

Universitat de les Illes Balears

Escola Politècnica Superior

21719 - Avaluació del Comportament de Sistemes Informàtics.

Práctica 1: Tema 2 - Monitorización.



Universitat
de les Illes Balears

Khaoula Ikkene

Grupo 102

khaoula.ikkene1@estudiant.uib.cat

Tabla de contenido

- I. [Monitorización de la CPU](#)
 - II. [Monitorización de la memoria principal](#)
 - III. [Monitorización en paralelo](#)
 - IV. [Pregunta voluntaria](#)
-

I. Monitorización de la CPU

Para esta parte se nos pide monitorizar la CPU con uso del monitor TOP durante una hora y con intervalos de muestreo de dos segundos.

1. ¿Cuántas CPUs tiene el sistema que se ha monitorizado? ¿De dónde se ha obtenido esa información?

En nuestro terminal de Linux ejecutamos el comando `cat /proc/cpuinfo` para encontrar la cantidad de CPUs que contiene nuestro sistema, que en este caso son 10.

```
siblings      : 10
core id       : 9
cpu cores     : 10
apicid        : 9
initial apicid : 9
```

2. ¿Cuál es la utilización media de la CPU en modo usuario, sistema y en global?

Para evitar trabajar con varios programas, se ha decidido desarrollar un script en shell que integra tanto los comandos relacionados con la monitorización (comando TOP) como el tratamiento de los datos finales en Python.

En primer lugar, se ejecutó el comando top especificando el intervalo de muestreo (2 segundos) y la cantidad de intervalos necesarios, calculados de la siguiente manera:
1 hora = 60 minutos = 3600 segundos → $3600/2 = 1800$ intervalos de muestreo de 2 segundos.

La información proporcionada por el comando top se ha guardado en el archivo CPU.txt. Luego, mediante un programa en Python, se han especificado manualmente los campos de datos de interés y se ha generado el archivo de salida "monitorizacionCPU.csv".

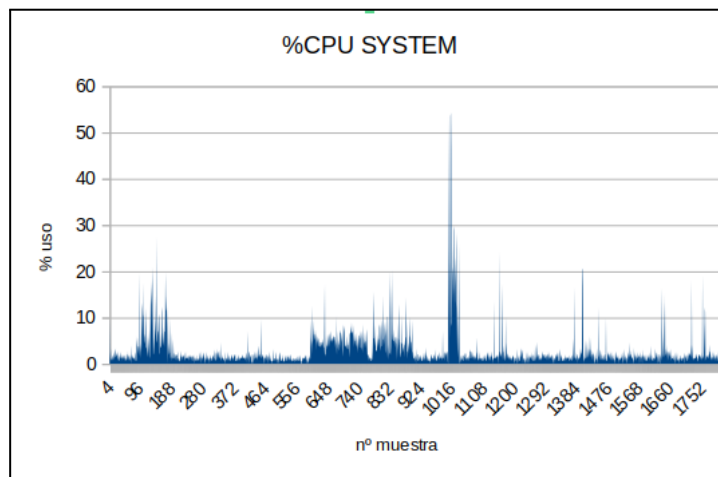
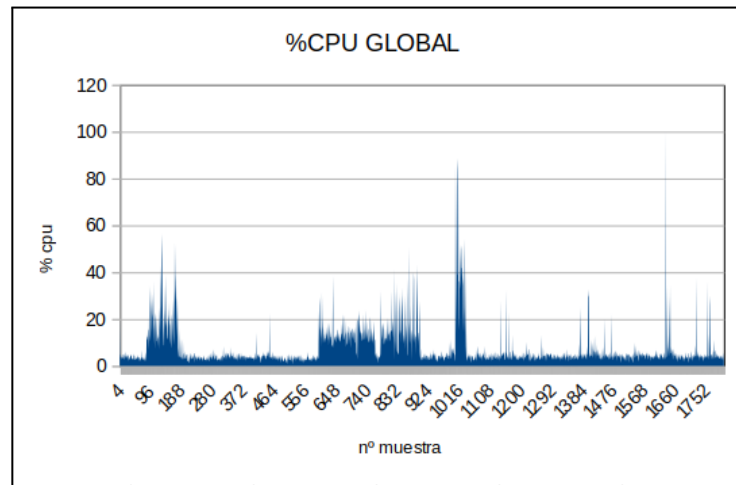
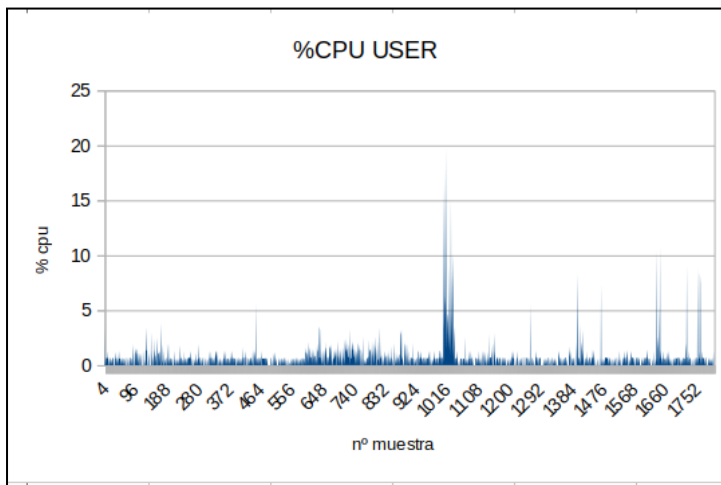
Toda la información (timestamp, %CPU (usuario), %CPU (sistema)) del archivo monitorizacionCPU.csv se extrae directamente del archivo generado por el comando top. Únicamente el porcentaje de CPU global implica el siguiente cálculo básico:

$$\%cpu (global) = 100 - (\$cpu \text{ en modo inactivo}) = 100 - (\%cpu id).$$

Ahora que tenemos los datos podemos calcular las medias:

% MEDIA CPU		
Global	User	System
7.504	0.628	2.865

3. ¿Cómo se comportan las medidas anteriores a lo largo del tiempo de observación? Muestra las tres métricas de forma gráfica



Como se puede observar en los gráficos, el usuario ha sido el que menos utilizaba la CPU, ya que durante la recopilación de estas muestras no estaba realizando ninguna tarea adicional en el sistema Linux aparte de la toma de muestras. Sin embargo, se observa un aumento notable después de aproximadamente 45 minutos desde el inicio de la toma de muestras.

Hablando del uso de CPU por parte del sistema, hay aproximadamente 3 intervalos con un aumento notable. El primero se produce después de 4 minutos (aproximadamente) de iniciar la toma de muestras, con un uso máximo alcanzado de 20.8%.

El segundo intervalo abarca desde la muestra número 594 (00:48:85) hasta la muestra número 894 (01:02:02), con una duración de 14 minutos. Finalmente, el tercer intervalo ocurre desde la muestra número 996 (01:05:52) hasta la muestra número 1033 (01:12:01), con un valor máximo igual a 54.2%.

Finalmente, para el gráfico de % cpu global se puede observar que es la unión de los otros dos gráficos (%CPU SYSTEM y %CPU USER). Sin embargo, en algunos intervalos de este gráfico los porcentajes son ligeramente mayores, cosa que puede indicar que se trataban de la ejecución tareas que no realizaba ni el sistema ni el usuario.

4. ¿Cuál es la sobrecarga provocada por el monitor TOP?

La fórmula para calcular la sobrecarga viene dada por:

$$\text{sobrecarga} = \frac{\text{Tiempo de ejecución del monitor}}{\text{Intervalo de medida}}$$

De los factores de esta fórmula solo conocemos el intervalo de medida(2s) y para obtener el tiempo de ejecución del monitor ejecutamos el comando **\$time top -b -n 1**.

Así que la sobrecarga es

$$\text{sobrecarga} = \frac{0.169}{2} = 0.0845 \rightarrow 8.45\%$$

```
real    0m0,169s
user    0m0,004s
sys     0m0,014s
```

II. Monitorización de la memoria principal

En esta segunda parte se nos pide monitorizar la memoria principal del sistema durante dos horas y con intervalos de muestreo de 5 segundos, y generar un fichero de la siguiente forma:

Timestamp	Capacidad disponible	Capacidad utilizada	% Memoria utilizada
...
...

1. ¿Qué capacidad total tiene la memoria principal del sistema? ¿De dónde se ha obtenido ese dato?

Ejecutando `cat /proc/meminfo` podemos obtener información relacionada con la memoria y específicamente la capacidad total de la memoria principal de nuestro sistema. En este caso es: **10946740 kB**.

```
khaoula@KIBVMd:~$ cat /proc/meminfo
MemTotal:      10946740 kB
MemFree:       5952672 kB
MemAvailable:  8443708 kB
Buffers:       236332 kB
Cached:        2460096 kB
SwapCached:    0 kB
```

2. ¿Cuál es la utilización media de la memoria? ¿Y la capacidad media utilizada?

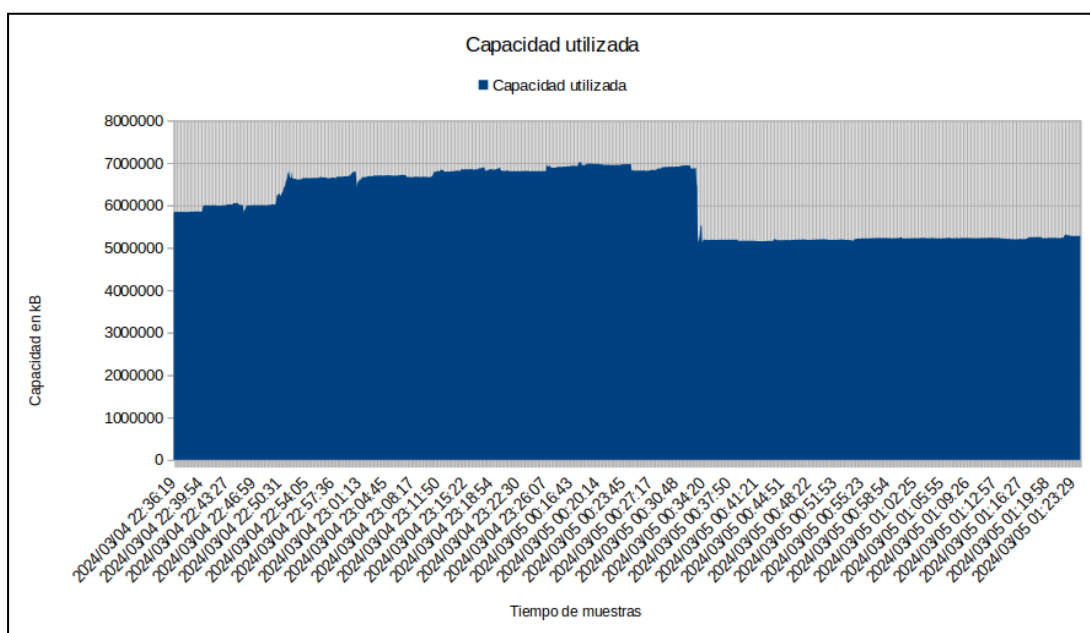
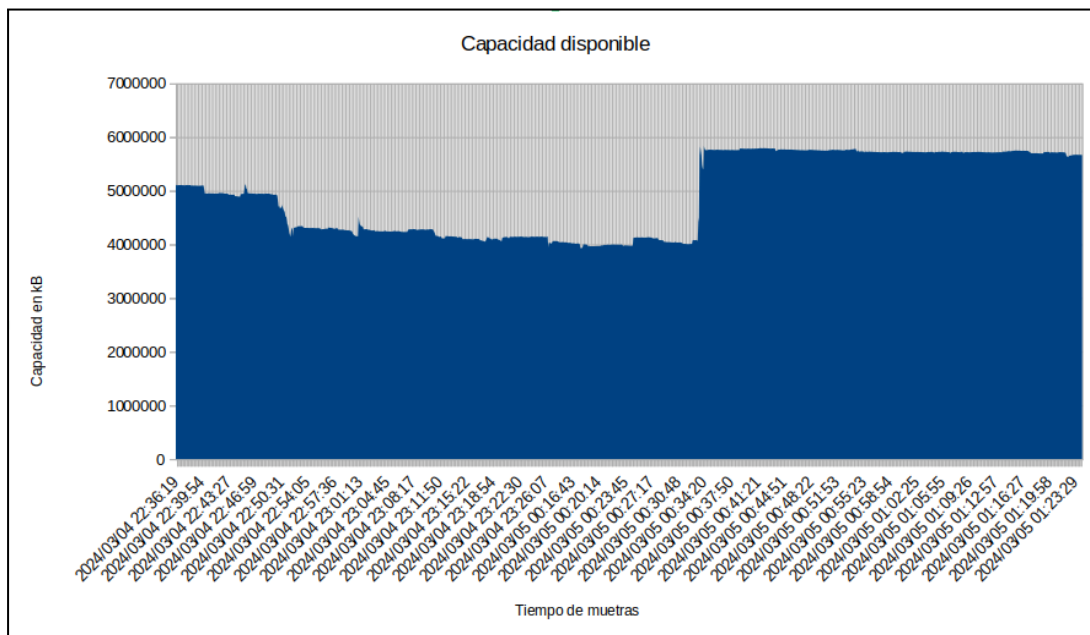
Para obtener estos datos, primero monitorizamos la memoria principal utilizando el comando `vmstat` durante 2 horas con intervalos de 5 segundos. Es decir, habrá $2 \text{ horas} \times 3600/5 = 1440$ intervalos de 5 segundos.

En nuestro terminal ejecutamos el script `getMemoryInfo.sh` para monitorizar la memoria y obtener la información relevante. Una vez que nuestro archivo está en formato `.csv` con las columnas (Timestamp, Capacidad disponible, Capacidad utilizada, % de memoria utilizada), lo abrimos con Excel y calculamos las medias de la memoria y de la capacidad utilizada. Los resultados se presentan en la siguiente tabla:

Capacidad disponible	Capacidad utilizada	% Memoria utilizada
4917444.81111	6029295.18889	55.08%

Como se puede observar, la capacidad utilizada es bastante alta, y supera la capacidad disponible, y esto principalmente se debe a que el usuario estaba realizando tareas que implicaban el uso de memoria (gráficas en Excel de la primera parte de la práctica, corrección de scripts de la monitorización de la CPU, creación, modificación y eliminación de los ficheros que iba generando al probar el código), esto también explica al alto porcentaje de uso de la memoria.

3. ¿Cómo se comporta la utilización de la memoria y la capacidad utilizada? Representa estas métricas gráficamente.



De una simple vista, se puede observar que las variaciones en la capacidad utilizada y la capacidad disponible son inversas; es decir, cuando aumenta el uso de la memoria, disminuye la memoria disponible. Esto es lógico, dado que hemos definido que la memoria utilizada es la diferencia entre la memoria total y la memoria disponible.

En cuanto a la memoria disponible, se observa un aumento significativo desde las 23:00 hasta aproximadamente las 00:34. Como se mencionó en la respuesta anterior, durante este período se estaban realizando tareas que requerían el uso de memoria, mientras que durante el resto del tiempo de toma de muestras, el portátil estaba en reposo. Esta es la razón detrás del mantenimiento del uso de la memoria en esos intervalos.

Finalmente, se nota un ligero aumento en el uso de la memoria a partir de la 01:19. En ese momento, estaba revisando el archivo de salida para determinar cuánto tiempo faltaba para completar la toma de todas las muestras.

4. ¿Cuál es la sobrecarga provocada por el monitor VMSTAT?

```

root@KIBVMd:/home/khaoula# bash GetMemoryInfo.sh
root@KIBVMd:/home/khaoula# time vmstat 1 1 -n
procs -----memory----- ---swap-- -----io----- -system-- -----cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo    in   cs  us  sy  id  wa  st
 2   0       0 5675028 145536 4008804    0    0    10    24   85  117  1  2 97  0  0

real    0m0,006s
user    0m0,004s
sys     0m0,002s
root@KIBVMd:/home/khaoula#

```

Siguiendo los mismos pasos en el apartado 4 de la parte anterior, la sobrecarga provocada por el monitor VMSTAT se calcula de la siguiente forma :

$$sobre carga = \frac{\text{Tiempo de ejecución del monitor}}{\text{Intervalo de medida}} = \frac{0.006}{5} = 0.0012 \rightarrow 0.12\%$$

III. Monitorización en paralelo

En esta tercera parte se pide monitorizar en paralelo la CPU y la memoria de nuestro sistema durante 2 horas y con una sobrecarga inferior al 10%.

Como se ha mencionado en ocasiones anteriores, la sobrecarga es equivalente al tiempo de ejecución del monitor dividida por el tiempo de intervalo de medida. Por lo tanto, hay que determinar los intervalos de medida que cumplen la siguiente inecuación:

$$\frac{10}{100} \geq \frac{\text{tiempo de ejecución del monitor}}{\text{intervalo de medida}}$$

Por lo tanto, primero obtenemos los tiempos de ejecución de los monitores TOP y VMSTAT:

\$time top -b -n 1

```
real    0m0,176s
user    0m0,007s
sys     0m0,015s
```

\$time vmstat 1 1 -n

```
real    0m0,004s
user    0m0,003s
sys     0m0,001s
```

Así que tenemos

$$\frac{10}{100} \geq \frac{0.176 + 0.004}{\text{intervalo muestreo}} \rightarrow 0.1 \geq \frac{0.18}{\text{intervalo muestreo}} \rightarrow \text{intervalo muestreo} \geq \frac{0.18}{0.1}$$

$\rightarrow \text{intervalo muestreo} \geq 1.8 \text{ segundos}$

Pues, podemos fijar los intervalos de medida de 2 segundos para obtener un número entero de intervalos que calcularemos a seguida:

$$2 \text{ h} = 2 * 3600 \text{ s} = 7200 \text{ s} \rightarrow 7200 / 2 = 3600 \text{ intervalos de monitoreo de 2 segundos.}$$

Una vez que hemos obtenido los datos que nos faltaban, podemos empezar la monitorización paralela de la CPU y la memoria principal. Por ello, ejecutaremos el siguiente comando en nuestro Shell: `$bash monitoreoParalelo.sh`.

El script `monitoreoParalelo.sh` invoca a los scripts utilizados para la monitorización tanto de la CPU como de la memoria principal. Luego, crea un único archivo de salida que combina los datos de los dos archivos generados por los scripts mencionados anteriormente, con las siguientes columnas:

Timestamp	% global CPU	Capacidad de memoria utilizada	% Memoria utilizada
...
...

Nota Importante

Para la primera y segunda parte de esta práctica se ha trabajado con la máquina virtual, pero debido a algunos problemas con el Ubuntu que no pude solucionar, tuve que crear una nueva máquina para completar la parte de monitorización paralela. Esto explica el cambio de memoria total en los scripts de **10946740 kB a 10946416 kB**.

IV. Pregunta voluntaria

Expresar la fórmula de la sobrecarga de la monitorización cuando dos (o más) monitores se están ejecutando en paralelo. Por ejemplo, se han lanzado los monitores \$top y \$vmstat a la vez.

Sea x el número de monitores que se ejecutan en paralelo. Dado que cada proceso se ejecuta y emplea los recursos y el espacio de memoria que se le asignan de forma independiente de los otros procesos, podemos decir que cada proceso aporta su propia carga al sistema y, por lo tanto, la sobrecarga total de la monitorización de x se puede calcular de la siguiente forma:

$$\text{sobrecarga} = \frac{\sum_{i=1}^x \text{tiempo de ejecución } i}{\text{intervalo de medida}}$$