

INGENIERÍA DE SERVIDORES (2016-2017)
GRADO EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

Memoria Práctica 3

Adrián Morente Gabaldón

5 de diciembre de 2016

Índice

1	a) ¿Qué archivo le permite ver qué programas se han instalado con el gestor de paquetes? b) ¿Qué significan las terminaciones <i>1.gz</i> o <i>2.gz</i> de los archivos en ese directorio?	4
1.1	Consulta de los paquetes instalados con APT	4
1.2	Archivos <i>1.gz</i> y <i>2.gz</i> del gestor APT	5
2	¿Qué archivo ha de modificar para programar una tarea? Escriba la línea necesaria para ejecutar una vez al día una copia del directorio <code>/codigo a /seguridad/\$fecha</code> donde <code>\$ fecha</code> es la fecha actual (puede usar el comando <code>date</code>).	5
3	Pruebe a ejecutar el comando, conectar un dispositivo USB y vuelva a ejecutar el comando. Copie y pegue la salida del comando. (considere usar <code>dmesg tail</code>). Comente qué observa en la información mostrada.	6
4	Ejecute el monitor de “System performance” y muestre el resultado. Incluya capturas de pantalla comentando la información que aparece.	8
5	Cree un recopilador de datos definido por el usuario (modo avanzado) que incluya tanto el contador de rendimiento como los datos de seguimiento: 1) Todos los referentes al procesador, al proceso y al servicio web. 2) Intervalo de muestra 15 segundos. 3) Almacene el resultado en el directorio <code>Escritorio/logs</code> . 4) Incluya las capturas de pantalla de cada paso.	8
6	Visite la web del proyecto y acceda a la demo que proporcionan (http://demo.munin-monitoring.org/) donde se muestra cómo monitorizan un servidor. Monitoree varios parámetros y haga capturas de pantalla de lo que está mostrando comentando qué observa.	8
7	Escriba un breve resumen sobre alguno de los artículos donde se muestra el uso de <code>strace</code> o busque otro y coméntelo.	8
8	Escriba un script en Python o PHP y analice su comportamiento usando el profiler presentado.	9
9	Acceda a la consola MySQL (o a través de phpMyAdmin) y muestre el resultado de mostrar el “profile” de una consulta (la creación de la BD y la consulta la puede hacer libremente).	9

Índice de figuras

1.1.	Consulta de los paquetes instalados con APT en el directorio <code>/var/log/apt</code> . - Adrián Morente Gabaldón [04/12/2016]	4
------	---	---

1.2. Contenido del archivo <i>term.log</i> y los resultados de la ejecución de APT. - Adrián Morente Gabaldón [04/12/2016]	5
3.1. Muestra de los resultados recopilados del kernel por <i>dmesg</i> . - Adrián Morente Gabaldón [04/12/2016]	7

Índice de tablas

1. a) ¿Qué archivo le permite ver qué programas se han instalado con el gestor de paquetes? b) ¿Qué significan las terminaciones *1.gz* o *2.gz* de los archivos en ese directorio?

1.1. Consulta de los paquetes instalados con APT

Como comentamos siempre en clase de prácticas, una de las cosas que tenemos que revisar como administradores de sistemas son los archivos *.log*, los cuales contienen frecuentemente información relevante sobre paquetes instalados o servicios ofrecidos. En este caso, podemos consultar los paquetes instalados de APT a través del archivo ubicado en */var/log/apt/history.log*. En la misma carpeta podemos encontrar también el archivo *term.log*, que contiene los resultados de instalación, consulta y borrado de paquetes realizados por APT. En la siguiente imagen, vemos el contenido del directorio mencionado, e imprimimos el contenido del fichero *history.log* incluido en tal sitio; ya que en él encontramos la información pedida por el enunciado:

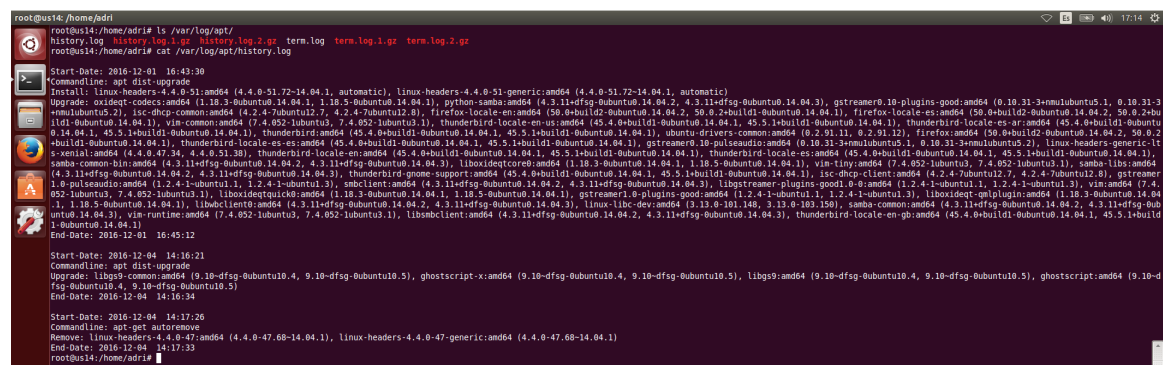


Figura 1.1: Consulta de los paquetes instalados con APT en el directorio /var/log/apt. - Adrián Morente Gabaldón [04/12/2016]

En cuanto a la extensión del contenido del archivo, que vemos que es muy reducida, hablaremos de ello en el siguiente apartado del ejercicio. En este caso, podemos ver en los apartados **Commandline: apt ...** que las últimas acciones que realicé en la administración de Ubuntu Server fueron las actualizaciones del sistema, con los consiguientes borrados automáticos de programas y/o paquetes innecesarios.

A continuación, vemos parte del contenido del archivo *term.log*, que concuerda con las actualizaciones y borrados realizados arriba, ya que aquí se recogen los resultados de dichas acciones:

```
root@us14:/home/adri# tail -n 30 /var/log/apt/term.log
Configurando liboxideqtcore0:amd64 (1.18.5-0ubuntu0.14.04.1) ...
Configurando liboxideqtquick0:amd64 (1.18.5-0ubuntu0.14.04.1) ...
Configurando liboxideqt-qmlplugin:amd64 (1.18.5-0ubuntu0.14.04.1) ...
Procesando disparadores para libc-bin (2.19-0ubuntu6.9) ...
Log ended: 2016-12-01 16:45:12

Log started: 2016-12-04 14:16:21
(Leyendo la base de datos ... 287296 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../ghostscript 9.10-dfsg-0ubuntu10.5 amd64.deb ...
Desempaquetando ghostscript (9.10-dfsg-0ubuntu10.5) sobre (9.10-dfsg-0ubuntu10.4) ...
Preparando para desempaquetar .../ghostscript-x 9.10-dfsg-0ubuntu10.5 amd64.deb ...
Desempaquetando ghostscript-x (9.10-dfsg-0ubuntu10.5) sobre (9.10-dfsg-0ubuntu10.4) ...
Preparando para desempaquetar .../libs9-common 9.10-dfsg-0ubuntu10.5 all.deb ...
Desempaquetando libs9-common (9.10-dfsg-0ubuntu10.5) sobre (9.10-dfsg-0ubuntu10.4) ...
Preparando para desempaquetar .../libs9 9.10-dfsg-0ubuntu10.5 amd64.deb ...
Desempaquetando libs9 (9.10-dfsg-0ubuntu10.5) sobre (9.10-dfsg-0ubuntu10.4) ...
Procesando disparadores para man-db (2.6.7.1-1ubuntu1) ...
Configurando libs9-common (9.10-dfsg-0ubuntu10.5) ...
Configurando libs9 (9.10-dfsg-0ubuntu10.5) ...
Configurando ghostscript (9.10-dfsg-0ubuntu10.5) ...
Configurando ghostscript-x (9.10-dfsg-0ubuntu10.5) ...
Procesando disparadores para libc-bin (2.19-0ubuntu6.9) ...
Log ended: 2016-12-04 14:16:34

Log started: 2016-12-04 14:17:26
(Leyendo la base de datos ... 287295 ficheros o directorios instalados actualmente.)
Desinstalando linux-headers-4.4.0-47-generic (4.4.0-47.68-14.04.1) ...
dpkg: aviso: al desinstalar linux-headers-4.4.0-47-generic, el directorio «/lib/modules/4.4.0-47-generic» no está vacío, por lo que no se borra
Desinstalando linux-headers-4.4.0-47 (4.4.0-47.68-14.04.1) ...
Log ended: 2016-12-04 14:17:33
root@us14:/home/adri#
```

Figura 1.2: Contenido del archivo *term.log* y los resultados de la ejecución de APT. - Adrián Morente Gabaldón [04/12/2016]

1.2. Archivos *1.gz* y *2.gz* del gestor APT

Como podemos ver en los manuales oficiales de GNU [3], los archivos con extensión *.gz* son archivos comprimidos manejados por sistemas operativos basados en Unix. El gestor de paquetes APT deposita el historial y los resultados de sus ejecuciones en los archivos *history.log* y *term.log* respectivamente, como hemos visto antes; pero conforme esos archivos comienzan a tener un tamaño elevado, el sistema decide comprimir su contenido para ahorrar espacio. Por ejemplo, para el archivo *history.log*, comprimirá todo su contenido en un nuevo archivo *history.log.X.gz* siendo X el número de veces que se ha realizado esta acción. Es decir, si el sistema ha hecho esto tres veces, tendremos los tres archivos *history.log.1.gz*, *history.log.2.gz* y *history.log.3.gz*; siendo siempre más reciente aquel con numeración menor.

2. ¿Qué archivo ha de modificar para programar una tarea? Escriba la línea necesaria para ejecutar una vez al día una copia del directorio */codigo* a */seguridad/\$fecha* donde \$ fecha es la fecha actual (puede usar el comando *date*).

Para informarnos sobre el uso de *cron* (herramienta ya utilizada en la asignatura de Sistemas Operativos) podemos utilizar el manual a través de la línea de comandos, aunque en mi caso utilizaré también la documentación oficial de Oracle al respecto [1], que en mi caso es más ilustrativa para el uso de dicha herramienta.

Como vemos en la web, el archivo modificado al programar una tarea es */var/spool/cron/-crontabs*, pero es preferible no modificarlo directamente; sino hacerlo a través de la ejecución de la herramienta *crontab*, como haremos en este caso práctico. Para empezar, creare-

mos el script *script_seguridad.sh* que después ejecutaremos periódicamente, que tendrá el siguiente contenido:

```
#!/bin/bash
#guarda la fecha actual en la variable
date=$(date '+%Y%m%d_%H%M%S')
#si no existe, crea el directorio (primera vez)
mkdir -p /seguridad/$date_time
#copia los archivos de /codigo a /seguridad/fecha
cp -r ~/codigo/* ~/seguridad/$date_time
```

A continuación, daremos permiso de ejecución al script (*chmod u+x script_seguridad.sh*) y crearemos varios ficheros de texto plano en el directorio */codigo* con el contenido de prueba que queramos.

Antes de programar el archivo, volvamos a mirar la documentación de Oracle para “refrescar la memoria” en cuanto a la sintaxis de éstos [4]; y veremos que el orden de sus parámetros es:

<minuto> <hora> <día_mes> <mes> <día_semana> <comandos>

Para terminar, crearemos el archivo *cron* pero tal y como hemos dicho antes: sin acceder al fichero directamente sino a través de la herramienta *crontab*. Como queremos que se ejecute una vez al día, podemos fijar que sus parámetros de ejecución sean siempre a las 5 de la tarde, lo cual sería de la siguiente forma:

```
crontab -e {0 17 * * * ~/. / script_seguridad.sh}
```

3. Pruebe a ejecutar el comando, conectar un dispositivo USB y vuelva a ejecutar el comando. Copie y pegue la salida del comando. (considere usar *dmesg* | *tail*). Comente qué observa en la información mostrada.

Como sabemos, el comando *dmesg* puede ser muy útil para detectar problemas o cambios en el hardware conectado al kernel de un sistema; su uso es sencillo y da pie a muchas opciones (apenas usadas), según vemos en el manual del proyecto *Linux Information Project* [2].

Pasemos a ejecutar el comando *dmesg* | *tail* para mostrar parte de esta información, y procedemos a insertar un pendrive USB en el ordenador, que seleccionamos a través del menú “Dispositivos” de la máquina virtual en VirtualBox de forma que el sistema virtual lo reconozca. Hecho esto, nos encontraremos con la siguiente imagen, en la que podemos ver cómo el kernel pasa a detectar el pendrive insertado, dándonos información del tipo de dispositivo (*USB DISK 2.0*), el tamaño total (16GiB, 14.9GiB utilizables, en mi caso), el tamaño de cada bloque de datos en bytes (512B), y el modo de protección de escritura de datos (desactivado, ahora mismo) entre otros.

```

(dom dic 04-21:30:02)-{adri@us14:~}$ dmesg | tail
[14976.983499] usb 1-1: USB disconnect, device number 3
[14976.989042] e1000: eth0 NIC Link is Down
[14977.586889] usb 1-1: new full-speed USB device number 4 using ohci-pci
[14977.840725] usb 1-1: New USB device found, idVendor=80ee, idProduct=0021
[14977.840733] usb 1-1: New USB device strings: Mfr=1, Product=3, SerialNumber=0
[14977.840737] usb 1-1: Product: USB Tablet
[14977.840741] usb 1-1: Manufacturer: VirtualBox
[14977.856986] input: VirtualBox USB Tablet as /devices/pci0000:00/0000:00:06.0/usb1/1-1/1-1:1.0/0003:80EE:0021.0003/input/input9
[14977.912191] hid-generic 0003:80EE:0021.0003: input,hidraw0: USB HID v1.10 Mouse [VirtualBox USB Tablet] on usb-0000:00:06.0-1/input0
[14983.003135] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
(dom dic 04-21:34:50)-{adri@us14:~}$
(dom dic 04-21:35:12)-{adri@us14:~}$ dmesg | tail
[20829.069093] usbcore: registered new interface driver uas
[20830.074669] scsi 4:0:0:0: Direct-Access          USB DISK 2.0           PMAP PQ: 0 ANSI: 6
[20830.077842] sd 4:0:0:0: Attached scsi generic sg3 type 0
[20830.096554] sd 4:0:0:0: [sd] 31326208 512-byte logical blocks: (16.0 GB/14.9 GiB)
[20830.105547] sd 4:0:0:0: [sd] Write Protect is off
[20830.105550] sd 4:0:0:0: [sd] Mode Sense: 23 00 00 00
[20830.115058] sd 4:0:0:0: [sd] No Caching mode page found
[20830.115062] sd 4:0:0:0: [sd] Assuming drive cache: write through
[20830.181894]  sdc: sdc1
[20830.237106] sd 4:0:0:0: [sd] Attached SCSI removable disk
(dom dic 04-21:35:15)-{adri@us14:~}$

```

Figura 3.1: Muestra de los resultados recopilados del kernel por *dmesg*. - Adrián Morente Gabaldón [04/12/2016]

4. Ejecute el monitor de “System performance” y muestre el resultado. Incluya capturas de pantalla comentando la información que aparece.
5. Cree un recopilador de datos definido por el usuario (modo avanzado) que incluya tanto el contador de rendimiento como los datos de seguimiento: 1) Todos los referentes al procesador, al proceso y al servicio web. 2) Intervalo de muestra 15 segundos. 3) Almacene el resultado en el directorio Escritorio/logs. 4) Incluya las capturas de pantalla de cada paso.
6. Visite la web del proyecto y acceda a la demo que proporcionan (<http://demo.munin-monitoring.org/>) donde se muestra cómo monitorizan un servidor. Monitorice varios parámetros y haga capturas de pantalla de lo que está mostrando comentando qué observa.
7. Escriba un breve resumen sobre alguno de los artículos donde se muestra el uso de strace o busque otro y coméntelo.

En mi caso, haré un análisis del segundo artículo ofrecido en la práctica, proveniente del blog oficial de **SoftLayer, una compañía de IBM** [5].

Para empezar, plantean una descripción que nos sitúa en el marco de uso de la herramienta *strace*, haciéndonos ver que su utilización va orientada a los administradores de sistemas (en este caso, nosotros); y que a través de su ejecución podemos analizar y corregir los errores de los servicios que ofrecemos mostrándonos las llamadas al sistema y las señales resultantes de la ejecución de dichos servicios. Gracias a esto, a través de los archivos *.log* podemos ver cuál de los parámetros no están bien configurados o qué paquetes están fallando de alguna forma.

descartar que sea una herramienta para programadores, ya que está destinada a los administradores de sistemas (en este caso, nosotros); dado que su utilización va orientada a la detección y solución de errores en la ejecución de los servicios que ofrecemos en nuestro sistema. Para poder mostrarnos cuál de nuestros programas está fallando, o qué está realizando cada uno de ellos, esta herramienta nos muestra las llamadas y las señales del sistema resultantes de dichos programas.

8. **Escriba un script en Python o PHP y analice su comportamiento usando el profiler presentado.**
9. **Acceda a la consola MySQL (o a través de phpMyAdmin) y muestre el resultado de mostrar el “profile” de una consulta (la creación de la BD y la consulta la puede hacer libremente).**

Referencias

- [1] Creación y edición de archivos crontab - guía oficial de oracle. <https://docs.oracle.com/cd/E19455-01/805-7229/6j6q8svfo/index.html>. Consultado en 04/12/2016.
- [2] El comando dmesg y sus opciones - guía oficial del linux information project. <http://www.linfo.org/dmesg.html>. Consultado en 04/12/2016.
- [3] Gnu gzip, compresión de ficheros en unix - página oficial de gnu. <https://www.gnu.org/software/gzip/manual/gzip.html>. Consultado en 04/12/2016.
- [4] Sintaxis de un archivo crontab - guía oficial de oracle. <https://docs.oracle.com/cd/E19455-01/805-7229/6j6q8svfm/index.html#sysrescron-62861>. Consultado en 04/12/2016.
- [5] Using strace to monitor system calls - softlayer blog. <http://blog.softlayer.com/2013/sysadmin-tips-and-tricks-using-strace-to-monitor-system-calls>. Consultado en 04/12/2016.