



Arranque del sistema Linux



Nota de Copyright

© 2006 Diego Chaparro. Algunos derechos reservados.

Este trabajo se distribuye bajo la licencia Creative Commons Attribution-ShareAlike. Para obtener la licencia completa, véase <http://creativecommons.org/licenses/by-sa/2.1/es>



Arranque del sistema

- Independientemente de como arranque el sistema, lo primero que se ejecuta es el cargador (boot loader)
- El kernel debe estar en un disquete o en el directorio /boot
- El nombre del kernel suele ser: vmlinuz-version
- Lo primero: se carga el kernel en memoria
- Cuando se está cargando el kernel va mostrando una serie de mensajes
 - Estos mensajes los guarda en /var/log/messages
 - Se pueden ver también con el comando dmesg



Arranque del sistema

- Estos mensajes incluyen información sobre:
 - Detección bus PCI y dispositivos PCI que detecta
 - Estimación de la velocidad del procesador
 - Cantidad de memoria
 - Tipo de procesador
 - Versión del kernel
 - Drivers de dispositivos cargados
 - Cantidad de memoria swap
 - Adaptadores de red y configuraciones



Arranque del sistema

- **inittab**

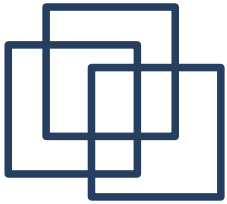
- Lo último que hace el kernel es iniciar el proceso init, que permanece activo hasta que se apaga la máquina
- Es el encargado de crear el resto de procesos
- Por ejemplo, cuando salimos de una consola, éste proceso reinicia el login de esa consola
- El funcionamiento de este proceso está indicado en el contenido de su fichero de configuración: /etc/inittab



Arranque del sistema

- **/etc/inittab**

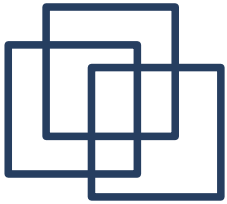
- Cada línea de este fichero contiene lo siguiente:
 - ID:runlevel:action:process
 - ID Uno o dos caracteres identificando esa entrada
 - runlevel Indica a qué runlevel se aplica esa línea.
 - action Cómo manejar las entradas
 - boot Ejecutado cuando /etc/inittab es leído por primera vez
 - bootwait Ejecutado después de las entradas de arranque
 - initdefault Configura el runlevel inicial
 - ondemand Reinicia los procesos si se paran
 - powerfail Ejecutado cuando recibe una señal de fallo de energía
 - sysinit Ejecutado antes de acceder a la consola
 - respawn Mantiene el proceso ejecutando, reinicia si se para
 - wait Arranca el proceso una vez
 - process Comando a ejecutar



Arranque del sistema

- **runlevels**

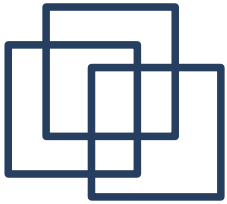
- Un nivel de arranque define un conjunto de procesos a arrancar como parte del arranque del sistema
- Este rango puede ser una cantidad mínima de procesos para administrar la máquina o todos los procesos para tener todos los dispositivos configurados
- El número de runlevels está definido en /etc/inittab:
 - 0 halt
 - 1 single user
 - 2 multiuser
 - 3 full multiuser mode
 - 4 unused
 - 5 X11
 - 6 reboot



Arranque del sistema

- **runlevels**

- Estos runlevels han sido siempre los clásicos, aunque ahora solo se usan:
 - 0 halt
 - 1 single user
 - 2-5 multiuser
 - 6 reboot
- El 0 y el 6 se usan para parar y rearrancar el sistema respectivamente
- El 1 es diferente de los 2-5, se usa para realizar tareas de administración, como recuperar el sistema de un sistema de ficheros corrupto. Nadie más que el root puede hacer login



Arranque del sistema

- **runlevels**

- Cuando se arranca en runlevels de 2-5, se ejecutan las líneas con acciones `sysinit`, `boot` y `bootwait`
 - `id:3:initdefault:`
 - Define el runlevel por defecto
 - `si::sysinit:/etc/rc.d/rc.sysinit`
 - Define un script para ejecutar al arrancar
 - La siguiente sección define los scripts a ejecutar en cada runlevel:
 - `10:0:wait:/etc/init.d/rc 0`
 - `11:1:wait:/etc/init.d/rc 1`
 - ...



Arranque del sistema

- **/etc/inittab**

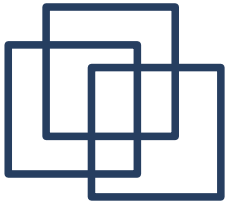
- Después especifica qué ejecutar cuando se detectan cambios en la corriente del sistema, normalmente por una UPS:
 - `pf::powerwait:/etc/init.d/powerfail start`
 - `pn::powerfailnow:/etc/init.d/powerfail now`
 - `po::powerokwait:/etc/init.d/powerfail stop`
- El proceso `init` permanece activo después de leer `/etc/inittab`. Cuando se tiene que cambiar el `runlevel` o cuando un proceso que inició se para, hace dos cosas:
 - Lee el `/etc/inittab` y comprueba si hay que reiniciar el proceso, y lo hace
 - Graba esta información en `/etc/wtmp` o `/etc/utmp`



Arranque del sistema

- **/etc/inittab**

- El contenido del fichero se puede cambiar editando el fichero, pero los cambios no se efectuarán hasta que el proceso init re-lea el fichero, y esto ocurre:
 - Cuando se cambia el runlevel
 - Ocurre un fallo de corriente
 - Se ejecuta: `init q`



Arranque del sistema

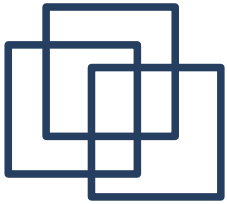
- **init/telinit**

- Se usa para cambiar el runlevel
- Ejemplo:
 - `init 5`
 - `telinit 5`
- Se le puede especificar cuando tiempo debe esperar desde que le manda un SIGTERM a los procesos hasta que les manda un SIGKILL (por defecto 5 segundos)
 - `telinit -t 10 5`



Arranque del sistema

- **Procesos de cada nivel de ejecución**
 - En /etc/rcX.d/
 -



Arranque del sistema

- En el fichero **/etc/modules.conf** o **/etc/conf.modules** se definen algunos parámetros para algunos módulos, como interrupciones, direcciones entrada/salida
- Ejemplos:
 - alias eth0 3c59x
 - alias sound es1370
- En el fichero **/etc/modules** se especifican los módulos que queremos cargar al arrancar



LILLO

- Linux LOader: es el cargador del sistema que se encuentra en el disco duro o en un disquete
- Reemplaza el MBR del disco
- Puede arrancar hasta 16 imágenes diferentes de diferentes particiones, cada una con su contraseña
- Cuando aparece

LILLO

- sin prompt hay que pulsar ALT, SHIFT o CTRL para que aparezca el prompt del LILLO
- Después podemos pulsar TAB para que se muestren todas las opciones existentes



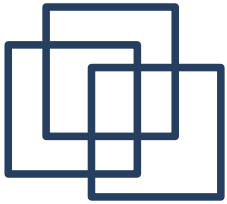
LILO

- Se le pueden pasar varias opciones al seleccionar el kernel:
 - `kernel single` Arranca en modo monousuario
 - `kernel root=device,ro` Arranca de esa partición en read-only
- **`/etc/lilo.conf`**
 - Es donde se guarda la configuración del LILO
 - `boot=/dev/hda`
 - Define donde se instala el lilo
 - `root=/dev/hda3`
 - Partición que se montará como /
 - `delay=5`
 - Cinco segundos que espera antes de arrancar el kernel por defecto



/etc/lilo.conf

- Es donde se guarda la configuración del LILO
- prompt
 - Sale el prompt automáticamente
- timeout
 - Cuando ponemos prompt, especifica cuanto tiempo espera antes de arrancar el kernel por defecto
- default=Linux-2.2.20
 - Especifica el kernel por defecto
- image=/boot/vmlinuz-2.4.20
 - Especifica una imagen del kernel
- other=/dev/hda1
 - Especifica otro tipo de sistema que arranca desde esa partición



Parada del sistema

- Solo el root puede parar el sistema a no ser que el usuario aparezca en `/etc/shutdown.allow`
- Se manda la señal `SIGTERM` y luego `SIGKILL` a todos los procesos
- Se ejecutan los scripts en el nivel 0 o 6
- **reboot**
 - Reinicia el sistema
- **halt**
 - Apaga el sistema



shutdown

- Apaga el sistema
- Sintaxis: shutdown [-t seg] [opciones] hora [warning-mensaje]
- Opciones:
 - k No apaga, solo manda el warning
 - r Reinicia el sistema
 - h Apaga el sistema
 - f No hace fsck al reiniciar
 - F Fuerza fsck al reiniciar
 - c Cancela un shutdown en ejecución
- La hora se puede poner en varios formatos:
 - hh:mm
 - +m Número de minutos en adelante
 - now Es igual a poner +0