Introducción al Linux

Primer Taller CEDIA

29 de Febrero, 2004

Presentado por Hervey Allen
Network Startup Resource Center

Bienvenidos

- Introduccion
  - Instructores:
    Hervey Allen
    Brian Candler
    Carlos Vicente
  - Ayudantes:
    Albert Espinal
    Federico Domínguez
    Dolores Lizarzaburu
    Neil Nuñez
- ¿A que nivel estamos?
- Como corre el clase
- Que harémos hoy?

Compendio

- Hablar sobre Instalar Red Hat Server 9 (Kickstart).
- Comandos de Sistema (cp, ls, cd, rm, etc.).
- Uso del editor basico, como vi.
- Crear cuentas de usuarios, remover cuentas de usuarios.
- Discurso /etc/passwd, /etc/group, /etc/shadow.
- Comandos, programas, shell y paths.
- Uso del comando 'su' por 'root', y /etc/sudoers.
- Bajar paquetes de RPM usando FTP y instalarlos.
- Aprender como apagar y reiniciar el servidor. Discurso de los niveles de inicializacion.
- Discurso de /etc/ y /etc/rc.d/init.d/
- Practicar de que hemos aprendido si hay tiempo.

Instalando Red Hat 9

- Como se puede instalar?
  - Usando CD/DVD.
  - Floppy y un servidor de ftp/tftp.
  - Kickstart.
  - NIC PXE boot.
- Revisamos las opciones de instalacion usando la herramienta grafíca para Kickstart.
- Lunes en la noche instalarémos Red Hat 9

Comandos basicos

- cp, cd*, ls, mv, rm y man
  - (*un parte de interpreter de comandos , “shell”).
- ¿Donde “viven” los camandos?
- /bin, /usr/bin, /usr/local/bin, /sbin, /usr/sbin
  - La diferencia entre “sbin”, “bin” y “ /usr”
- Si conoce DOS:
  - cp = copy
  - cd/chdir = cd/chdir
  - ls = dir
  - mv = move (antes era copy y delete/erase
  - rm = del[ete] y/o erase

Comandos basicos continuado

No es un comando pero vamos a practicar de inicializar un proceso separado del terminal.

Para hacer esto se uso el simbolo “&” *despues* que el comando que va a abrir una ventana de XWindows.

Por ejemplo, para abrir otro terminal desde un terminal se tipea:

```
/usr/bin/gnome-terminal &
```

Mas comandos

- bg
- bzip
- cat
- chgroup*
- chmod
- clear
- chown*
- exec
- df
- dmesg
- du
- export
- find
- gcc
- grep
- groupadd*
- groupdel*
- gzip
- history
- ifconfig*
- info
- init*
- insmod*
- kill
- ln
- locate
- lsmod
- lsof
- mkdir
- “|” (pipe)
- man
- mount*
- more
- netstat
- nmap
- ping
- printenv
- ps
- pwd
- route*
- rpm*
- rmdir
- rmmod*
- set
- su
- tail
- tar
- telinit*
- top
- touch
- traceroute*
- uname
- unset
- unzip
- usermod*
- users
- watch
- whereis
- which
- whoami

- “ctrl-u”
- date*
- hexdump
- hwclock*
- mkisofs
- tcpdump

* = root

Buscando mas informacion

No solo puedes usar comandos para encontrar informacion sobre su sistema, pero puedes ver adentro varios archivos, o explorar el directorio /proc.

Ejemplos de archivos con datos utiles:

- /etc/motd
- /etc/issue
- /etc/resolve.conf
- /etc/services
- /etc/X11/XF86Config
- /etc/modules.conf
- /etc/fstab

El editor vi

- Para que usar vi? Porque no a emacs, xemacs, joe, pico, etc.? (anota: “pico -w”)
- vi existe en casi todo los distribuciones de Linux/Unix/BSD.
- Si tienes que trabajar en una maquina nueva, casi siempre habra vi para configurarla.
- En realidad, probablemente vas a usar otro programa, como xemacs o joe, por proyectos mas grande, pero veamos que se puede hacer vi -->

Comandos basicos de vi

Impresiona a tus amigos...

- **Abrir:** vi fn, vi -r fn, vi + fn, vi +n fn, vi +/pat fn
- **Cerrar:** :w, :wq, :q, :q!
- **Movimiento:** h,j,k,l y w, W, b, B, :n (y, flechas)
- **Editar:** i, o, x, D, dd, yy, p
- **Buscar:** /patron, ?patron, n, N

Crear y eliminar cuentas usuarios

- /etc/passwd, /etc/group, /etc/shadow, /etc/sudoers
- /usr/sbin/adduser --> /usr/sbin/useradd
- /usr/sbin/userdel
- /etc/default/useradd
- /etc/skel
- /etc/login.defs
- /etc/profile
- chsh, passwd, groupadd, groupdel, groupmod, usermod

/etc/passwd

El archivo /etc/password tiene el siguiente formato (muestra):

```
hervey:x:500:500:Hervey Allen:/home/hervey:/bin/bash
```

Es decir:

```
usuario:pw:UID:GID:nombre:directorio:shell
```

Usando /etc/shadow el “pw” se representa con un “x”. Si el usuario representa algo como un servicio el “shell” esta representado con “/sbin/nologin”.

/etc/shadow

Se lo usa este archivo para esconder los claves encodificados. El archivo solo se puede leer el usuario root.

Tiene el formato asi (muestra):

```
hervey:$1$w!@F62adfk3$LCYjTI3udsd/tGP1pux1:12452:0:99999:7:::
```

Es Decir:

- usuario:
- pw encodificado con crypt:
- dias despues de 1 de enero 1970 desde que ha cambiado la contraseña:
- Dias hasta que se puede cambiar la contraseña:
- Dias despues de que tiene que cambiar la contraseña:

/etc/shadow continuado

```
hervey:$1$w!@F62adfk3$LCYjTI3udsd/tGP1pux1:12452:0:99999:7:::
```

- Dias antes que se vence la contraseña que se avisa el usuario:
- Dias despues que se vence la contraseña que se deshabilita la cuenta de usuario:
- Dias desde que 1 de enero 1970 de que la cuenta usuario ha estado deshabilitado:
- Reservado

/etc/group

En este archivo se especifica el nombre del grupo (no más de 8 caracteres), la clave, un numero de identificacion unica (GID), y una lista de miembro del grupo separado por “,”:

Por ejemplo:

```
mail:x:12:mail,postfix
```

Nota el “x” - Si hay una contraseña por el grupo esta en el archivo /etc/shadow.

¿Y, para que sirven los grupos?

Comandos – programas – shell – path

¿Que es un “comando” y un “programa”?

¿Para que no siempre puedo correr todo los comandos y programas?

¿Donde se arregla esto?

¿Donde vea que esta disponible?

- /etc/skel
- /etc/profile
- /home/user/.bashrc
- /home/user/.bash_profile
- set, printenv, export

Uso del comando su

Se usa “su” para asumnir otro identidad, como root, sin tener que logout y login como otro usuario.

Para usar su para cambiarse a “root” tiene que tener permiso por esto en /etc/sudoers.

Puede permitir usuarios correr algunos comandos privilegiados usando /etc/sudoers y “sudo”

Puede asignar gente al grupo “wheel” y usando /etc/sudoers se puede correr todo los comandos.

Mas uso del comando su

En vez de tener que tomar un shell como root, se puede correr un comando privilegiado asi:

```
sudo comando
```

Por ejemplo:

```
sudo mount /mnt/cdrom
```

Y, si quiere cambiarse por ser otro usuario, y correr sus scripts de login:

```
su - usuario
```

Instalar un paquete usando RPM

- RPM significa “Red Hat Package Manager”
- Hay varios sistemas para controlar y instalar software (rpm, dpackage/apt-get, source) y cada uno tiene ventajas y desventajas.
- RPM te permite instalar y remover software en tu sistema. Tambien, puedes ver los archivos que se va a instalar y donde.
- RPM no puede resolver las dependencias entre paquetes de software.

Usando RPM

Se puede hacer un monton de cosas con RPM, pero bien tipico es instalar:

rpm -Uvh nombre-de-paquete.rpm

rpm -ivh nombre-de-paquete.rpm

Preguntar si algo esta instalado:

rpm -qa | grep nombre-de-paquete

Preguntar sobre algo que ya esta instalado:

rpm -qi nombre-de-paquete

rpm -ql nombre-de-paquete

Usando RPM continuado

Cada persona usa RPM un poco diferente. Vea la diferencia entre “rpm -Uvh” y “rpm -ivh”, o si es realmente necesario usar la opcion “h”?

La clave con algo que trata de organizar tu software es si realmente funciona. RPM ayuda, pero no soluciona el problema de saber que tienes instalado, donde, y que pertenece a cada paquete de software.

Vea cuantas veces usarás “--nodeps” en el futuro...

Apagar y reinicializar un servidor

¿Como se apaga un maquina de Linux?

- shutdown -t 60 mensaje
- halt
- init/telinit 0

¿Y, para reinicializar?

- reboot
- shutdown -r ahora

Niveles de inicializacion

¿Y, que fue el “init/telinit 0”?

Linux tiene el concepto de nivels de initializacion. Cada nivel tiene un significado, y ellos son:

```
0 - Apagar. No haz esto como el “default”
1 - Modo de usuario solo
2 - Modo de multi-usuarios, pero sin sistema
    de archivo de la Red (NFS).
3 - Modo de multi-usuarios con NFS.
4 - No usado
5 - Corre X11 (Gnome, KDE, XWindows).
6 - reinitializar. No haz esto como “default”
```

Niveles de inicializacion continuado

El nivel de que empieza una maquina esta definido en /etc/inittab.

Casi siempre va a querer empezar su maquina en nivel 3 o 5 (excepcion - “single user”)

La regla basica es; “servidor, nivel 3, computador por uso personal, nivel 5”

Se puede cambiar nivels usando “telinit --> init”, es decir telinit es un alias por init.

Niveles de inicializacion continuado

Para apagar XWindows (Gnome, KDE) y trabajar en un shell (texto), abre un shell en XWindows y tipea:

```
init 3
```

Para reinicializar XWindows tipea:

```
init 5
```

Ahora, veamos como controlar que corre en cada nivel...

Servicios que corren en cada nivel

Es un poco complicado, pero los pedazos y herramientas que usamos son:

- /etc/rc.d/
- /etc/rc.d/rc0.d a rc6.d
- /etc/rc.d/init.d/
- /etc/inittab
- /sbin/service script opcion
- chkconfig

Controlar servicios

Primero, veamos /etc/rc.d/init.d. Aqui hay los scripts que controla la mayoria de servicios corriendo (¡ojo con xinetd!)

Paramos y reinicializamos nfslock.

Veamos en que niveles corre nfslock:

```
/sbin/chkconfig –list | grep nfslock
```

Ahora, vamos a usar chkconfig para remover nfslock de todo los niveles de inicializacion.

Controlar servicios continuado

Primero, vamos a /etc/rc.d/init.d/rc3.d y rc2.d para explicar que pasa en estes directorios.

Despues que esto, configuramos que nfslock no corre la proxima vez que inicializamos la maquina (en /etc/rc.d/init.d):

```
chkconfig –del nfslock
```

Vamos a jugar con chkconfig, los directorios, y servicios para entender que esta pasando.

Servicios resumen

Tal vez viste /etc/rc.d/rc, rc.sysinit, y rc.local. En otros sistemas estes archivos controla como corren los servicios mucho mas.

Este estructura de directorios que hemos visto esta llamado “system v” y viene de sistema operativos anterior al Linux.

Entender como parar temporarmente y permanente los servicios es clave para correr un servidor de Linux.

Resumen

Hemos visto algo de como corre un servidor de Linux. Mañana hablarémos mucho mas sobre la estructura de un servidor Linux y vamos a visitar de nuevo algunas de estas temas.

Ahora, si hay tiempo, volvemos al principio y practicamos con la lista mas grande de comandos.