[Cod.: CC481 Curso: Administración de Redes]
[Tema: Monitorización del sistema]
[Prof.: José Manuel Castillo Cara]

Laboratorio dirigido 4

Monitorización del sistema

Instrucciones:

1. Fecha de entrega será antes del **domingo a las 23:59**.
2. El formato de entrega será *pdf*.
3. El laboratorio tendrá una puntuación sobre 20.
4. En el solucionario se deberá copiar los enunciados y dejarlos en negrita.
5. La primera hoja será para la portada que se especificará, el número de laboratorio, nombre y apellidos de los integrantes, nombre de la asignatura y el escudo de la UNI. La segunda página será el índice. Donde se deberá tener en cuenta la página de cada Actividad.
6. Las citas y extracciones realizadas de Internet se deberán especificar. **No se corregirá el laboratorio en caso de copiar y pegar fragmentos sin especificar su fuente.**
7. Utilizar letra clara y adecuada a un documento técnico con tamaño 12 y márgenes superior e inferior de 3 cm y laterales de 2,5 cm.
8. Se corregirá la claridad y exactitud de la pregunta, en ningún caso se expondrán fundamentos no preguntados, además de la claridad del documento.
9. En las actividades realice una captura de pantalla mínimo por actividad para verificar su autoría.

Objetivos

En esta práctica estudiaremos los aspectos básicos sobre la monitorización del sistema que incluye principalmente la gestión y monitorización de procesos y los archivos de *log* del sistema. Además veremos cómo automatizar la ejecución de tareas.

Contenidos

- Preparando el entorno...
- Monitorización y Gestión de Procesos
  - Control de Trabajos en la *Shell*
  - Monitorización de procesos
  - Automatización de procesos
- *Logs* del Sistema

1. Preparando el entorno

En esta práctica necesitaremos dos máquinas virtuales:

- El *router* virtual para acceder a Internet y los repositorios de paquetes.
- Una máquina estación de trabajo donde realizaremos las pruebas.

2. Monitorización y Gestión de Procesos

En primer lugar estudiaremos cómo monitorizar los procesos que hay en ejecución en el sistema y los recursos que consumen. Esta sección está dividida en dos partes: la gestión de las tareas iniciadas en la *shell*; y los procesos de sistema.

2.1. Control de Trabajos en la Shell

El control de trabajos iniciados en la *shell* permite consultar, detener y reanudar su ejecución. La *shell* mantiene una lista de trabajos arrancados que puede consultarse con el comando *jobs*. Cada trabajo tiene un identificador, un *PID* del proceso asociado y la orden de ejecución.

Actividad 1

1. Para probar diferentes acciones sobre los procesos vamos a usar un comando gráfico que permita comprobar fácilmente el estado del proceso asociado.

   Escribir el siguiente contenido en un archivo (*progreso.sh*) y darle permisos de ejecución.

```
#!/bin/bash
   ( for i in $(seq 00 99) ; do
       echo $i
       sleep 1
   done ) |
zenity --progress
```

2. Ejecutar el script anterior y comprobar su funcionamiento.
3. Para detener la ejecución terminar un proceso se envía una señal. La combinación ***Ctrl+c*** envía la señal de interrupción (***SIGINT***) que interrumpe el proceso y ***Ctrl+z*** lo detiene (***SIGSTOP***). Comprobar el efecto de ambos sobre el comando.
4. Ejecuciones en primer y segundo plano. Cuando ejecutamos un comando decimos que está en primer plano (*foreground*) y puede recibir las señales generadas por teclado (ej. ***Ctrl+c***). Equivalentemente, un proceso en segundo plano mantiene su

ejecución pero desvincula su *ID* de grupo de procesos del de la *shell*. Los comandos ***fg*** y ***bg*** permiten poner en primer (*foreground*) y segundo plano (*background*) un proceso, respectivamente:

1. Enviar un proceso a segundo plano: (1) arrancar *progreso.sh*; (2) *Ctrl+z*; (3) *bg*.
2. Recuperar la ejecución en primer plano y terminar su ejecución *Ctrl+c*
3. ***bg*** y ***fg*** admiten especificar el trabajo que modifican de varias formas (***%n***). Usar esta funcionalidad cuando hay varios procesos en segundo plano o suspendidos. Ver la lista con jobs (opción ***-l***).
4. Consultar la página de manual *bash* en releación a ***jobs***, ***fg*** y ***bg***.

5. Es posible arrancar un proceso en segundo plano directamente usando **&** al final. Arrancar el comando de prueba en segundo plano y comprobarlo con *jobs*.
6. Terminal de control. Los procesos tienen asociados un terminal de control al que se envían las salidas estándar y de error y que recoge la entrada estándar si es necesaria. Ejecutar el siguiente comando (***sleep 1; date ; cat -> /tmp/entrada; sleep 1; date***) en segundo plano y analizar dónde se muestra la salida estándar y qué sucede cuando lee de la entrada estándar.
7. Arrancar el comando de prueba en segundo plano, cerrar la ventana del terminal donde se arranco y comprobar qué sucede. Ejecutar el mismo comando con la orden ***nohup*** (*man nohup*) también en segundo plano y repetir el ejercicio.

2.2. Monitorización de procesos

Actividad 2

1. La herramienta principal para ver los procesos en ejecución del sistema es ***ps***. Ejecutar la orden sin argumentos y ver su salida.
2. Estudiar la página de manual de *ps*, especialmente las opciones más comunes (ej. ***aux***) y el significado de los datos mostrados por cada proceso.
3. El consumo de la memoria virtual se puede obtener con ***free***(1). Consultar el consumo de memoria del sistema.
4. La herramienta ***top*** es muy útil porque muestra un resumen del sistema que incluye: carga (ver comando ***uptime***), estado de la memoria (***free***) y procesos (***ps***). Esta herramienta permite filtrar por usuario, enviar señales a procesos, ordenar la lista de procesos según diferentes criterios y configurar la información mostrada. Ejecutar ***top*** y estudiar su uso (*man top*; pulsar ***h*** una vez arrancada).
5. De la misma forma el comando ***vmstat*** permite recoger información sobre el rendimiento dinámico del sistema. Estudiar su uso (*man vmstat*) y el significado de la información que muestra.
6. Otra forma sencilla de localizar procesos específicos (por ejemplo para enviar una señal) es mediante los comandos ***pgrep*** y ***pidof***. Estudiar el uso de ***pgrep*** y ***pidof***, buscar por ejemplo los procesos que encajen con *gnome*.
7. El comando ***kill*** sirve para enviar señales a un proceso:

1. Consultar ***kill -l*** para ver las señales disponibles.
2. *kill* puede usar un *PID* o un *job* (***%n***). Repetir los ejercicios *fg*/*bg* usando *kill* en lugar de *Ctrl+c* y *Ctrl+z*.
3. Se puede enviar un señal a un proceso por nombre (a todos lo procesos que encajen) con ***pkill***.
4. Probar el envío de señales en top con el comando ***k*** (*kill*).

8. Finalmente el comando ***lsof*** permite ver los descriptores que tiene abierto un proceso, consultar la página de manual y probar las siguientes opciones:
   1. Procesos tienen abierto para escritura */var/log/messages*.
   2. Procesos que tienen abierto algún fichero o directorio de */etc* (***+D***).
   3. Ficheros que tiene abierta la *shell* que está usando (***-p***).
   4. NOTA: Las conexiones de red (*sockets*) se pueden ver con ***-i***.

2.3. Automatización de procesos (*cron* y *at*)

Es el sistema de automatización permite ejecutar procesos a una hora y con una determinada periodicidad (e.g. reconstruir los índices de archivos del sistema los lunes a las 2am). Normalmente el demonio responsable es ***cron***, aunque *CentOS* usa ***anacron*** para equipos que se apagan regularmente.

Actividad 3

1. Las programación de trabajos generales se especifica en ***/etc/crontab***. Cada línea especifica una variable o un trabajo, mediante 5 parámetros temporales. Un * en uno de los parámetros significa para

todos los posibles valores. Determinar la configuración para ejecutar un comando:

1. 15 minutos después de la medianoche todos los sábados.
2. El primer día de cada mes a las 3:30 AM.

2. Consultar los archivos del directorio /etc/cron.d e interpretar su contenido.
3. ***Cron*** está a disposición de los usuarios:
    1. ***crontab -e***, permite editar una nueva entrada. Planificar la ejecución de un comando (***date >> /tmp/salida***) cada 5 minutos (****/5***).
    2. comprobar la planificación de trabajos con ***crontab -l***.
4. El comando ***at*** sirve para planificar trabajos específicos que se ejecutan una sola vez, planificar la ejecución de un comando (ej. *date > /tmp/at.ejemplo*):
    1. ***at <hora>***, la hora se puede especificar de muchas formas ver (*man at*), por ejemplo *at now + 3 minutes*
    2. Escribir los comandos que se ejecutarán, terminar con ***Ctrl+d***
    3. Consultar los trabajos planificados con ***atq*** (***atrm*** elimina comandos)

3. *Logs* del sistema

Una parte importante del mantenimiento de un sistema es monitorizar las acciones que ocurren en él. Los archivos de ***log*** recogen esta

información y son gran ayuda cuando es necesario analizar un problema. El servicio de *logs* por defecto en *RHEL* y *CentOS* es ***rsyslog***.

Actividad 4

1. Comprobar que el paquete ***rsyslog*** está instalado y que está en ejecución.
2. El archivo de configuración es */etc/rsyslog.conf*. Estudiar la sección rules, cada regla hacereferencia ***<servicio>.<severidad>***. Los servicios son *authpriv*, *cron*, *kern*, *mail*, *news*, *user*, y *uucp*; y los niveles de severidad, de menor a mayor son: *debug, info, notice, warn, err, crit, alert, emerg*.
3. Estudiar los archivos de *log* en ***/var/log***, ejemplo ***boot.log, syslog. y auth.log***.
4. Los archivos de *log* pueden crecer demasiado lo que dificulta su manejo. Por defecto la utilidad ***logrotate*** rota los *logs* cada semana. Estudiar los contenidos de ***/etc/logrotate.conf*** y los archivos de configuración específicos de cada servicio en ***/etc/logrotate.d***.