Comandos de Consola

Administración de archivos

ls
Descripción: = list. listar contenido de directorios.
Ejemplos: ls, ls -l, ls -fl, ls --color

cp
Descripción: =copy. copiar ficheros/directorios.
Ejemplos:cp -rfp directorio /tmp, cp archivo archivo_nuevo

rm
Descripción: =remove. borrar ficheros/directorios.
Ejemplos: rm -f fichero, rm -rf directorio, rm -i fichero

mkdir
Descripción: =make dir. crear directorios.
Ejemplos: mkdir directorio

rmdir
Descripción: =remove dir. borrar directorios, deben estar vacíos.
Ejemplos: rmdir directorio

mv
Descripción: =move. renombrar o mover ficheros/directorios.
Ejemplos: mv directorio directorio, mv fichero nuevo_nombre, mv fichero a_directorio

chmod
Descripción: cambia los permisos de lectura/escritura/ejecución de ficheros/directorios.
Ejemplos: chmod +r fichero, chmod +w directorio, chmod +rw directorio -R, chmod -r fichero

chown
Descripción: =change owner. cambia los permisos de usuario:grupo de ficheros/directorios.
Ejemplos: chown root:root fichero, chown pello:usuarios directorio -R

ln
Descripción: =link. para crear enlaces, accesos directos.
Ejemplos: ln -s /directorio enlace

file
Descripción: nos dice de que tipo es un fichero.
Ejemplos: file fichero, file *

find

Descripción: Permite encontrar un archivo con el nombre especificado, a partir del directorio especificado.

Ejemplos find / -name nombrearchivo.

Comandos Para manejo de dispositivos

Montar usb, cd, floppy, particiones windows fat32 y ntfs

* **mount:** Permite que particiones del sistemas, CD-ROMs, floppys puedan ser leídas en el sistema. Su formato: mount -t <file system(ext2,vfat)> <partición (/dev/hda1,/dev/cdrom)> <punto de lectura "mount point" (/mnt/home/ /mnt/cdrom)>.

* **umount:** Desactiva la partición que se indicada, los parámetros que toma este comando son similares a los de mount .

Comandos Generales

* **dmesg:** Imprime los mensajes desplegados por el "kernel" al inicio.
* **depmod -a:** Genera un archivo que contiene las dependencias de los módulos que son cargados para el "Kernel",esto es, es capaz de reconocer cuales módulos deben ser cargados para que un tercero sea utilizado en el sistema.
* **free:** Estadísticas de uso de Memoria.
* **init q:** Comando que vuelve a leer los parámetros que se encuentran en inittab .
* **insmod:** Habilita ("loads") el modulo que se especifica en la línea, para que el "kernel" sea capaz de utilizarlo.(ejemplo: insmod ip_alias.o)
* **ldconfig:** Actualiza las librerías utilizadas por el sistema, recomendable ejecutarlo cada vez que se instale un programa.
* **lsmod:** Despliega la información referente a los módulos que están habilitados por el "kernel".
* **slocate:** Actualiza la base de información que es utilizada para encontrar archivos con el comando locate .
* **stat:** Despliega información detallada sobre el archivo especificado como: fechas de modificación y cambio, dueño del archivo, etc.
* **uname -a:** Información completa sobre el "Host".
* **uptime:** Hora actual, tiempo que lleva el sistema corriendo desde el ultimo "reboot", usuarios conectados al servidor, carga del sistema en los últimos 1,5 y 15 minutos.
* **chkconfig:** Este comando despliega la información sobre los niveles de ejecución de los "scripts" ubicados en el directorio /etc/rc.d/init.d

Código:

chkconfig --list httpd Este comando despliega:
httpd 0 ff 1 ff 2 ff 3 n 4 n 5 n 6 ff

Lo anterior indica que cuando se utilice el nivel de arranque 3, el "script" httpd en el directorio /etc/rc.d/init.d recibirá el argumento "start", cuando se corra el nivel de arranque 6, httpd recibirá el argumento "stop", etc...

Para modificar hacia argumento "start":

Código:

chkconfig --add <nombre del script en directorio init.d>
--level <nivel de arranque[0 a 6]>

Para modificar hacia el argumento "stop":

Código:

chkconfig --del <nombre del script en directorio init.d>
--level <nivel de arranque[0 a 6] >

* Es precisamente de los directorios /etc/rc.d/rc [0-6] de donde proviene la información que despliega **chkconfig**.

* **ntsysv:** Es una herramienta gráfica que tiene la misma funcionalidad que **chkconfig**, la diferencia es que esta herramienta despliega todos los "scripts" por nivel, esto es, si se utiliza el comando ntsysv --level 3 , la gráfica mostrará el status "stop" o "start" de todos los "scripts" para el nivel de arranque 3 . De la misma forma se utilizan: ntsysv --level 5 , ntsysv --level 0 ,etc.

Al igual que **chkconfig** , **ntsysv** modifica y toma la información que se encuentra en los directorios /etc/rc.d/rc [0-6]

Comandos Entorno De Red

En Entorno de Red

***hostname:** El nombre del "Host".

* **host:** Determina la dirección IP de un "Host" , host -a despliega toda la información de DNS.

* **ifconfig:** Permite configurar una interfaz de Red y ver el "status" de ésta. Es de la forma ifconfig <interfaz>, ejemplo: ifconfig eth0

* **ifup:** Habilita la interfaz especificada, ejemplo: ifup eth0 .

* **ifdown:** Deshabilita la interfase especificada, ejemplo: ifdown eth0 .

* **netstat -a:** Todas la conexiones de Red originadas y recibidas por el "Host"

* **netstat -r:** Muestra la tabla de ruteo "routing table" del sistema

* **netstat -i:** Estadísticas de red de cada interfase

* **nslookup:** Busca información en los servidores DNS, ejemplo: nslookup -query=mx osomosis.com , si no se especifican parámetros se entra en modo interactivo

* **ping -s 1016:** Manda paquetes de ping de 1024 bytes (header 8 bytes), mientras que el "default" es 512.

* **route add:** Permite agregar tablas de ruteo de y hacia el "Host".
Ejemplo: Para guiar toda la información de la red 206.171.55.16 netmask 255.255.255.240 vía la interfaz eth0:

Código:

route add -net 206.171.55.16 255.255.255.240 eth0
Para rutear todo el tráfico por cierta interfaz ("Default Gateway"):

Código:

route add default gw 206.171.55.51 eth0

Esto enviará toda la información por la dirección 206.171.55.51

route -n: Despliega la tabla de ruteo del "Host". NOTA: Debe de estar "IP Forwarding" ON en /etc/sysconfig/network, además el "kernel" debe de estar configurado para "IP Forwarding".

smbclient: Funciona como un cliente FTP, que simula conexiones que serán realizadas a través de Samba.

tcpdump: Permite el "debugging" de una interfase en el host.

testparm: Verifica la validez del archivo smb.conf utilizado por Samba.

Comandos Para Control De Procesos

Control de Procesos:

ps -aux: Despliega todos los procesos del sistema, con nombre y tiempo de inicio.
kill: Es utilizado para mandar señales a los procesos en Unix.
* **kill -HUP <pid>:** Señala al proceso con numero <pid>, que vuelva a leer sus archivos de configuración.
* **kill -INT <pid>:** Señala al proceso con numero <pid>, que será interrumpido .
* **kill -TERM <pid>:** Señala al proceso con numero <pid>,que debe de terminar, a diferencia de -KILL , esta opción da la oportunidad al proceso de terminar.
* **kill -STOP <pid>:** Señala al proceso con numero <pid>, que pare

momentáneamente.
* **kill -CONT <pid>:** Señala al proceso con numero <pid>, que continué, este comando se utiliza para reanudar un proceso que le fue aplicado -STOP.
* **kill -KILL <pid>:** Señala al proceso con numero <pid>,que termine de inmediato, el proceso es terminado abruptamente.
# **killall:** A diferencia de kill , killall permite mandar un señal al proceso por nombre.
* **killall <nombre del proceso >:** Envía la señal -TERM al proceso con el nombre especificado. NOTA: Por "default" la señal que toma kill y killall es -TERM.
# **ps -l:** Este comando despliega dos parámetros PRI y NI. El parámetro PRI indica la prioridad actual del proceso, que es calculada por el sistema operativo, el valor de NI es tomado en cuenta cuando se determina el PRI.

* **Que es NI? :** NI es llamado el número gentil o "nice number", este número es especificado por el "superusuario"("root") o dueño del proceso y afecta el orden final del PRI, le da prioridad a los menos gentiles. Sus valores oscilan desde -20 (menos gentil = más prioridad) y 20 (más gentil = menos prioridad)

nice: Este comando especifica el número NI de cada proceso.

* **nice -10 named:** Esto bajaría la prioridad de named en 10 unidades.(Si estaba en -10, pasará a -20).

* **nice +10 named:** Esto incrementaría la prioridad de named en 10 unidades.(Si estaba en 0, pasaría a +10).

snice y renice: El mismo funcionamiento que nice, excepto que utiliza el número de proceso :

* ***snice -10 <pid>***
# **<comando> &:** El & es utilizado para indicar que el proceso debe de ejecutarse en el fondo.

top: Esta herramienta muestra información sobre varios recursos del sistema y tiene un carácter dinámico, muestra uso de CPU por proceso, cantidad de memoria, tiempo desde su inicio, etc.

vmstat: Es muy similar a top ya que es un condensado de los procesos del sistema, para que esta herramienta se vuelva dinámica se deben especificar los argumentos: vmstat -n <número de segundos por actualización >

at: Este comando permite programar ciertas actividades a una cierta hora, ejemplo: at 22:00 , el comando anterior abre un "prompt" de la forma at> , sobre este "prompt" se especifican todos los comandos que se deseen ejecutar, en este caso a las 22:00, una vez especificados, se utiliza Ctlrl-d para salir.

Ya finalizado, los comandos estarán programados para ejecutarse a la hora indicada, el directorio /var/spool/at contiene el trabajo.

El comando atq despliega los trabajos at que están pendientes, y el comando atrm <número de at> elimina un trabajo programado con at . Vea también /etc/at.deny y /etc/at.allow

crontab: Al igual que at especifica el tiempo al cual se ejecutará un programa "script", crontab tiene la siguiente forma: minutos horas días meses fin_de_semana nombre_de_usuario instrucción argumentos

El siguiente ejemplo ejecutará el programa oracle.pl cada media hora todos los días:

Código:

```
30 * * * * root /usr/oracle.pl
```

Si se desea realizarlo mensualmente:

Código:

```
01 3 1 * * root /usr/oracle.pl
```

Lo anterior ejecutará **oracle.pl** el día primero de cada mes, a las 3:01 AM.

Para especificar trabajos cron, cada usuario mantiene un archivo en el directorio /var/spool/cron/ , que puede ser accesade por cada usuario con el comando crontab -e

La ejecución de crontab se facilita debido al archivo /etc/crontab que especifica trabajos crontab por hora, día, semana y mes, de esta forma solo se requiere que el usuario coloque un archivo en los directorios correspondientes: **/etc/cron.hourly | /etc/cron.daily | /etc/cron.weekly | /etc/cron.monthly**

Comandos Para Registros y Sistema

Control de Registros "Logs"

* **tail:** Permite ver el final de un archivo, este comando es útil ya que los archivos de registros "logs" crecen constantemente tail --f /var/log/messages

También se puede especificar el número de renglones que se deben observar:

Código:

tail --f --line 15 /var/log/messages

Este comando anterior despliega las ultimas 15 líneas del archivo ("default" = 10). La --f mantiene el archivo abierto para poder observarlo conforme se agreguen eventos.

Configuración de Sistema
/usr/sbin/sndconfig: Ejecutable utilizado para configurar el sonido del sistema.
/bin/netconf: Ejecutable utilizado para configuración de Interfases de Red.

Comandos De Administración

Comandos de administración

sysctl
Descripción: Configurar los parámetros del kernel en tiempo de ejecución.
Ejemplos: sysctl -a

ulimit
Descripción: muestra los limites del sistema (máximo de ficheros abiertos, etc...)
Ejemplos: ulimit

adduser
Descripción: añadir usuario de sistema.
Ejemplos: adduser pepe, adduser -s /bin/false pepe

userdel
Descripción: = eliminar usuario de sistema
Ejemplos: userdel pepe

usermod
Descripción: = modificar usuario de sistema
Ejemplos: usermod -s /bin/bash pepe

df
Descripción: = disk free. Espacio en disco disponible. Muy util.
Ejemplos: df, df -h

uname
Descripción: = unix name. Información sobre el tipo de Unix en el que estamos, kernel, etc.
Ejemplos: uname, uname -a

netstat
Descripción: la información sobre las conexiones de red activas.
Ejemplos: netstat, netstat -ln, netstat -l, netstat -a

ps
Descripción: = proccess toda la información sobre procesos en ejecución.
Ejemplos: ps, ps -axf, ps -A, ps -auxf

free
Descripción: muestra el estado de la memoria RAM y el SWAP.
Ejemplos: free

ping
Descripción: herramienta de red para comprobar entre otras cosas si llegamos a un host remoto.
Ejemplos: ping www.rediris.es

traceroute
Descripción: herramienta de red que nos muestra el camino que se necesita para llegar a otra maquina.
Ejemplos: traceroute www.rediris.es

du
Descripción: =disk use. Uso de disco. Muestra el espacio que esta ocupado en disco.
Ejemplos: du *, du -sH /*, du -sH /etc

ifconfig
Descripción: =interface config. Configuración de interfaces de red, modems, etc.
Ejemplos: ifconfig, ifconfig eth0 ip netmask 255.255.255.0

route
Descripción: gestiona las rutas a otras redes.
Ejemplos: route, route -n

iptraf
Descripción: muestra en una aplicación de consola TODO el tráfico de red IP, UDP, ICMP.
Permite utilizar filtros, y es SUMAMENTE UTIL para diagnostico y depuración de firewalls
Ejemplos: iptraf

tcpdump
Descripción: vuelca el contenido del trafico de red.
Ejemplos: tcpdump, tcpdump -u

lsof
Descripción: muestra los ficheros (librerías, conexiones) que utiliza cada proceso
Ejemplos: lsof, lsof -i, lsof | grep fichero

lsmod
Descripción: Muestra los módulos de kernel que están cargados.
Ejemplos: lsmod

modprobe
Descripción: Trata de instalar un modulo, si lo encuentra lo instala pero de forma temporal.
Ejemplos: modprobe ip_tables, modprobe eepro100

rmmod
Descripción: Elimina módulos del kernel que están cargados
Ejemplos: rmmod <nombre de modulo>

sniffit
Descripción: Sniffer o husmeador de todo el trafico de red. No suele venir instalado por defecto.
Ejemplos: sniffit -i

Tuberías

En el mundo Unix una tubería es una forma de comunicar dos programas. Con ellas se consigue conectar la salida estándar de un programa con la entrada estándar de otro. Veamos algunos ejemplos:

- Listado de todos los dispositivos del sistema: ls /dev/* | less
- Listado ordenado de todos los dispositivos del sistema: ls /dev/* | sort | less
- Cambia todas las letras "a" de un fichero por "b": cat hola.txt | tr a b
- Imprime un fichero de texto: cat hola.txt | lpr

Redirecciones

Permiten modificar el comportamiento de algunos programas de forma que no generen o reciban información por la vía habitual sino por otra. Esto, que así explicado parece algo difícil, es muy fácil de comprender con algunos ejemplos:

- Almacena el listado de un directorio en un fichero: ls > listado.txt
  En este ejemplo lo que se hace es redirigir la salida estándar de ls, que es el terminal, a un fichero.
- Añade un fichero nuevo al final de otro: cat 2.dat >> 1.dat

Comodines

Podemos utilizar comodines para sustituir parte del nombre de un fichero o un grupo de ellos. Esta es una forma cómoda de referirnos a un conjunto de ficheros o directorios.

comodín	**sustitución**
*	cualquier cadena
?	cualquier carácter
[]	una de los caracteres entre corchetes

Veamos algunos ejemplos:

- Como mover un grupo de ficheros a un directorio: mv *.mp3 música
- Como agrupar varios ficheros: cat *.dat > todo.dat
- Listado de todos los ficheros y directorios que comiencen por a, b o c: ls [abc]*

Encadenamiento de órdenes

Varias órdenes pueden ser encadenadas de diversas formas. Algo que debemos saber para poder aplicar esta posibilidad es que cada vez que una orden finaliza con éxito esta devuelve un valor 0 al sistema operativo. En caso de error se devuelve un valor distinto de 0 que podría utilizarse para descubrir que ha pasado. Veamos a continuación que posibilidades existen:

sufijo	**acción**	**ejemplo**
;	ejecuta órdenes de forma secuencial independientemente del resultado de cada una de ellas	ls; ps
&&	ejecuta órdenes de forma secuencial mientras el resultado de cada una de ellas sea 1	sort 1.dat && echo "ok"
||	ejecuta órdenes de forma secuencial mientras el resultado de cada una de ellas sea 0	sort 1.dat || echo "error"