

# **INGENIERÍA DE SERVIDORES**

## **Guión de la Práctica 3**

---

### **Monitorización de servicios**

---

**Curso 2014-2015**

# Índice

1.- Objetivos Mínimos .....	3
2.- Bibliografía .....	3
3.- Contenidos .....	3
4.- Monitorización de sistemas Linux .....	3
4.1.- Conociendo el subsistema de archivos .....	4
4.2.- Programación de tareas con cron .....	4
4.3.- Analizando qué ocurre en el kernel con dmesg .....	5
4.4.- Monitor general gnome-system-monitor .....	5
5.- Monitorizando Windows: perfmon.....	5
6.- Monitorizando el Hardware .....	6
7.- Otros monitores del Sistema .....	7
7.1.- Munin .....	7
7.2.- nagios.....	7
7.3.- Ganglia .....	7
7.4.- ZABBIX .....	7
7.5.- cacti.....	8
7.6.- AWstats .....	8
8.- Profiling.....	8
9.- Normativa .....	10
10.- Calificación .....	11

## 1.- Objetivos Mínimos

Los objetivos mínimos que el alumno debe alcanzar son:

- 1) Conocer y saber usar las herramientas que permitan obtener datos sobre el sistema a nivel hardware y software (SO y servicios)
- 2) Saber interpretar los resultados proporcionados por las aplicaciones de monitorización
- 3) Conocer los archivos que proporcionan información del sistema
- 4) Saber utilizar "profilers" para analizar el código de las aplicaciones y scripts programados.

## 2.- Bibliografía

La proporcionada en la asignatura y enlaces en el guión.

## 3.- Contenidos

Esta práctica consiste en la utilización de herramientas de monitorización del sistema para visualizar cómo ejerce el sistema ciertas actividades que los usuarios u otros servicios generan. En teoría verán el concepto de carga de modo que ya sabrán cómo "monitorizarla", es decir, ver cómo afecta ésta al sistema real.

Las herramientas de monitorización presentan medidas del sistema en tiempo real, permitiendo generar informes e históricos que puedan ser de utilidad para un análisis a posteriori (*offline*).

El objetivo de esta práctica es monitorizar varios sistemas mientras realizan ciertas tareas. Posteriormente, se deben **resumir y representar los resultados de un modo adecuado**.

Se da por hecho que si no tiene algún servicio instalado (p.ej. Intérprete de python) usted ya tiene el conocimiento para instalarlo. **En tales casos, indique cómo lo ha hecho.**

## 4.- Monitorización de sistemas Linux

Hay una serie de comandos que permiten ver el estado de la máquina y muestran el resultado en la consola. A continuación veremos algunos de ellos y el alumno, con la ayuda de las páginas del manual y las referencias proporcionadas en la presentación de la asignatura, deberán interpretar los resultados.

## 4.1.- CONOCIENDO EL SUBSISTEMA DE ARCHIVOS

En linux (UNIX) todo se manipula a través de archivos de una manera cómoda y transparente. Existe un directorio especial: /proc (visto en clase de teoría) y /var (algo se ha comentado también al respecto).

### **Cuestión 1:**

a) ¿En qué archivos se guarda registro de los paquetes instalados en sistema con los gestores de paquetes de Ubuntu y CentOS? Durante la práctica 2 instaló LAMP como un único paquete o instalando cada componente diferenciado. Busque en el archivo de registro las líneas correspondientes a la instalación y preséntelas.

b) En el directorio /var/log es común encontrar archivos con extensiones en formato <numero>.gz. Por ejemplo, .1.gz. Explique como se generan estos archivos y que relación guardan entre ellos

Con el programa **gnome-system-log** podemos tener acceso a todos estos archivos de una manera cómoda a través de una GUI.

### 4.1.1 Volviendo con el RAID1

Dentro de /proc existe un archivo para el estado de los *multidevice* como es el caso de nuestro /dev/md0 creado en la P1.

Vamos a monitorizar el proceso de sustituir un disco dañado por uno nuevo y cómo podemos saber cuándo el sistema está preparado para continuar operando con normalidad.

Con la máquina encendida, quite uno de los discos del RAID e inserte, posteriormente, uno nuevo (que estará completamente vacío). En las máquinas virtuales no puede retirar un disco activo, pero si puede marcarlo como defectuoso empleando el comando "mdadm".

Un modo de *monitorizar* el proceso de réplica mediante:

```
watch -n2 cat /proc/mdstat
```

**Cuestión 2:** Indique los pasos que ha seguido, comandos empleados y significado de los mismos. Junto a cada comando, presente las líneas del registro del RAID que son significativas en cada paso: indicación de fallo, reemplazo, inicio y finalización de la reconstrucción del RAID.

## 4.2.- PROGRAMACIÓN DE TAREAS CON CRON

El comando cron permite ejecutar cada cierto intervalo de tiempo una tarea concreta. Esto es muy útil de cara a recopilar información o monitorizar el sistema realizando una tarea concreta p.ej. Enviar un correo electrónico cuando la carga esté por encima de un valor determinado.

**Cuestión 3:** Añada a la configuración de cron una tarea que se ejecute diariamente y que copie una vez al día el contenido del directorio `~/codigo` a `~/seguridad/$fecha` donde `$fecha` es la fecha actual del sistema (puede usar el comando `date`). Otra tarea, se ejecutará una vez al mes y reunirá todos los directorios diarios creados para el mes pasado en un archivo `~/seguridad/dirCodigo.<numero>.gz`. Presente las líneas de configuración de cron afectadas, explicando su significado. Si crea ficheros por lotes, presente y explique el código.

### 4.3.- ANALIZANDO QUÉ OCURRE EN EL KERNEL CON DMESG

El *kernel* de Linux permite conocer qué actividad ha ocurrido gracias a los mensajes que proporciona el kernel. Esto es especialmente útil para detectar problemas con el HW o periféricos.

**Cuestión 4:**

- a) Pruebe a ejecutar el comando, conectar un dispositivo USB y vuelva a ejecutar el comando. Copie y pegue las líneas que hacen mención al dispositivo conectado (considere usar `dmesg | tail`).
- b) Explique las diferencias (si las hay) entre o consultar el contenido del archivo `/var/log/dmesg`?

### 4.4.- MONITOR GENERAL GNOME-SYSTEM-MONITOR

Además de los comandos comentados previamente, existe una aplicación que permite monitorizar tanto los procesos como los sistemas de memoria y de red: `gnome-system-monitor`.

## 5.- Monitorizando Windows: perfmon

En Windows, el equivalente al programa `top` es el "Administrador de tareas" que puede invocarse tras pulsar `Ctrl+Alt+Supr`.

Sin embargo, existe una aplicación mucho más potente que nos permite controlar el estado del servidor de manera visual así como recopilar datos para su posterior uso. Su nombre es `perfmon`. Esta herramienta se encarga de recopilar los datos proporcionados por otros sistemas dentro de Windows.

Para iniciar `perfmon`, podemos abrir una consola y ejecutar el comando con el mismo nombre (para una lista con los comandos disponibles, visite: <http://technet.microsoft.com/en-us/library/cc772390%28v=ws.10%29.aspx>)

Las características detalladas del software se pueden consultar en <http://technet.microsoft.com/en-us/library/cc749249.aspx?ppud=4> aunque en el

mismo programa, haciendo click en "Obtener más información" también obtendrá la documentación correspondiente.

Nos centraremos en la creación de un recopilador de datos y en la realización de un informe de diagnóstico.

Tras ejecutar el monitor, en el panel de la izquierda encontraremos "Conjunto de recopiladores de datos" y dentro de la subsección veremos unos definidos por el usuario (debe estar vacío) y otros definidos por el sistema. Pulsando sobre uno de éstos con el botón derecho, podemos iniciarlos (puesto que se encuentran detenidos por defecto) y, tras esperar un minuto, se generará automáticamente un informe.

### **Cuestión 5:**

- a) Ejecute el monitor de "System Performance" y muestre el resultado. Incluya capturas de pantalla y comente la información que aparece.
- b) Cree un recopilador de datos definido por el usuario (modo avanzado) que incluya tanto el contador de rendimiento como los datos de seguimiento:
  - Todos los referentes al procesador, al proceso y al servicio web.
  - Intervalo de muestra 15 segundos
  - Almacene el resultado en el directorio Escritorio\logsIncluya las capturas de pantalla de cada paso.

## **6.- Monitorizando el Hardware**

Además del estado del software del servidor, también existen programas que nos permiten ver el estado del hardware de nuestra máquina. En primer lugar, muchas BIOS nos permiten acceder a cierta información sobre el estado del HW, sin embargo, para no tener que reiniciar, podemos utilizar estas herramientas:

*Para Linux:*

Para la temperatura del HD tenemos: [hddtemp](#)  
Proyecto lm-sensors: <http://lm-sensors.org/>  
que posee una GUI: xsensors

*Para Windows:*

- Open Hardware Monitor: <http://openhardwaremonitor.org/> (también funciona bajo Linux aunque depende de Mono [http://www.mono-project.com/Main\\_Page](http://www.mono-project.com/Main_Page))
- SpeedFan: <http://www.almico.com/speedfan.php>
- RealTemp: <http://www.techpowerup.com/realtemp/>
- Core Temp: [www.alcpu.com/CoreTemp/](http://www.alcpu.com/CoreTemp/)
- CPUID: <http://www.cpubid.com/software/hwmonitor.html>
- Speccy: <http://www.piriform.com/speccy>
- HD Tune <http://www.hdtune.com/>
- HW Info <http://www.hwinfo.com/>

## 7.- Otros monitores del Sistema

Además de los comandos integrados vistos en teoría y lo que ha manejado en prácticas, existen otros programas muy populares que permiten monitorizar el sistema.

Hay algunos de entorno corporativo de grandes empresas como **NetApp** (<http://www.netapp.com/es/products/management-software/>), aunque los que se verán en las siguientes subsecciones también son usados por grandes instituciones y empresas.

### 7.1.- MUNIN

Munin (significa memoria) está disponible en <http://munin-monitoring.org/>

Para su instalación (en CentOS) puede hacerlo compilando el código fuente (como ha podido hacer con `lm_sensors`) alojado en la página o a través del paquete disponible en el repositorio EPEL (<http://fedoraproject.org/wiki/EPEL/es>). "¿Cómo puedo utilizar estos paquetes adicionales?" es el título de la subsección dentro de la página donde se explica cómo activar el repositorio que contiene los paquetes. Tan solo debe instalar un `.rpm` y podrá usar `yum` para instalar `munin`.

Para Ubuntu, está disponible sin tener que añadir ningún repositorio adicional.

### 7.2.- NAGIOS

Es otro software muy usado para monitorizar sistemas. En su página web <http://www.nagios.org/> se encuentra más información así como diversos tutoriales sobre cómo instalarlo en CentOS.

### 7.3.- GANGLIA

Es un proyecto alojado en <http://ganglia.sourceforge.net/> que monitoriza sistemas de cómputo distribuidos (normalmente para altas prestaciones) y permite una gran escalabilidad.

Como puede verse en su página, importantes instituciones y compañías lo usan (p.ej. UC Berkely, MIT, Twitter, ...).

### 7.4.- ZABBIX

Es otro programa que permite monitorizar el sistema también de código abierto. Su instalación es muy sencilla ya que solo hay que añadir los repositorios (visto en la práctica anterior) e instalarlo con el gestor de paquetes

## 7.5.- CACTI

Cacti ( <http://www.cacti.net/> ) Es un *front-end* para **RRDtool** ( <http://oss.oetiker.ch/rrdtool/> ) que permite monitorizar sin perjudicar mucho el rendimiento del sistema muchos parámetros.

## 7.6.- AWSTATS

Es un monitor de servicios específicos de servidores web p.ej.: http, ftp, correo. La página del proyecto es <http://awstats.sourceforge.net/>. Se puede compilar el código fuente aunque están disponibles los paquetes en los repositorios.

**Cuestión 6:** Elija uno de los sistemas de monitorización descritos en este apartado e instálelo. Describa los pasos seguidos así como posibles incidencias en la instalación que ha debido resolver. Monitorice uno o varios de sus servidores y presente ejemplos de algunas medidas que considere significativas, explicando su significado y los valores reales observados.

## 7.7 MONITORIZANDO UN SERVICIO (o ejecución de un programa)

Hay un conjunto de programas que permiten hacer una traza de las llamadas al sistema realizadas por un programa (servicio) en ejecución, p.ej. `strace` (system call tracer).

Este tipo de programas pueden ser útiles de cara a detectar problemas que no se muestran en los archivos de "log".

En <http://chadfowler.com/blog/2014/01/26/the-magic-of-strace/> , [https://www.debian-administration.org/article/352/Using\\_strace\\_to\\_debug\\_application\\_errors](https://www.debian-administration.org/article/352/Using_strace_to_debug_application_errors)

y

[http://blog.softlayer.com/2013/sysadmin-tips-and-tricks-using-strace-to-monitor-system-calls#utm\\_source=twitter&utm\\_medium=social&utm\\_content=beyond-the-command-line-with-strace&utm\\_campaign=blog\\_development-tips-and-tricks](http://blog.softlayer.com/2013/sysadmin-tips-and-tricks-using-strace-to-monitor-system-calls#utm_source=twitter&utm_medium=social&utm_content=beyond-the-command-line-with-strace&utm_campaign=blog_development-tips-and-tricks)

tiene ejemplos de uso y podrá apreciar la utilidad de este tipo de programas.

## 8.- Profiling



“Profiling” podría considerarse como una monitorización estática en el sentido de que no estamos controlando el comportamiento de un sistema en un momento concreto sino la ejecución de un programa/script/consulta/operación en el momento en el que lo deseemos.

Es un matiz suave que en lengua inglesa y dentro del ámbito tecnológico (el término profiler no está recogido en el diccionario Cambridge) hace que los términos “monitor” y “profiling” sean distintos.

En este apartado vamos a ver algunos programas que permiten analizar el comportamiento de un elemento de ejecución concreto de modo que podamos ver qué parte es menos eficiente y dónde se puede mejorar el rendimiento.

## **8.1 Ejecución de Programas**

### **8.1.1 GPROF y VALGRIND**

Podemos ver cómo se ejecuta un programa y ver cuánto tiempo pasa en cada función con la utilidad gprof.

La documentación detallada se encuentra en:

<https://sourceware.org/binutils/docs/gprof/> (página mantenida por Red Hat)

donde hay un tutorial con ejemplos de uso muy detallado y bien descrito.

Otro programa muy bueno, interesante y popular de cara al “profiling” de un programa compilado es Valgrind: <http://valgrind.org/> que posee GUIs, permite seguir las hebras, etc.

Por cuestión de tiempo, no podemos detenernos en probar estos programas. Sin embargo debe saber que con estos programas podríamos analizar el comportamiento de páginas web escritas en C++ (usando p.ej. <http://www.webtoolkit.eu/wt> y <http://cppcms.com/wikip/en/page/main> )

### **8.1.2 PHP**

Para estudiar el comportamiento de los scripts que desarrollemos podemos usar varios profilers:

- Xdebug: <http://www.xdebug.org/index.php>
- XHProf: <http://pecl.php.net/package/xhprof>  
<https://github.com/facebook/xhprof>

Además de la documentación oficial, en la página de un prestigioso sistema de enseñanza virtual hay unos tutoriales interesantes sobre cómo realizar el profiling.

- [http://docs.moodle.org/dev/Profiling\\_PHP](http://docs.moodle.org/dev/Profiling_PHP)

### 8.1.3 PYTHON

Para hacer “profiling” de los scripts escritos en este lenguaje se puede utilizar:

- <http://docs.python.org/2/library/profile.html>

### 8.1.4 POWERSHELL

Al igual que con los otros lenguajes de script, PowerShell posee un “profiler”: PoshProfiler

- <http://gallery.technet.microsoft.com/Powershell-script-profiler-4382ffad>

## 8.2 Bases de Datos

### 8.2.1 MySQL

El mismo MySQL que instalamos en la práctica anterior tiene un “profiler” para analizar cuánto tardan las consultas. Puede ver la documentación en:

- <http://dev.mysql.com/doc/refman/5.0/en/show-profiles.html>
- <http://dev.mysql.com/doc/refman/5.0/en/show-profile.html>

### 8.2.2 MongoDB

Al igual que MySQL, MongoDB también trae integrado un “profiler”:

- <http://docs.mongodb.org/manual/tutorial/manage-the-database-profiler/>

**Cuestión 7:** Diseñe un pequeño modelo de BBDDs para un problema de su elección e impleméntelo en MySQL. También puede emplear un sistema de código abierto del que conozca su diseño de BBDDs. Plantee, una combinación de consultas (al menos dos) que considere significativas y explique los resultados obtenidos en su “profile”. Se valorará especialmente, que sea capaz de introducir cambios en el diseño de tablas o en las consultas que mejoren los resultados y que sepa justificar la mejora.

## 9.- Normativa

La especificada en la guía de prácticas.

## **10.- Calificación**

La práctica se calificará entre 0 y 10 empleando los criterios descritos en al guía de prácticas.