

# Tarea 1: Programación básica en C

(5%)

## Objetivos:

Los objetivos de esta tarea son:

1. Iniciarse en la programación en C y el uso de las librerías estándares.
2. Familiarizarse con el manejo de memoria dinámica y estructuras de listas en C
3. Familiarizarse con el compilador y el uso de *make*

## Enunciado:

Desarrollar un programa en C (que llamaremos `filetree`) que represente un árbol de directorio a través de listas con las siguientes consideraciones:

1. Los nodos del árbol representarán un archivo o un directorio. La información que cada nodo debe contener es:
  - Nombre del archivo o directorio.
  - Tipo (directorio o archivo)
  - Número de archivos (sólo si es un directorio)
  - Apuntadores necesarios para armar la estructura del árbol.
2. El nodo raíz debe ser el directorio `/` y debe estar creado por defecto. Esto quiere decir que la lista que representa el árbol de directorio, inicialmente se creará con un solo nodo: el nodo de `/`.
3. El resto de los archivos y directorios se irán agregando con los comandos:  
`mk < nombre absoluto del nuevo directorio>`  
`md < nombre absoluto del nuevo archivo>`
4. Adicionalmente se podrán eliminar archivos o directorios con el comando:  
`rm < nombre absoluto del archivo o directorio>`

Sólo se permite borrar un directorio si no contiene archivos.

5. La secuencia de comandos para agregar/eliminar archivos/directorios será leída desde un archivo cuyo nombre es pasado como parámetro al momento de la ejecución. Habrá un comando por línea, sin líneas en blanco ni comentarios. Ejemplo del contenido de un archivo de comandos:

```
md /file1.dat
mk /Cursos
mk /Juegos
mk /Cursos/ci-3881
md /Cursos/ci-3881/clases.tex
md /file2.tex
rm /file1.dat
rm /Cursos/ci-3881
```

6. Luego de ejecutar todos los comandos se debe imprimir por pantalla el árbol resultante, en

orden de búsqueda en anchura (*BFS-Breadth First Search*).

7. Para iniciar la corrida del programa se usará el comando:  
`filetree < nombre del archivo de comandos>`

## Observaciones importantes:

1. No deben descuidar la utilización de la memoria, esto implica hacer un buen uso de las llamadas `malloc( )` y `free( )`.
2. Deben hacer uso del `makefile`.
3. Deben hacer un programa modular y claro. Recuerden usar librerías y headers (archivos `.h`).
4. La tarea puede ser desarrollada en Linux. Es importante que se aseguren que se ejecute correctamente en las máquinas del ldc.

## Entrega:

La entrega se realizará el día 06-03-2014 en formato electrónico a través del Aula Virtual y consistirá en un archivo comprimido `tar.gz` con los programas fuentes, el `makefile` (obligatorio) y un archivo de texto `Leeme.txt` con la documentación básica del programa: autores, fecha, qué hace el programa, cómo se compila, cómo se invoca, qué entradas requiere y qué salidas produce.

Prof. Yudith Cardinale