

TRABAJO PRACTICO N° 1: **KSAMP**

ALUMNOS:

Sosa Ludueña, Diego Matias.
Choquevilca Zotar, Gustavo Braian.

MATERIA: Sistema Operativo I

PARTE A

Lo que se realizó fue buscar información en el directorio **/proc** para observar los datos que contiene dicha estructura del directorio, para familiarizarse con información que brinda y de esta manera avanzar en las siguientes partes.

- Para obtener los datos de tipo y modelo de CPU, se debe entrar a **/proc/infocpu**.
vendor_id : GenuineIntel
model name : Pentium(R) Dual-Core CPU T4500 @ 2.30GHz
- Para obtener el dato de versión del kernel, se debe entrar a **/proc/version**.
Linux version 3.16.0-23-generic
- Para obtener el dato de Tiempo en días, horas, minutos y segundos que han transcurrido desde que se inició el sistema operativo se debe entrar a **/proc/uptime**.
9236.96
Hay que tener en cuenta que este numero esta en segundos y se debe pasar a formato ddD:hh:mm:ss, el cual se realiza con un algoritmo que lo haremos en las siguientes partes.
- Para obtener los datos de cuánto tiempo de CPU ha sido empleado para procesos de usuario, de sistema y cuando tiempo no se usó, se debe entrar a **/proc/stat**.
cpu 186709 735 31261
Como vemos todos estos números están en centésimas de segundos, donde el primero es el tiempo empleado para procesos de usuario, el segundo es el tiempo empleado para procesos de sistemas y el tercero para procesos idle.
- Para obtener los datos de cantidad de memoria libre y cantidad de memoria total, se debe entrar a **/proc/meminfo**.
MemTotal: 1932912 kB
MemFree: 364344 kB
- Para obtener la cantidad de sistemas de archivo soportados por el kernel, se debe entrar a **/proc/filesystem**. Si entramos a este fichero nos mostrará los tipos de archivos soportados por el kernel pero para saber la cantidad solo hay que realizar un algoritmo para que recorra todo el fichero y obtener dicha cantidad, esto se realizó en etapas posteriores.
- Para saber la cantidad de pedidos W/R a disco que se han realizado, se debe entrar a **/proc/diskstats**.
11815
- Para saber la cantidad de cambios de contexto que han sucedido, se debe entrar a **/proc/stat**.
ctxt 14484013

- Para saber los procesos creados desde que se inició el sistema, se debe entrar a **/proc/stat**.
processes 4637

PARTE B

En esta parte se realizó la primera version del ksamp, donde se mostraran capturas para apreciar los datos que se pedían.

```
diego@diego-Satellite-L515:~$ cd Documentos/S01/TP1/KSAMP
diego@diego-Satellite-L515:~/Documentos/S01/TP1/KSAMP$ make
gcc ksamp.c -o ksamp.out
diego@diego-Satellite-L515:~/Documentos/S01/TP1/KSAMP$ ./ksamp.out

//////////////////SISTEMA OPERATIVO 1//////////////////
TRABAJO PRACTICO N°1: KSAMP
ALUMNO: SOSA LUDUEÑA DIEGO MATIAS
//////////////////

//////////////////CABECERA//////////////////
NOMBRE DE LA MAQUINA: diego-Satellite-L515
FECHA Y HORA ACTUAL: 2016-09-23 02:26:44
//////////////////

//////////////////PARTE B//////////////////
TIPO DE CPU: GenuineIntel
MODELO DE CPU: Pentium(R) Dual-Core CPU T4500 @ 2.30GHz
VERSION DE KERNEL: 3.16.0-23-generic
TIEMPO ACTIVO DE SO: 0D:3:9:15
CANTIDAD DE SISTEMA DE ARCHIVOS SOPORTADOS POR EL KERNEL: 25
//////////////////
diego@diego-Satellite-L515:~/Documentos/S01/TP1/KSAMP$
```

Como se observa incluye esta parte la cabecera y los datos pedidos en el enunciado. Como vemos se usa **makefile** como se pide en el enunciado y cuando se coloca **./ksamp.out** realizamos el uso de la primera version del ksamp.

PARTE C

En esta parte se realizó la segunda version del ksamp, en la que se utiliza todo lo pedido para la parte b pero además se incorpora otra serie de datos del directorio **/proc**. Se mostrará una captura para apreciar los datos.

```
diego@diego-Satellite-L515:~$ cd Documentos/S01/TP1/KSAMP
diego@diego-Satellite-L515:~/Documentos/S01/TP1/KSAMP$ make
gcc ksamp.c -o ksamp.out
diego@diego-Satellite-L515:~/Documentos/S01/TP1/KSAMP$ ./ksamp.out -s

////////////////////SISTEMA OPERATIVO 1////////////////////
TRABAJO PRACTICO N°1: KSAMP
ALUMNO: SOSA LUDUEÑA DIEGO MATIAS
////////////////////

////////////////////CABECERA////////////////////
NOMBRE DE LA MAQUINA: diego-Satellite-L515
FECHA Y HORA ACTUAL: 2016-09-23 02:35:33
////////////////////

////////////////////PARTE B////////////////////
TIPO DE CPU: GenuineIntel
MODELO DE CPU: Pentium(R) Dual-Core CPU T4500 @ 2.30GHz
VERSION DE KERNEL: 3.16.0-23-generic
TIEMPO ACTIVO DE SO: 0D:3:18:4
CANTIDAD DE SISTEMA DE ARCHIVOS SOPORTADOS POR EL KERNEL: 25
////////////////////

////////////////////PARTE C////////////////////
CANTIDAD DE TIEMPO DE CPU EN SEGUNDOS UTILIZADO POR:
USUARIO: 2716
SISTEMA: 7
PROCESO IDLE: 457
CANTIDAD DE CAMBIOS DE CONTEXTO: 17289703
FECHA Y HORA CUANDO EL SISTEMA FUE INICIADO: 2016-09-22 23:17
NUMERO DE PROCESOS CREADOS DESDE EL INICIO DEL SISTEMA: 5017
////////////////////
diego@diego-Satellite-L515:~/Documentos/S01/TP1/KSAMP$ █
```

Como se observa incluye la parte b y además la parte c. Como vemos se usa **makefile** como se pide en el enunciado y cuando se coloca **./ksamp.out -s** realizamos el uso de la segunda version del ksamp.

PARTE D

En esta última versión del ksamp, además de incluir las partes anteriores se anexara la parte d. Esta parte fue la más laboriosa debido por los ítems que se pedían. Ahora se mostrará una captura de pantalla para observar la información.

```
diego@diego-Satellite-L515:~$ cd Documentos/S01/TP1/KSAMP
diego@diego-Satellite-L515:~/Documentos/S01/TP1/KSAMP$ make
gcc ksamp.c -o ksamp.out
diego@diego-Satellite-L515:~/Documentos/S01/TP1/KSAMP$ ./ksamp.out -l 1 3
```

```
/////////////////SISTEMA OPERATIVO 1/////////////////
TRABAJO PRACTICO N°1: KSAMP
ALUMNO: SOSA LUDUEÑA DIEGO MATIAS
/////////////////
```

```
/////////////////CABECERA/////////////////
NOMBRE DE LA MAQUINA: diego-Satellite-L515
FECHA Y HORA ACTUAL: 2016-09-23 02:43:09
/////////////////
```

```
/////////////////PARTE B/////////////////
TIPO DE CPU: GenuineIntel
MODELO DE CPU: Pentium(R) Dual-Core CPU T4500 @ 2.30GHz
VERSION DE KERNEL: 3.16.0-23-generic
TIEMPO ACTIVO DE SO: 0D:3:25:39
CANTIDAD DE SISTEMA DE ARCHIVOS SOPORTADOS POR EL KERNEL: 25
/////////////////
```

```
/////////////////PARTE C/////////////////
CANTIDAD DE TIEMPO DE CPU EN SEGUNDOS UTILIZADO POR:
USUARIO: 2916
SISTEMA: 8
PROCESO IDLE: 498
CANTIDAD DE CAMBIOS DE CONTEXTO: 18502675
FECHA Y HORA CUANDO EL SISTEMA FUE INICIADO: 2016-09-22 23:17
NUMERO DE PROCESOS CREADOS DESDE EL INICIO DEL SISTEMA: 5268
/////////////////
```

```
/////////////////PARTE D/////////////////
NUMERO DE PETICIONES A DISCO REALIZADAS: 12492
MEMORIA TOTAL: 1932912 kB
MEMORIA LIBRE: 234508 kB
PROMEDIO DE CARGA EN EL ULTIMO MINUTO: 0.64
[PAUSA DE 1 SEGUNDOS]
```

```
NUMERO DE PETICIONES A DISCO REALIZADAS: 12492
MEMORIA TOTAL: 1932912 kB
MEMORIA LIBRE: 234068 kB
PROMEDIO DE CARGA EN EL ULTIMO MINUTO: 0.59
[PAUSA DE 1 SEGUNDOS]
```

```
NUMERO DE PETICIONES A DISCO REALIZADAS: 12492
MEMORIA TOTAL: 1932912 kB
MEMORIA LIBRE: 233856 kB
PROMEDIO DE CARGA EN EL ULTIMO MINUTO: 0.59
[PAUSA DE 1 SEGUNDOS]
```

```
/////////////////
diego@diego-Satellite-L515:~/Documentos/S01/TP1/KSAMP$ █
```

Como se observa se muestran todas las partes. Como vemos se usa **makefile** como se pide en el enunciado y cuando se coloca **./ksamp.out -l a b** realizamos el uso de la última versión del ksamp.

Para juntar estas tres partes (**ParteB.h**, **ParteC.h**, **ParteD.h**), se realizó un archivo principal (**ksamp.c**). Luego se utilizó el **makefile** para ayudar a compilar nuestro programa, es de gran importancia este tipo de archivo ya que presenta muchas ventajas para programas grandes, debido a que es capaz de saber qué cosas hay que recompilar. Nos guarda los comandos de compilación con todos sus parámetros para encontrar librerías, ficheros de cabecera.

Al compilar nuestro proyecto primero se lo testeó con flags **wall** y **pedantic** para que nos muestre los mensajes de error si es que los hay y las advertencias del compilador. En nuestro caso se corrigieron los errores y se obtuvo el siguiente resultado:

```
gabriel@gabi-Linux:~$ cd Documentos/S01/TP1/KSAMP
gabriel@gabi-Linux:~/Documentos/S01/TP1/KSAMP$ gcc -Wall ksamp.c
gabriel@gabi-Linux:~/Documentos/S01/TP1/KSAMP$ gcc -pedantic ksamp.c
gabriel@gabi-Linux:~/Documentos/S01/TP1/KSAMP$
```

Cuando analizamos el código con **cppcheck** nos advirtió de unos errores los cuales fueron modificados. Por lo tanto hicimos una captura de pantalla con las el análisis, obteniendo el código sin errores y sin warnings la cual es:

```
gustavo@gustavo-VPCEA35FL:~$ cd Descargas/KSAMP
gustavo@gustavo-VPCEA35FL:~/Descargas/KSAMP$ cppcheck .
Checking ksamp.c...
gustavo@gustavo-VPCEA35FL:~/Descargas/KSAMP$
```

CONCLUSIÓN

Al observar cada archivo del directorio **/proc** muestra lo importante y útiles que son cada uno de estos ficheros para testear de manera dinámica todos los procesos que maneja el kernel, también resulta interesante poder capturar cualquiera de estos ficheros para poder tratarlo o simplemente ver su estado en tiempo real. En cuanto a lo que respecta de nuestro tratamiento que le hicimos a los datos capturados de cada fichero, agregamos funcionalidades que pueden ser útiles ya sea para otros usuarios o para el mismo sistema. Al utilizar tanto flags **wall** como **pedantic** pudimos observar los mensajes de error y las advertencias del compilador, editando nuestro código y de esta forma dejar de manera óptima la codificación. En cuanto a la herramienta **cppcheck** también nos proporcionó grandes sugerencias que fueron corregidas.