

Administración Avanzada y Redes TCP/IP en UNIX

Gabriel Infante-Lopez

Facultad de Matemática, Astronomía y Física
UNC, Córdoba (Argentina)
<http://www.cs.famaf.unc.edu.ar/~gabriel>

Escuela Complutense Latinoamericana
7 de Febrero de 2006

contenidos

Arranque y Apagado del Sistema

Proceso init y runlevels

Parada del sistema

Kernel y módulos

Configuración del kernel

Lecturas Recomendadas

Ejercicios

Arranque y Apagado del Sistema

El arranque de un sistema Linux consta de las siguientes fases:

1. Ejecución del gestor de arranque (p.ej. lilo)
2. Carga y ejecución del kernel
3. Ejecución de init (proceso número 1)
4. Ejecución de scripts de iniciación genéricos en `/etc/rcS.d`
5. Entrada en el runlevel por defecto y ejecución de scripts del runlevel en `/etc/rcX.d`.

Kernel

1. Se ejecuta realmente el kernel
2. Realiza un lista de comprobaciones
3. Activa módulos internos del mismo.
4. monta el disco principal donde se almacena el sistema operativo
5. Ejecuta el primer proceso: init.

Init

El init tiene como misión

1. ejecutar el resto de procesos del sistema
2. comprobación de discos, detección/configuración de hardware adicional
3. apertura de terminales, servidores.

Lilo en mas Detalle - Donde Instalarse

Lilo se configura mediante `/etc/lilo.conf`.

```
boot=/dev/hda
```

especifica la unidad en cuyo sector de arranque en el que debe instalarse lilo. MBR del disco `a`

```
boot=/dev/hda2
```

No en el MBR, partición primaria. Debe ser activa.

Lilo en mas Detalle - Mensaje de Inicio

```
root=/dev/hda1
```

Especifica donde está la raiz (root) del sistema de ficheros de Linux.

```
message=/boot/bootmess.txt prompt delay=100  
timeout=100
```

En las versiones actuales, Lilo muestra automáticamente un menú con las opciones de arranque.

Lilo en mas Detalle - Parametros de Kernel

```
append="mem=256M hdc=ide-scsi"
```

- ▶ indica opciones adicionales que hay que pasar al kernel
- ▶ indica la cantidad de RAM instalada
- ▶ debe usarse la emulación IDE a SCSI

```
image=/vmlinuz label=Linux read-only
```

donde encontrar la imagen del kernel

Lilo en mas Detalle - Otros Sistemas

```
other=/dev/hda1 label = Windows alias = 3
```

Donde encontrar otros sistemas operativos

Proceso INIT y Runleves

- ▶ `Init` es el proceso número 1
- ▶ `Init` ejecuta otros programas.
- ▶ se configura en el fichero `/etc/inittab`.
- ▶ no hace nada por si mismo.
- ▶ Se ejecutan por orden alfabético todos los scripts en `/etc/rcS.d` que comiencen por `S` con `start` como argumento.
- ▶ Se nombran con número de dos cifras para establecer el un orden adecuado

Ejemplo de Scripts de Inicio

- ▶ `S10checkroot.sh` comprueba el sistema de ficheros.
- ▶ `S20adjtimex` ajusta el reloj del sistema.
- ▶ `S40networking` activa los interfaces de red.

Ejemplo de Scripts de Inicio

- ▶ estos scripts no se sitúan directamente en `/etc/rcS.d`
- ▶ se guardan en el directorio `/etc/init.d` y desde ahí se hacen enlaces simbólicos a `/etc/rcS.d`.
- ▶ Luego de `/etc/rcS.d` se ejecutan otra serie de scripts correspondientes a un `runlevel`.

Runleves

Los runlevels son un mecanismo para permitir que el ordenador trabaje con diferentes configuraciones de arranque (diferentes servicios, etc.).

Runlevels - Denominación

- ▶ Los runlevels se numeran del 0 al 6.
- ▶ 0 se ejecuta para parar el sistema (`halt`),
- ▶ 6 para reiniciar (`reboot`)
- ▶ 1 para arrancar en modo `single user`,
- ▶ El resto de los runlevels son para funcionamiento normal.
- ▶ El runlevel por defecto es el 2 (se configura en `/etc/inittab`),

Init - sigue cargando el sistema

- ▶ En Debian, los runlevels del 2 al 5 se configuran inicialmente de forma idéntica.
- ▶ Init suele continuar ejecutando los scripts del runlevel correspondiente situados en `/etc/rcX.d`,
- ▶ donde `X` es el número del runlevel.
- ▶ se ejecutan primero los scripts que comiencen con `K` con argumento `stop`
- ▶ luego los que comienzan con `S` con argumento `start`.
- ▶ Cada Script gestiona un servicio.
- ▶ Por que se ejecutan los script con argumento `stop`?

Cambiando de Runlevel

En cualquier momento, el administrador puede hacer que el sistema cambie a otro runlevel

```
[login@host homedir] $telinit 3
```

Parada del Sistema

La parada del sistema se produce cuando se entra en los `runlevel 0` o `6`.

```
apagar el sistema
```

Parada del Sistema - Shutdown

```
shutdown -r now
```

reinicia el sistema inmediatamente (reboot).

Parada del Sistema - Shutdown

```
shutdown -h now
```

para el sistema inmediatamente (reboot).

Parada del Sistema - Shutdown

```
shutdown -h +10 "Vamos a parar el sistema en 10 min."
```

para el sistema dentro de 10 minutos enviando un aviso a todos los terminales.

Parada del Sistema - Shutdown

```
shutdown -r 20:00
```

reinicia el sistema a las 20:00 horas.

Kernel y Modulos

1. interacción final con dispositivos la realizan los controladores o el kernel.
2. Las fuentes en C del kernel cuentan con controladores para diversos dispositivos.
3. Algunos de esos controladores pueden unirse con el kernel mismo (estáticamente),
4. otros pueden dejarse como módulos para cargarse/descargarse.
5. otros pueden ser excluidos.
6. están ubicados en el directorio `/lib/modules/version`, organizados en directorios que indican el tipo de dispositivo o el propósito,

Configuración de Módulos

Para lograr configurar un dispositivo controlado por un módulo, puede emplear las herramientas del paquete modutils o modconf para:

- ▶ Asegurar que no haya conflictos entre el dispositivo con otros y eventualmente conocer la configuración que usa (algunos controladores autodetectan la configuración del dispositivo, pero no todos).
- ▶ Encontrar un módulo que pueda manejar el dispositivo.
- ▶ Eventualmente pasar opciones al módulo de acuerdo a la configuración del dispositivo (e.g IRQ o dirección base).
- ▶ el paquete modutils se configura con `/etc/modutils` (con la información de esos archivos y de `/etc/modules` se genera el archivo de configuración de módulos `/etc/modules.conf`).

Herramientas modutils - lsmod

```
[login@host homedir] $lsmod
```

Lista los módulos cargados, de cada uno presenta nombre, tamaño, cuenta de usos y lista de módulos que lo usan (es equivalente a `cat /proc/modules`).

Herramientas modutils - rmmmod

```
[login@host homedir] $rmmmod modulos
```

Descarga uno o más módulos cargados, mientras estos no están siendo usados. Con la opción `-r` intenta descargar recursivamente módulos de los cuales el módulo especificado dependa.

```
Descargue todos los modulos que no estan siendo  
usados
```

Herramientas modutils - insmod

```
[login@host homedir] $insmod módulo [opciones]
```

- ▶ Trata de cargar el módulo especificado.
- ▶ Las opciones dependen del modulo específico.
- ▶ Puede indicarse una ruta no estándar en la variable MODPATH o en /etc/modules.conf.
- ▶ deben ser compilados para una versión precisa, o con la opción -f puede evitarse el chequeo de versiones.

Busqueda de Modulos

Es posible que el kernel del que disponga no cuente con módulos que soporten ciertos dispositivos, en tal caso se puede:

1. buscar en Internet un módulo precompilado para la versión del kernel que tiene,
2. buscar fuentes de un módulo o
3. buscar una versión del kernel que incluya soporte para el dispositivo.
4. En el último caso las instrucciones de la siguiente sección le serán de utilidad.

Configuración del kernel

- ▶ algunas características pueden ser modificadas en funcionamiento.
- ▶ escribiendo en algunos archivos del directorio `/proc`, o
- ▶ con el programa `sysctl`.
- ▶ la opción `-a` presenta todas las variables modificables y su valor.
- ▶ Para cambiar un valor se emplea:

```
[login@host homedir] $/sbin/sysctl -w variable=valor
```

Configuración del kernel

El kernel se puede recompilar para modificar el comportamiento de módulos.

- ▶ Limpieza: Para limpiar posibles compilaciones
- ▶ Definición del nombre (`uname -r`). Variable `EXTRAVERSION=-tobias`
- ▶ Configuración del Kernel: `make menuconfig`
- ▶ Compilación del Kernel
- ▶ Compilación de los módulos
- ▶ instalación del kernel
- ▶ configuración del LILO

Lecturas Recomendadas

1. Para manejar mejor las herramientas de módulos, puede consultar las páginas man de `modules.conf`, `insmod`, `modprobe`
2. "Introduction to Linux Kernel Modules" en <http://www.luv.asn.au/overheads/kernelmodules/>
3. modulos que requieren ser compilados.
<http://www.scyld.com/expert/modules.html>
4. <http://structio.sourceforge.net/paz/actualiz-kernel.htm>

Ejercicios

1. ¿Que versión del kernel Linux tiene instalado su sistema?
Ayuda: emplee el programa `kernelversion`
2. Revise los módulos que están cargados en su sistema, y trate de determinar que dispositivo controla o qué servicio presta cada uno.
3. Examine el archivo de configuración de X-Window, si desea experimentar con este, saque una copia y modifíquelo directamente o por medio del programa `XF86Setup` (si no tiene disponible `XF86Setup` puede intentar con `xf86config`).
4. Al cargar un módulo, `insmod` revisa que los símbolos del kernel que el módulo espera puedan ser resueltos.
5. ¿Qué son estos símbolos? Describa también como buscó su respuesta.