

# UNIVERSIDAD NACIONAL DE CÓRDOBA

## Facultad de Ciencias Exactas Físicas y Naturales



## Sistemas Operativos I

### Laboratorio 1: Observando el Comportamiento de Linux

#### Autor:

- Quinteros Castilla, Nicolás D. - 34351470

#### Docentes:

- Pablo Martínez
- Hugo Carrer

# Objetivos

Iniciar los laboratorios con un proyecto sencillo, donde se muestre como a través del sistema de archivos virtual `/proc` de Linux, podemos inspeccionar información interna del kernel. Se deberá generar la utilidad `ksamp` que muestra de diversas formas algún subconjunto de la información disponible en `/proc`.

## Desarrollo

Se implementó la aplicación en lenguaje C, mediante el entorno de desarrollo Eclipse, que recopila información del directorio `/proc` de Linux según diferentes opciones ingresadas por el usuario.

### Tarea A

Para comenzar, se recopiló información acerca de la estructura del directorio **`/proc`** de los siguientes ficheros:

Hostname: `/proc/sys/kernel/hostname`

Tipo y modelo de CPU: `/proc/cpuinfo`

Versión del kernel: `/proc/version`

Uptime: `/proc/uptime`

Cantidad de filesystems que soporta el kernel: `/proc/filesystems`

Tiempo de CPU: `/proc/stat`

Cambios de contexto, boot time y procesos: `/proc/stat`

Peticiones a disco: `/proc/diskstats`

Memoria total y disponible: `/proc/meminfo`

Promedio de carga en el último minuto: `/proc/loadavg`

Se utilizó **`getopt_long`**, para procesar opciones cortas y largas ingresadas por el usuario. Las entradas permitidas son:

- |   |   |
|---|---|
| <b><code>-h , --help</code></b>           | Despliega la ayuda.   |
| <b><code>-s , --parteC</code></b>         | Muestra la cabecera y la parte B de la consigna   |
| <b><code>-l a b , --parteD a b</code></b> | Muestra la cabecera, la parte B y la parte D de la consigna, donde <b><code>a</code></b> es el intervalo y <b><code>b</code></b> la duración. |

### Tarea B

Cuando el usuario inicie la aplicación con cualquier opción, se mostrará por defecto la siguiente información:

- Tipo y modelo de CPU.
- Versión del kernel.
- Cantidad de tiempo transcurrido desde que se inició el sistema operativo (uptime).
- Cantidad de sistemas de archivo soportado por el kernel.

Para leer las cadenas de caracteres se implementó la función **void getValor(char\* file, char\* value, char\* key)**, la cuál obtiene la cadena que acompaña a una palabra clave y la almacena en **value**.

Para dar formato a las fechas leídas en segundos se implementó la función **void formatear(float seg, char\* tiempo)**, que recibe las mismas y las devuelve con el formato **dd hh:mm:ss**.

## Tarea C

Cuando el usuario ingresa la opción corta **-s**, o la opción larga **--parteC**, se agrega a la salida la siguiente información:

- Cantidad de tiempo de CPU utilizado para usuarios, sistema y proceso idle.
- Cantidad de cambios de contexto.
- Fecha y hora cuando el sistema fue iniciado.
- Número de procesos creados desde el inicio del sistema.

## Tarea D

Cuando el usuario ingresa la opción corta **-l** seguido de dos parámetros **a** y **b** (**intervalo** y **duración**), se procede a imprimir por pantalla la información de la tarea B, la tarea C, además de lo siguiente:

- Número de peticiones a disco realizadas.
- Cantidad de memoria configurada en el hardware.
- Cantidad de memoria disponible.
- Lista de los promedios de carga de 1 minuto.
- Se muestran los 4 puntos anteriores durante “b” segundos tomando muestras en intervalos de 2 segundos.

Se adjunta junto al informe los siguientes archivos:

- ksamp.c
- makefile

El archivo makefile ejecutará la compilación de ksamp.c, seguido comandos válidos e inválidos para la aplicación, para poder verificar el comportamiento de la misma con diferentes parámetros de entrada.

#### **#Comandos validos**

##### **#Ayuda**

./ksamp.o -h

./ksamp.o --help

##### **#Parte C**

./ksamp.o -s

./ksamp.o --parteC

##### **#Parte D**

./ksamp.o -l 2 10

./ksamp.o --parteD 1 5

#### **#Comandos invalidos**

./ksamp.o -a

./ksamp.o -s 2

./ksamp.o -l 1

./ksamp.o -l 10 5

./ksamp.o -s -l