UD.5.- Monitorización Operativos Libres e Propietarios	9 sesións
Obxectivos asociados ao R.A.5.	
• Coñecer e describir os diferentes programas de monitorización. • Identificar os diferentes problemas de rendemento das unidades de almacenamento. • Realizar tarefas de mantemento do software do sistema en función da observación da actividade do mesmo.	
Contidos conceptuais	
• Arranque do sistema operativo en rede. • Descrición dos fallos producidos no arranque e as posibles solucións. • Uso de ferramentas para o control e o seguimento do rendemento do sistema operativo en rede. • Xestión de discos: cotas. • Xestión dos procesos relativos aos servizos do sistema operativo en redee. • Automatización das tarefas do sistema.	

1. Arranque del sistema operativo en red

Es el proceso por el cual el gestor de arranque toma la información necesaria de un dispositivo, la carga en memoria principal y le cede el control de la ejecución.

Una vez cargado el núcleo del sistema, se procede a activar los procesos que van a definir el comportamiento de nuestro sistema. **Grub es gestor de arranque** que se ha impuesto en el mundo Linux.

La primera operación que realiza el núcleo es leer el fichero **/etc/inittab** que describe qué procesos se inician en la carga y durante la operación normal. Este fichero contiene la configuración **del proceso init** que siempre tiene el **PID 1** como primer proceso del sistema. I**nit es el padre de todos los procesos**, y su misión fundamental es crear los procesos a partir de los guiones indicados en el fichero /etc/inittab.

- **Práctica.-** Visualiza el fichero de configuración /etc/inittab

Init distingue **múltiples niveles de ejecución (runlevels)**, cada uno de los cuales puede tener su propio conjunto de procesos que se inician o se cancelan. Los niveles de ejecución válidos **son del 0 al 6**, S y s más A, B y C para entradas bajo demanda.

Los **niveles de ejecución 0, 1 y 6 están reservados**. El nivel de ejecución 0 se reserva para detener el sistema y el nivel 6 se utiliza para reiniciarlo. En nivel 1 se utiliza para llevar al sistema al modo monousuario. El nivel de ejecución S en realidad no se usa directamente, sino que toma parte de sus datos del nivel de ejecución 1.

Los niveles de ejecución restantes serían el 2 que corresponde al modo multiusuario sin parte de la configuración de red, el **3 que es un sistema mutiusuario completo**, el 4 que no se usa y el 5 que corresponde al modo gráfico.
Una entrada del fichero inittab tiene el siguiente formato:

```
id:niveles_ejecución:acción:proceso
```

Las acciones válidas para el campo acción son:

respawn: El proceso se reiniciará cuando termine (v.g. getty).

wait: El proceso se iniciará una vez cuando se entre en el nivel de ejecución especado e init esperará a su terminación.

once: El proceso se ejecutará una vez cuando se entre en el nivel de ejecución especificado.

boot: El proceso se ejecutará durante el arranque del sistema. El campo niveles_ejecución se ignora.

bootwait: El proceso se ejecutará durante el arranque del sistema, mientras init espera su terminación (v.g. /etc/rc). El campo nivel_ejecución se ignora.

off: Esto no hace nada.

ondemand: Se ejecutará cuando se llame al nivel de ejecución especificado ondemand.

initdefault: Una entrada initdefault especifica el nivel de ejecución en el cual se entrará tras el arranque del sistema. Si no existe ninguno, init pedirá un nivel de ejecución en la consola. El campo proceso se ignora.

sysinit El proceso se ejecutará durante el arranque del sistema. Se ejecutará antes de

cualquier entrada boot o bootwait. El campo niveles_ejecución se ignora.

powerwait: El proceso se ejecutará cuando init reciba la señal SIGPWR, indicando que hay algún problema con la alimentación eléctrica. Init esperará que el proceso termine.

powerfail: Como en powerwait, excepto que init no espera que el proceso se complete.

powerokwait: El proceso se ejecutará cuando init reciba la señal SIGPWR, con la condición de que haya un fichero llamado /etc/powerstatus que contenga la palabra OK. Esto significa que la alimentación eléctrica ha vuelto.

ctrlaltdel: El proceso se ejecutará cuando init reciba la señal SIGINT. Esto significa que alguien en la consola del sistema ha pulsado la combinación de teclas CTRL-ALT-DEL.

kbrequest El proceso se ejecutará cuado init reciba una señal del gestor de teclado que se ha pulsado una combinación especial de teclas en el teclado de la consola.

- **Práctica.-** Configura Ubuntu-Desktop para que NO inicialice el gestor gráfico. Para ello podemos hacerlo de dos formas:

– editar el fichero y modificarlo
– ejecutar como root la orden ***#init nivel_ejecución (1,2,3...)***

El proceso init también **lanza una serie de scripts q**ue configuran el resto del sistema; se completa la **gestión del hardware** configurándolo en **si::sysinit:/etc/rc.d/rc.sysini**t y se lanzan los servicios correspondientes cada nivel de ejecución en las líneas

```
l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6
```

En cada uno de los niveles de ejecución se ejecutan una serie de programas y verificaciones y se lanzan una serie de servicios.

Como señalamos **GRUB** es el gestor de arranque habitual en Linux. Hasta hace pocos años **LILO** era el más utilizado. Tenía la ventaja de su sencillez pero ha sido desplazado por el primero. Los gestores de arranque a veces no son necesarios sobre todo en aquellas BIOS capaces de cargar y pasar el control al núcelo de GNU/Linux si necesidad de gestor.

Modificar el gestor implica realizar cambios en el fichero **/boot/grub/grub.cfg** (en realidad este no se modifica sino a través de las modificaciones de otros), **/etc/default/grub** y los de la carpeta **/etc/grub.d.** El archivo principal **grub.cfg** se puede generar ejecutando:

 # sudo update-grub2

A continuación veremos unos ejemplos de modificación del **grub (**en internet hay muchos más)

- Práctica.- Cambiar el tiempo de espera y el sistema por defecto

Estos parámetros se modifican en el archivo **/etc/default/grub**:

$ sudo gedit /etc/default/grub

En la sección:

GRUB_DEFAULT=0

Sistema operativo por defecto, 0 es el primero, 1 el segundo, etc.

GRUB_TIMEOUT="10"

Tiempo de espera en segundos Despues de modificar el archivo, debemos actualizar grub.cfg con el comando:

$sudo update-grub2

- **Práctica: Eliminar el sistema de recuperación.**

Abrimos etc/default/grub

$sudo gedit /etc/default/grub

y cambiamos

#GRUB_DISABLE_LINUX_RECOVERY="false"

por:

GRUB_DISABLE_LINUX_RECOVERY="true"

Despues actualizamos grub2:

$sudo update-grub2

- **Práctica** Eliminar un kernel de Ubuntu

Para eliminar un kernel, podemos mover los archivos de ese kernel de la carpeta /boot a una carpeta creada por nosotros. Primero creamos la carpeta:

```
$sudo mkdir /boot/kernels
```

A continuación movemos la imagen del kernel (**debemos cambiar 2.6.31-14** por el kernel que queramos eliminar del menú):

```
$sudo mv /boot/vmlinuz-2.6.31-14-generic-pae /boot/kernels/
```

Y tambien podemos mover la imagen de que se utiliza para entrar en el modo recuperación (**debemos cambiar 2.6.31-14** por el kernel que queramos eliminar del menú):

```
$sudo mv /boot/initrd.img-2.6.31-14-generic-pae /boot/kernels/
```

Despues actualizamos grub.cfg:

```
$sudo update-grub2
```

- **Práctica** Proteger con contraseña

Para que ningún usuario no autorizado pueda modificar los valores de GRUB en tiempo de ejecución, podemos establecer una contraseña, de esta forma sólo pulsando la tecla 'p' e introduciendo la contraseña se podrán cambiar los parámetros del programa.

- Editamos el archivo de configuración del menú de arranque de GRUB:

$ sudo gedit /boot/grub/menu.lst

Buscamos la siguiente línea:

#password topsecret

- Borramos la almohadilla o numeral (#) de la línea, haciendo esto la descomentaremos.

password topsecret

- Guardamos el archivo y cerramos el editor.

Ahora la contraseña es **topsecret**, se puede cambiar e introducir la que se desee.

Hay varias **formas de recuperar** el gestor de arranque.

- Usando una **distribución LiveCD**
- Usando **SuperGrubDisk**
- Usando el **intérprete de comandos GRUB**

En el enlace se ven todas, a modo de práctica veremos la tercera.

- Práctica. Ejecutamos los siguientes comandos:

 # Ejecutamos el interprete de comandos del GRUB
$ sudo grub
 # Indicamos dónde se encuentra la partición de Ubuntu
> root (hdX,Y)
#Instalamos GRUB en ese disco
> setup (hdX)
 # Salimos del intérprete de comandos de GRUB
> quit

Donde **X** es el número de disco rígido, y **Y** es el número de partición. Este sistema difiere un poco del usado para montar las particiones en GNU/Linux; ambos son un único número decimal y comienzan en 0; por ejemplo:

- **hd0**: es el primero disco duro completo, al igual que *hda* o *sda*
- **hd0,0**: es la primera partición del primer disco duro, al igual que *hda1* o *sda1*
- **hd0,1**: es la segunda partición del primer disco duro, al igual que *hda2* o *sda2*
- **hd1,2**: es la tercera partición del segundo disco duro, al igual que *hdb3* o *sdb3*

El primer disco duro del GRUB es el primer disco duro maestro, el segundo es el primer disco duro esclavo, el tercero es el segundo disco duro maestro, y así sucesivamente.

En el arranque de **Windows XP implica los siguientes archivos:**

NTLDR --> C: (System Partition Root) --> Preinicio e Inicio (preboot y boot)

BOOT.INI --> C: --> Inicio: es un archivo de texto que utiliza el NTDLR para mostrar al inicio los sistemas operativos instalados

BOOTSECT.DOS --> C: --> Inicio (opcional) solo si uno de los sistemas es win9x

NTDETECT.COM --> C: --> Inicio detecta el hardware instalado

NTBOOTDD.SYS --> C: --> Inicio (opcional)

NTOSKRNL.EXE --> systemrootsystem32 --> Carga del Kernel (núcleo)

HAL.DLL --> systemrootsystem32 --> Carga del Kernel (núcleo) y hardware

SYSTEM --> systemrootsystem32 --> Inicialización del Kernel

dispositivos.sys --> systemroorsystem32drivers --> Inicialización del Kernel

Práctica.- busca y examina el fichero **boot.ini**

Muchas suele ser las razones, tanto software como hardware, por las cuales **windows no arranque.** En la web http://support.microsoft.com/ a modo de resumen las podemos resumir en

- Insertar el CD entrar en modo recuperación R y ejecutar en la línea de comandos los siguiente **fixmbr o fixboot.** Ambos permite la recuperación del arranque del sistema. Suelen resolver muchos de los problemas generados en dicho arranque.
- Insertar el CD y entrar en modo recuperación o restaurar la instalación (suele aparecer después de haber elegido instalar). En este caso recupera el sistema pero sin tocar los archivos de usuarios (documents and settings) y los programas instalados por este.
- En modo consola ejecutar **fdisk /mbr** esta opción se utiliza si tenemos instalado el gestor de arranque GRUB si compartimos en el mismo equipo ambos sistemas Linux y Windows. Elimina dicho gestor y arranca Windows por defecto.

2. Permisos y atributos

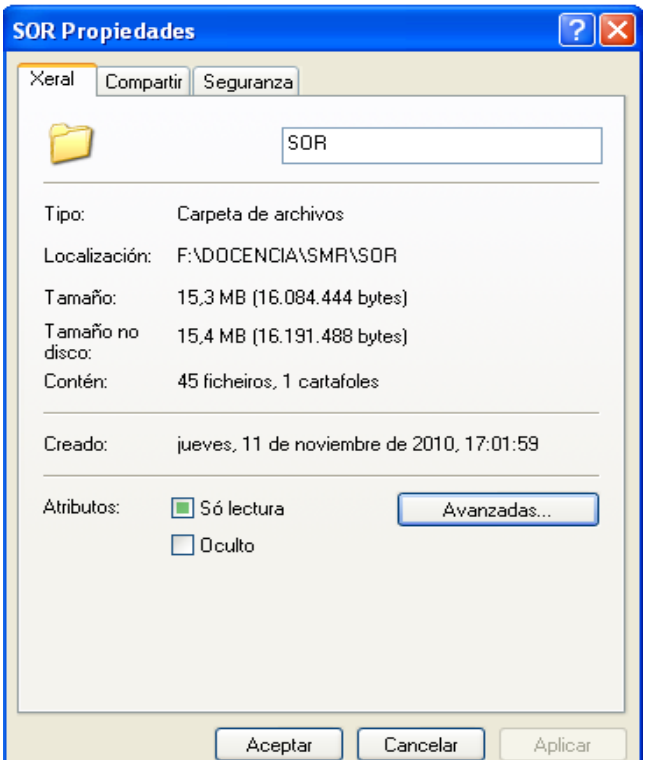
Windows utiliza cuatro bits para controlar la **lectura (read-only), sistema (system), oculto (hidden) y resto de usuarios.**

Unix-Linux en cambio utiliza el sistema UGO (usuarios, grupo y otros) para permisos sobre lectura (igual que Windows), escritura y ejecución (algo que Windows no contempló hasta 2000/XP). Para suplir esa diferencia en SAMBA se utiliza el **mapeo de bits sobre los permisos de archivos y directorios.** Al no coincidir el tipo de atributos en ambos sistemas (no existe permiso ejecución el Windows) SA;BA establece tres tipos de mapeo:

- **mapeo archivo contra los bits de permiso de ejecución del propietario**
- **mapeo system idem del grupo**
- **mapeo hiden idem del otros**

Ejemplo.

```
[data]
   path = /home/samba/data
   browseable = yes
   guest ok = yes
   writeable = yes
   map archive = yes
   map system = yes
   map hidden = yes
```



SAMB A tiene varias opciones para la creación y eliminación de máscaras que ayudan a definir los permisos que estamos viendo y sobre todo los relacionados con los recursos compartidos con Windows.

Veamoslo con ejemplos:

create mask controla los permisos de archivos. Así

create mask = 744

Establece permisos totales (rwx al propietario (owner), y de lectura al grupo y otros para un cliente Windows.

directory mask controla lso permisos de los directorio recién creado

directory create mask = 755

Establece permisos de totales a propietario, de **recorrer** y lectura al grupo y resto.

3. Herramientas de monitorización

Tanto en Windows XP/7 Pro como en Windows Server (en este caso son más elaboradas) las herramientas de configuración están en :

MiPC -> Panel de Control -> Herramientas administrativas y ahí encontramos

- **Monitor del sistema:** en el podemos observar desde el consumo de CPU, memoria, disco duro, bits de entrada-salida de eth0...
- **Registros y alertas de rendimiento.** Los "registros y alertas" aumentan las capacidades de seguimiento del monitor del sistema incluyendo funciones para almacenar información de registro y traza así como generar alertas. Las acciones que permite realizar son las siguientes:
  - **Registro de contado**r. Permite guardar en un fichero los valores de los contadores del sistema que previamente hayamos indicado.
  - **Registro de seguimiento.** Los registros de seguimiento registran los datos reunido**"Acción" se establece la acción a realizar**s por el proveedor del sistema operativo o datos de programas ajenos al proveedor del sistema operativo.
  - **Alertas.** Nos permite definir alertas a partir del valor de los contadores del sistema. Una vez establecidos los límites de los contadores en la pestaña **"Acción" se establece la acción a realizar**. La acción que puede realizar una alerta es: registrar una entrada en el registro de sucesos de aplicación, enviar un mensaje de red a un determinado equipo, iniciar un registro de seguimiento definido anteriormente, o incluso, ejecutar un programa, por ejemplo, que envíe un mensaje al móvil del administrador del sistema.
- **Visor de sucesos** El visor de sucesos permite ver y administrar los registros de sucesos, recopilar información sobre los problemas hardware y software, y supervisar los sucesos de seguridad de Windows. Los sucesos se dividen en tres categorías:

  - **Registro de aplicación.** El registro de aplicación contiene los sucesos registrados por las aplicaciones o programas. Por ejemplo, un programa de base de datos podría grabar un error de fichero en el registro de aplicación.
  - **Registro del sistema.** El registro del sistema contiene los sucesos registrados por los componentes de Windows. Por ejemplo, el error de la carga de un controlador u otro componente del sistema durante el inicio se graba en el registro del sistema. Los tipos de sucesos registrados por los componentes del sistema están predeterminados.
  - **Registro de seguridad.** El registro de seguridad puede grabar sucesos de seguridad, como los intentos de inicio de sesión válidos y no válidos, y los sucesos relativos al uso de recursos, como crear, abrir o eliminar ficheros. Por ejemplo, si ha habilitado la auditoria de inicios de sesión, los intentos de inicio de sesión en el sistema se graban en el registro de seguridad.

Práctica: establecer algún tipo de Alerta por ejemplo de consumo de la memoria por encima del 75% o de CPU por encima del 90%. También establecer un registro o auditoría que controle el número de inicios de sesión por parte de usuario o bien el acceso a un determinado fichero.

En Linux las herramientas para monitorización son numerosas y dependen del la elección del administrador. En primer lugar tenemos las de consolas propias del sistema y que son:

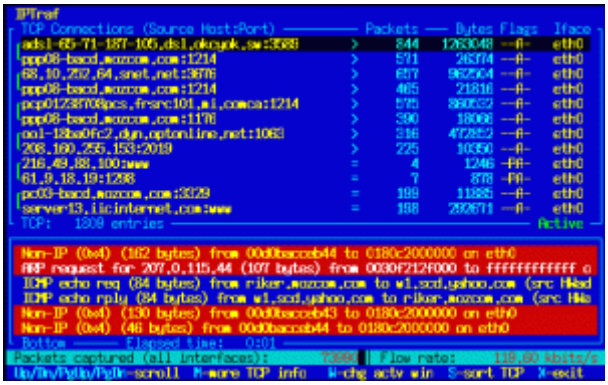
- **top:** en una pantalla nos muestra todos los proceso con su pid que se están ejecutando, el propietario del mismo, consumo de memoria y cpu. Una vez ejecutándose puedes introducir más opciones en una lína de comando por encima de la columna de presentación.
- **df:** nos muestra el grado de utilización del sistema de ficheros.
- **free:** grado de utilización de la memoria física.

La mayoría del tiempo de un **sysadmin (administrador de sistemas)** se la pasa monitoreando. Hay herramientas que pueden facilitar el trabajo.

1. Monitoreo de ancho de banda

Esto es muy importante, ya que al monitorear cuanto **ancho de banda** consume un servidor en tiempo real podemos determinar cuales son sus horas pico y así optimizarlo. Para ello existe el programa **iptraf** que monitorea la interface de red.

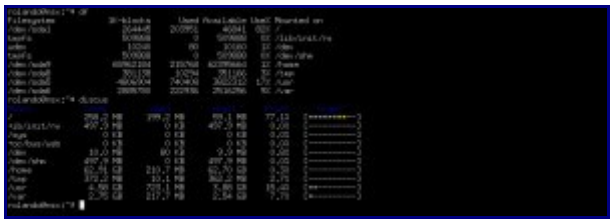
2. Monitoreo de consumo de CPU y Memoria



Otra de las tareas de un sysadmin es monitorear el consumo de CPU y Memoria (RAM y SWAP), linux viene con el comando “top”. Existe también **htop,** muy completo ya que nos detalla el consumo de CPU y Memoria por proceso, así como el consumo general de los recursos del sistema. Es mucho mas amigable a comparación del top normal. Conforme se va usando el procesador o los núcleos del procesador el programa lo indica de una manera “gráfica”.

3. Monitoreo de consumo de disco duro

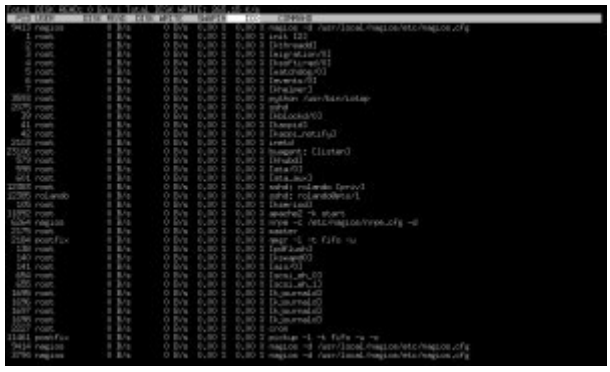
Cada cierto tiempo es necesario que entremos a revisar cuando espacio en disco hay libre en los servidores que manejamos. Para saber el espacio en disco duro se ejecuta el comando “df” en linux. Más “amigable” es **discus**. Este programa es que te indica cuanto espacio esta usado y cuando espacio hay libre, así como los porcentajes de los mismos.



4. Monitoreo del I/O

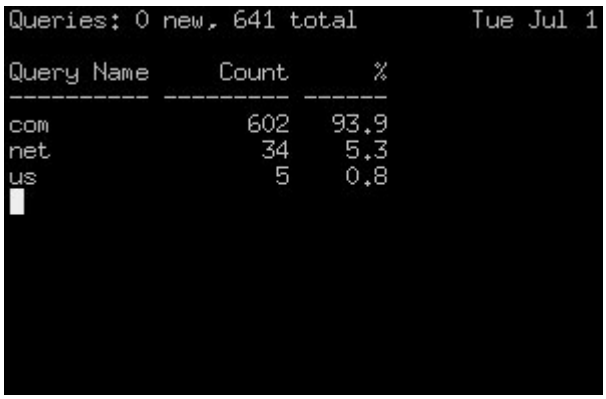
En computación, entrada/salida, abreviado E/S o I/O es la colección de interfaces que

usan las distintas unidades funcionales (subsistemas) de un sistema de procesamiento de información para comunicarse unas con otras, o las señales (información) enviadas a través de esas interfaces. Para monitoriza se usa la herramienta **iotop** que permite el proceso read/write de un disco, p.e.



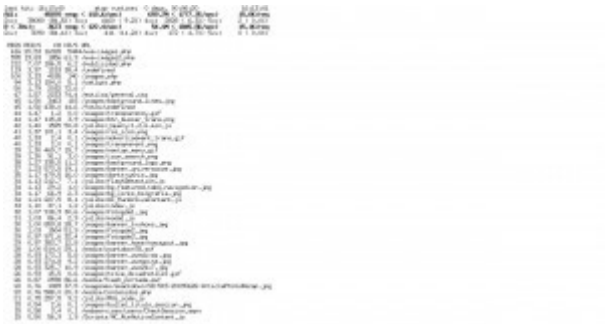
5. **Monitoreo de peticiones DNS**

Para monitorear esto utilizo el programa **dnstop** con este se puede monitorear cuantas peticiones esta teniendo tu servidor ya sea por clase de dominio .net .org, direcciones ip, por dominio o por sub dominios. Se ejecuta con la orden ***dns eth0*** especificando la NIC, y presionando los botones de 1 2 3 4 obtienes diferentes informes.



6. **Monitoreo de tráfico web**

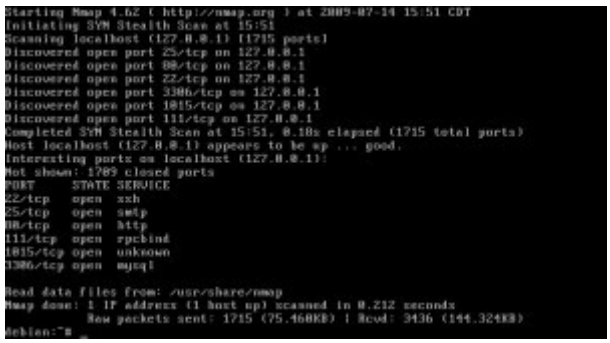
En servidores de tráfico web es importante monitorizar en tiempo real para ver que archivos estamos sirviendo, el tráfico en megas. La herramienta es **apachetop.** Entre la informacióne está las páginas que servimos, peticiones por seg, promedio general...



7. **Monitoreo de puertos abiertos**

Hay una regla principal en los servidores: **si un servicio no te sirve, quítalo del sistema.** La herramientas para monitorizar es **nmap**

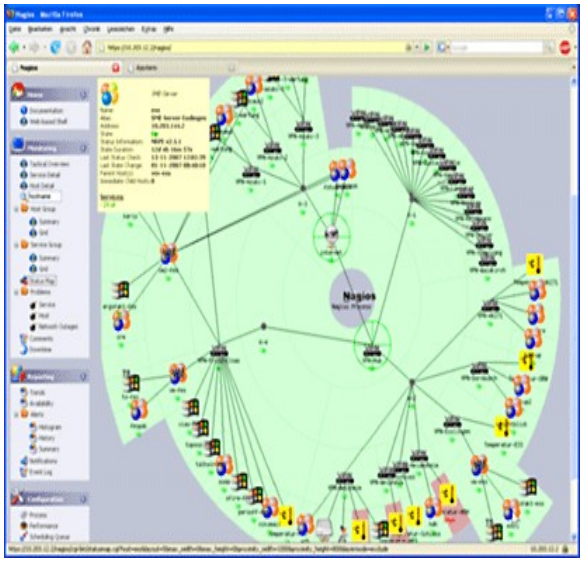
Su uso es **nmap -v 127.0.0.1** **(localhost)**



También está **ntop** para monitorizar la red.

Aparte de estos existe otros como:

- **cacti:** también para red, disco, sensores de voltaje y temperatura. Se usa con un front end vía web. Permite monitorizar varios servidores al mismo tiempo.
- **Nagios:** uno de los más potentes del mercado sino el que más dentro del campo de opensource. Comparable a cualquier otro de pago. Usado en ambientes profesionales (El Sergas lo utiliza para monitorizar sus servidores y la red). Tiene un front-end vía web. Compleja instalación.



Virtual hosts (virt-hostgroup)

Host	Status	Services	Actions
centos5qax64fv	UP	1 OK	
debian32fv	UP	1 OK	
f764pv	UP	1 OK	
fc6_0	UP	1 CRITICAL	
fc6_1	UP	1 CRITICAL	
freebsd32fv	UP	1 CRITICAL	
gentoo32fv	UP	1 CRITICAL	
rhel5qax32fv	UP	1 OK	

- **Webmin:** también tiene varias herramientas para monitorizar el servidor.

Otros como **zabbix, logwatch, bixdesktop, loadavg, pandora, hyperic** son también muy potentes pero de instalación algo compleja. Hay más pero los más utilizados son estos.

Practica.- Intenta instalar el servidor Nagios. Busca en internet su manual de instalación. Podeís trabajar en grupos.

Gestión de quotas

- **En XP**

Para ver la ayuda de Cuotas de discos ve a la carpeta **\WINDOWS\Help** y doble clic sobre el archivo dskquoui.chm

Esto consiste en asignar a un usuario o a un grupo una cierta cantidad de disco duro. Es decir, si **por ejemplo** el disco (o partición) es R y tiene 8 G, podemos hacer que el usuario Juan sólo pueda disponer de 1 G. Se utiliza sólo con NTFS.

Para asignarlo vamos a **Mi PC/disco R:/Propiedades** debe salir la ficha Cuota.

Podemos poner un valor de cuota de disco mediante : Limitar espacio de disco por ejemplo : 100 MB.

Podemos poner una advertencia cuando se acerque a esa cantidad : Establecer nivel de advertencia Ejemplo : 90 MB

Podemos **particularizar la cuota de disco a cada usuario mediante el botó**n: Valores de cuotas / Cuota / Nueva entrada de cuota ... / Ponemos o buscamos un usuario y le asignamos la cantidad de disco R: que puede utilizar.

Para habilitar la utilización de cuotas de disco vamos a :

Inicio/Ejecutar/gpedit.msc

Configuración de equipos/Plantillas administrativas/Sistema/Cuotas de disco/Habilitar cuotas de disco.

Práctica: De los usuarios que tengas en XP habilitale 40 Megas con aviso de 30 y bájate un fichero mayor para observar que sobrepasa el Ímte.

- En **Linux**
  - **Tipos de cuota**

    Por Bloques (blocks): Un bloque corresponde a 1 kb y una cuota por bloques correspondería al total de bloques que un usuario puede utilizar en el sistema. Recuerda que los archivos se guardan en bloques de dlisco. Asi un archivo de 100 bytes, ocupará un un bloque de 1kb en el disco duro.

    Por Inodos (inodes): Un inodo o inode en inglés (Index Node) es un número que actua como apuntador para el sistema de archivos de Linux y le indica en que bloques específicos del disco duro se encuentran los datos de un archivo.

  - **Límites**

    Tanto las cuotas por bloques o por inodos, tienen límites de uso y son de dos tipos:

    HARD: (Duro) Cuando se establece (para bloques o inodos), es el límite absoluto. El usuario no podrá exceder este límite.

    SOFT: (Suave) Este límite (para bloques o inodos) que es siempre menor al HARD, puede ser excedido por el usuario, pero será **constantemente advertido** que el límite de uso para bloques o inodos ya ha sido excedido. Si se ha establecido un tiempo de gracia (días, horas...) podrá usar bloques o inodos hasta alcanzarlo.

Para la **implementación se debe decidir** donde se instala el sistema de quotas, la mayoría de las veces es en el home de los usuarios, o bien en el caso de servidores web o ftp los directorios correspondientes /var/www...

Como **root editamos /etc/fstab** y añadimos **usrquota o grpquota o ambas.**

Ejemplo:

/dev/sda3 /home ext3 noatime,usrquota,grpquota 1 2

El siguiente paso es ejecutar **quotacheck** para que se **cree, verifique o repare el control de quotas.**

Ejemplo:
#**quota -augmv (nota u y g es usuario y grupo, m significa montar el sistema, a verificar)**

En /home si ahora ejecutamos **ls -al** observaremos dos ficheros **aquota.user y aquota.group. (están en binario no intentes acceder a ellos9**

Los comandos **quotaon y quotaoff** activan y desactivan el soporte de quotas.

Para activar la cuota a un usuario

#> edquota -u *user1*
Disk quotas for user user1 (uid 502):

Filesystem	blocks	soft	hard	inodes	soft	hard
/dev/sda3	56	0	0	14	0	0

Si quiesesemos a nivel de grupo, por ejemplo sería, **edquota -g ventas**

Para observar el uso de las quotas como root:

#> **quota -u user1**
Disk quotas for user user1 (uid 502):

Filesystem	blocks	quota	limit	grace	files	quota	limit	grace
/dev/sda3	56	70	100		14	0	0	

Un informe global

#> repquota /home
*** Report for user quotas on device /dev/sda3
Block grace time: 7days; Inode grace time: 7days

	Block limits				File limits			
User	used	soft	hard	grace	used	soft	hard	grace
root --	184280	0	0		11	0	0	
sergio --	42579852	0	50000000		34902	0	0	
user1 --	56	70	100		14	0	0	
user2 --	52	0	0		13	0	0	
user3 --	28	0	0		7	0	0	

Para establecer **tiempo de gracia**

#> edquota -t
Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
Filesystem	Block grace period	Inode grace period
/dev/sda3	7days	7days

Para fija **quotas de manera global a todos los usuarios,** es muy cómodo cuando administramos servidores con cientos de usuarios. Para ello realizaremos el siguiente **script,**

A. En primer lugar establecemos una quota para un usuario

#> edquota -u user1
Disk quotas for user user1 (uid 502):
Filesystem	blocks	soft	hard	inodes	soft	hard
/dev/sda3	68	300	400	17	0	0

:wq

B. A continuación usamos la opción -p para hacer duplicados.

#>edquota -p user1 user2 user3 .....

Aún así es algo lento. Podemos usar el comando **gawk**

#> gawk -F: '$3 > 499 {print $1}' /etc/passwd
user1
user2
user3
user4

Lo que hemos hecho es crear un pqueño script que extrae el nombre de todos lo usuarios del fichero /etc/passwd que se corresponde con la fila 1. Luego

#> edquota -p user1 `gawk -F: '$3 > 499 {print $1}' /etc/passwd`

¡¡ojo con el tipo de coma!!

- **Aviso de quotas excedidas**

Aparecen warnings del siguiente tipo:

warning, user block quota exceeded.

Si deseamos avisar a los usuarios via mail, se usa el comando **warnquota.**

También podemos añadir una línea en el **crontab para que haga su trabajo cada 12 horas o menos o más.**

#> vi /etc/crontab

0 0,12 * * * root /usr/sbin/warnquota

Automatización de tareas

La automatización de tareas permite al administrador de sistemas de tediosos trabajos de mantenimiento del sistema (actualizaciones, copias de seguridad, paradas del sistema programados, borrado de ficheros temporales /tmp...)

Uno de los elementos importantes es el **demonio cron** /usr/sbin/cron que ejecuta tareas programadas con precisión de 1 minuto. Para ello lee en el fichero **crontab** los trabajos que tiene que realizar. Los parámetros fundamentales son:

- **crontab [-u usuario] fichero**
- **crontab [-u usuario] { -l | -r | -e }**

"-u usuario": disponible solo para root, permite ver o modificar las tareas programadas para otro usuario.

"fichero": Ejecuta las tareas programadas que aparezcan en el fichero indicado.

"-l": Muestra la lista de todas las tareas programadas.

"-r": Elimina todas las tareas programadas.

"-e": Edita la lista de tareas programadas.

Los ficheros **/etc/cron.allow y /etc/cron.deny** indican los usuarios que tienen o no permiso para utilizar crontab.

En el crontab cada línea contiene 6 campos separados por espacios:

- minuto (0–59)
- hora (0–23)
- día del mes (1–31)
- mes (1–12)
- día de la semana (0–7). 0=7=domingo, 1=lunes, 2=martes. . .
- orden a ejecutar

Un * indica toso. Un numero indica momento exacto. Números separados por comas indica lista y separados por guión un rango.

0 9,18 * * 1-5 echo "hora de comer" | wall

Ejecuta la orden de lunes a viernes a las 9:00 y a las 18:00

0,30 * 13 * 5 echo “Hola” | wall

Envía el mensaje cada media hora, todos los viernes y además los días 13 de cada mes.

El demonio crontab examina, cada minuto, los ficheros crontab de los usuarios para ver las tareas a realizar

Hay unos **momentos especiales:**

@yearly: Se ejecuta una vez al a˜no.
@monthly: Se ejecuta una vez al mes.
@weekly: Se ejecuta una vez por semana.
@daily: Se ejecuta una vez al d´ıa.
@hourly: Se ejecuta una vez por hora.
@reboot: Se ejecuta al iniciarse la máquina

La orden crontab de cada usuario se guarda en /var/spool/cron/crontabs/usuario.

Ejemplo:

0 8 * * * root /etc/init.d/gdm restart
A las 8 de la mañana de todos los días, como usuario root, se reincia el servicio gdm.

El fichero crontab por defecto tiene la siguiente estructura:

```
17 * * * * root run-parts --report /etc/cron.hourly
25 6 * * * root run-parts --report /etc/cron.daily
47 6 * * 7 root run-parts --report /etc/cron.weekly
52 6 1 * * root run-parts --report /etc/cron.monthly
```

Ejecuta con la periodicidad indicada los scripts que se encuentren en esos directorios.

Las tareas periódicas indicadas en los directorios /etc/cron.daily, /etc/cron.weekly... suelen ser tareas rutinarias del sistema que conviene realizar de vez en cuando, como limpiar los temporales, gestionar colas de correo, realizar back-ups, etc.
Estas tareas se realizan en horas de poco uso de la máquina (de madrugada).
Pero es habitual que haya máquinas que no estén encendidas siempre. Para estas existe el demino **anacron, s**ervicio que se inicia al encender la máquina, y que realiza las tareas pendientes.

Los demonios **at y atd** ejecutan tareas concretas en un istante dado. Los ficheros de configuración son similares a los de cron.
Ejemplo
at fecha/hora HH[:]MM [am|pm] [mes día] tarea

at now midnight noon teatime today tomorrow tarea

at fecha/hora + número (minutes, hours, days, weeks) tarea

Práctica: Realiza la programación de alguna tarea, por ejemplo, limpiar el fichero /tmp, crear una copia de los ficheros de usuarios...

En XP

Para realizar tareas en Windows se usa el programa **Tareas Programadas**

Inicio -> Programas -> Accesorios -> Herram. Administratvas -> Tareas programadas

Las tareas en XP se pueden programar diariamente, semanalmente, mensualmente, solo una vez, al inciar la sesión o el equipo. Evidentemente también se especifica la hora.
Si quiero que se repita el mismo día en **Avanzadas** se indica **Repetir tarea.**

Practica.- Establece una tarea que consista en lanzar el WordPad cuando el usuario Adan incie su sesión.