totalidad o fragmentos de frases e imágenes de otro alumno

El profesor podrá hacer uso de detectores automáticos de plagio en las prácticas, tanto en la parte referente al código como a la memoria.