



---

## Tarea 1 Sistemas Operativos

### Llamadas a Sistema y Utilización de Ficheros

Profesor: Viktor Tapia  
Ayudantes: Fabian Saso - Juan Carlos Bustamante

23-08-2016

## 1. Ejercicio

Crear un programa en C utilizando llamados a sistema, debe ser capaz de desplazarse a través de los directorio en un sistema Linux, listar los contenidos del directorio actual y abrir un archivo mostrando su contenido. Para esto se deberán programar los siguientes comandos:

I. cd:

- a) cd ruta\_absoluta: cambia el directorio actual al especificado por la ruta\_absoluta.
- b) cd ruta\_relativa: cambia el directorio actual al especificado por la ruta\_relativa.
- c) cd ..: cambia el directorio actual al inmediatamente superior.

II. ls:

- a) ls n: lista el contenido del directorio actual ordenado por nombres, mostrando nombre, tipo y fecha de modificación.
- b) (opcional) ls t: lista el contenido del directorio actual ordenado por fecha de modificación, mostrando nombre tipo y fecha de modificación.

III. open:

- a) open ruta\_absoluta: abre el archivo especificado por la ruta\_absoluta y muestra su contenido por consola.
- b) open nombre\_archivo: abre el archivo especificado por el nombre\_archivo ubicado en el directorio actual y muestra su contenido por consola.

IV. close:

- a) Al ejecutarse, el programa debe mostrar la ruta actual seguida del símbolo \$ y cada vez que se ejecute un comando se vuelve a mostrar, exceptuando al comando open que solo mostrara el contenido del archivo y símbolo \$ al final para indicar una nueva línea de comando.

**Los alumnos de ILI**, además deberán analizar el comportamiento de un sistema Linux a través de la observación de las estructuras de datos del kernel, las cuales son expuestas a través del sistema del archivo virtual /proc. El objetivo de este proyecto es que escriban un programa que reporte el comportamiento del kernel de Linux. El programa debe imprimir los siguientes valores:

- Tipo de procesador.
- Versión del kernel de Linux.
- Tamaño de la memoria configurada en el computador.
- Tiempo transcurrido desde el último boot del sistema.



## 2. Ejemplo

```
JuanCarlos@notebook S0/tarea1\$ ./tarea1
/home/JuanCarlos/S0/tarea1\$ cd ..
/home/JuanCarlos/S0\$ cd tarea1
/home/JuanCarlos/S0/tarea1\$ ls n
makefile archivo 23-08-2016 15:29
README.txt
tarea1      archivo 23-08-2016 15:29
tarea1.c    archivo 23-08-2016 15:29.
tarea1.o    archivo 23-08-2016 15:29
/home/JuanCarlos/S0/tarea1\$ open tarea1.c
\#include <stdio.h>
\#include <stdlib.h>
.
.
.
\$ close
/home/JuanCarlos/S0/tarea1\$
```

## 3. Condiciones

- En C, las funciones `exec` y `system` no pueden ser utilizadas.
- El código tiene que estar bien comentado e indentado.

## 4. Entrega

- La tarea debe realizarse de manera individual.
- Para la los alumnos de **INF** la fecha de entrega es hasta el **Lunes 12 de Septiembre de 2016 a las 23:55 hrs**, mientras que los alumnos de **ILI** la fecha de entrega es hasta el **Miércoles 21 de Septiembre de 2016 a las 23:55 hrs**.
- Por cada día de atraso se descuentan 20 puntos (máximo 3 días de atraso).
- La tarea debe subirse a Moodle con el formato **Tarea1\_Apellido\_Rol.tar.gz**. Dentro del tar debe ir una carpeta con el mismo nombre que contiene el código fuente del programa en c con su makefile. Además del README.txt con las instrucciones de compilación de ser necesario.
- Cualquier intento de copia, **será sancionado con nota 0** para todos los involucrados.