

---

# Unidad 2: Arranque y Parada del sistema: Iniciación de servicios

**VI Curso de Introducción a la Administración de Servidores GNU/LINUX**  
**Extensión Universitaria. Universidad de Sevilla**  
**Febrero-Abril 2009**

Autores: Manuel J. Bellido, Jorge Juan <jjchico@gmail.com>.  
Usted es libre de copiar, distribuir y comunicar públicamente la obra y de hacer obras derivadas bajo las condiciones de la licencia Attribution-Share alike de Creative Commons.  
Puede consultar el texto completo de la licencia en <http://creativecommons.org/licenses/by-sa/3.0/>

## Contenidos

---

- Proceso de arranque del sistema
- Gestor de arranque GRUB
- Arranque del kernel
- Proceso init y *runlevels*
- Tareas típicas
- Parada del sistema
- Anexo: control de procesos

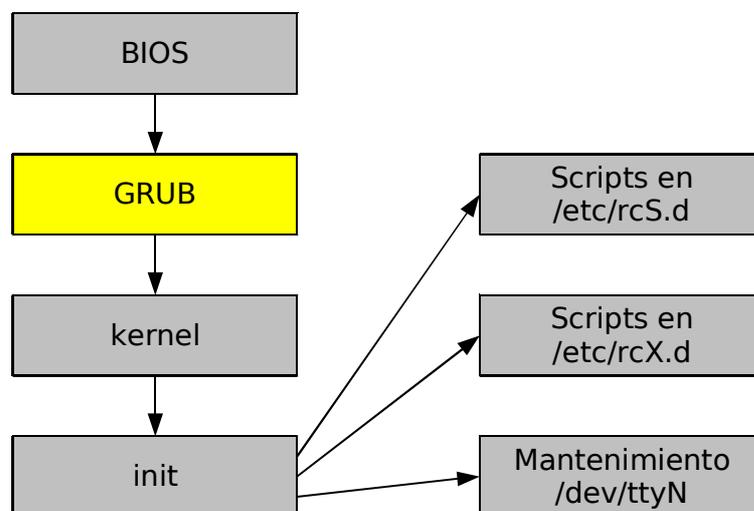
# Proceso de Arranque

---

- Fases del Proceso de arranque:
  - Selección del dispositivo de arranque (BIOS)
  - Ejecución del gestor de arranque desde el primer bloque del dispositivo de arranque
  - (Carga y ejecución de la segunda fase del gestor de arranque)
  - Carga y ejecución del kernel del S.O.
  - Ejecución del proceso init
  - Ejecución de scripts de iniciación (en /etc/rcS.d)
  - Entrada en el runlevel por defecto: ejecución de Scripts en /etc/rcX.d, donde X es el número del runlevel

# Proceso de arranque

---



# Gestor de arranque GRUB

---

- Permite iniciar diversos sistemas operativos desde un menú de arranque
- Se configura e instala durante el proceso de instalación, detectando otros SS.OO.
- Archivo de configuración: `/boot/grub/menu.lst`
  - Normalmente no es necesario editar este archivo porque el sistema lo mantiene automáticamente

# Gestor de arranque GRUB

---

- Fallos en el gestor de arranque
  - Se ha instalado otro sistema operativo (y ha borrado el sector de arranque)
  - La instalación no ha finalizado correctamente
  - Ha cambiado el número de dispositivo asignado por la BIOS
    - Se ha trasladado el disco a otro sistema
    - Se ha cambiado la configuración de la BIOS
  - Se han reescrito los archivos de grub en `/boot/grub`
- En estos casos es necesario reinstalar la fase 1 de grub

# Gestor de arranque GRUB

---

- Instalación de grub desde el sistema ejecutado
  - # grub-install '(hd0)'
    - Instala la fase 1 de grub en el primer disco reconocido por la BIOS
  - Alternativamente, puede indicarse un dispositivo linux o una partición. Ej: /dev/hda, /dev/sdb1
  - La correspondencia entre dispositivos y números de la BIOS está en /boot/grub/device.map
    - Puede editarse este fichero si la correspondencia no fuera la correcta, pero no suele ser necesario.

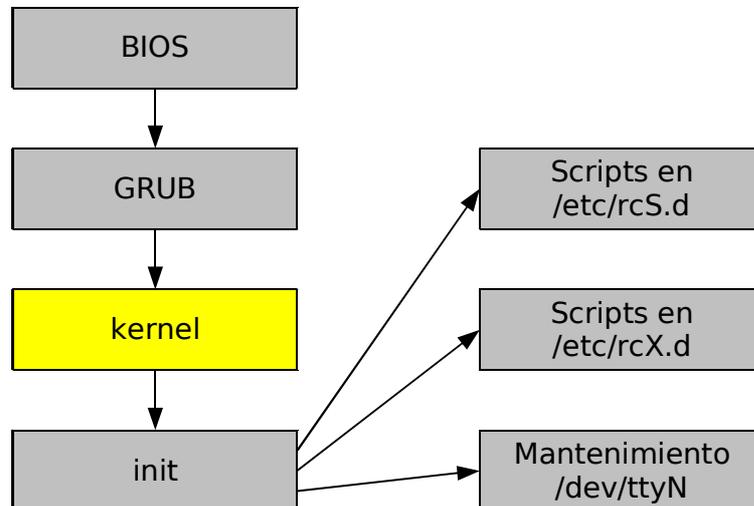
# Gestor de arranque GRUB

---

- Reinstalación de grub en un sistema que no se puede arrancar
  - Iniciar el sistema desde un dispositivo alternativo, ej: disco de instalación, disco de rescate, etc.
  - Montar la partición del sistema y hacer accesible /dev
    - # mkdir /mnt/tmp
    - # mount /dev/sda5 /mnt
    - # mount --bind /dev /mnt/tmp/dev
  - Ejecutar grub-install desde chroot:
    - # chroot /mnt/tmp /usr/sbin/grub-install /dev/sda
  - Desmontar y reiniciar el sistema
    - # umount /mnt/tmp/dev; umount /mnt/tmp; reboot

# Arranque del kernel

---



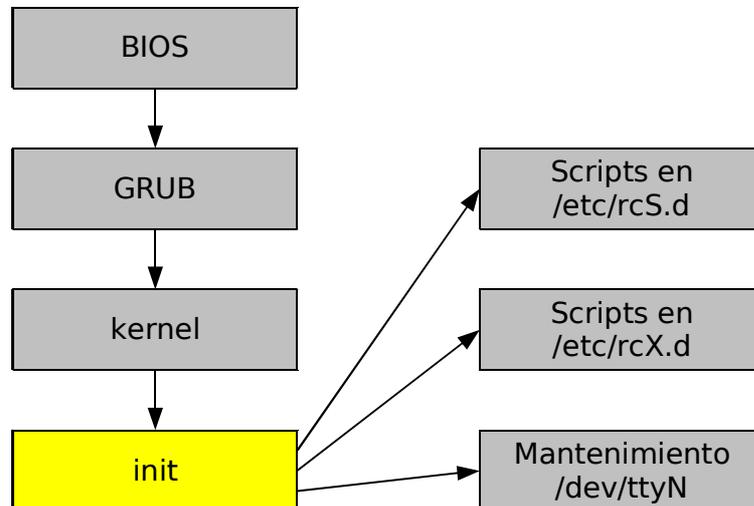
# Arranque del kernel

---

- Grub
  - Archivo con el kernel
  - Archivo con disco virtual (initrd) a cargar en memoria
  - Parámetros de arranque del kernel
    - Partición del sistema de archivos raíz (root=...)
    - Otros parámetros del kernel y de programas de inicio
- Kernel
  - Descompresión y arranque inicial: detección de aspectos básicos del hardware (microprocesador, etc.)
  - Ejecución de procesos en initrd: detección del hardware, carga de drivers esenciales, etc.
  - Montaje del sistema de archivos raíz.

# Proceso de arranque

---



# Proceso init y runlevels

---

- Una vez cargado el kernel se ejecuta el programa init que se encarga del resto del arranque.
- Versiones de init:
  - Init clásico (sysvinit)
  - Upstart (Ubuntu): compatible con sysvinit
- Configuración
  - /etc/event.d
  - /etc/inittab: crear uno para cambiar runlevel por defecto.

# Proceso init y runlevels

---

- Runlevel
  - define diferentes configuraciones del arranque/parada del sistema mediante un conjunto de scripts que se ejecutan para cada runlevel
- Runlevels definidos
  - 0: parar el sistema (halt)
  - 1: modo monousuario. Sólo el root tiene acceso, se desactivan todos los procesos, red, etc.
  - 2-5: modos multiusuario configurables para operación normal (por defecto, 2)
  - 6: reiniciar el sistema (reboot)

# Proceso init y runlevels

---

- Proceso de inicio
  - Se ejecuta “init”
  - Se ejecutan por orden alfabético todos los scripts de la carpeta `/etc/rcS.d` que comiencen por “S” con la opción “start”
  - Se ejecutan por orden alfabético los scripts del runlevel por defecto (normalmente 2 -- `/etc/rc2.d --`) con la opción:
    - Si comienzan con “K”: stop
    - Si comienzan con “S”: start
  - Se puede activar/desactivar servicios o tareas borrando o renombrando scripts en `/etc/rcN.d`

## Proceso init y runlevels

---

- En cualquier momento el administrador del sistema puede cambiar de runlevel:
  - comando:
    - # telinit <numero\_runlevel>
- Especialmente útil para tareas de administración es el runlevel 1 (monousuario)
  - desconexión de la red
  - desmontar sistemas de archivos
  - etc.

## Proceso init y runlevels

---

- Scripts de inicio:
  - Enlazados desde /etc/rcS y /etc/rcN
  - Ubicados realmente en /etc/init.d/\*
  - Cada script controla una tarea
  - Cada script puede ejecutarse con una opción que determina la acción a realizar:
    - start: inicia el servicio o tarea correspondiente
    - stop: detiene el servicio o tarea correspondiente
    - restart: reinicia el servicio o tarea
    - reload: recarga la configuración
  - Muchos scripts leen parámetros de archivos en /etc/default

# Proceso init y runlevels

---

- Muchos scripts controlan servicios y pueden usarse individualmente:
  - # /etc/init.d/apache stop
  - # /etc/init.d/samba reload
  - # /etc/init.d/ssh start
- /etc/rc.local
  - Se ejecuta al final del proceso de arranque
  - Editable por el administrador para ejecutar tareas "locales" (no estándar)
  - Ej: Configuración adicional de la red, de un firewall, etc.

# Tareas típicas

---

- Añadir/quitar opciones de arranque del kernel
  - /boot/grub/menu.lst
- Configurar arranque de otro sistema operativo
  - /boot/grub/menu.lst
- Iniciar, parar o reiniciar un servicio
  - /etc/init.d/<servicio> start/stop/restart/reload
- Hacer que un servicio no se ejecute al inicio, sin desinstalar el servicio
  - Renombrar script en /etc/rc2.d. Ej.
    - /etc/rc2.d/S16ssh -> /etc/rc2.d/XS16ssh

# Parada del sistema

---

- Para parar el sistema es necesario realizar varias tareas de forma ordenada. Esto se controla mediante los runlevels 0 (parar) y 6 (reiniciar) y el comando “shutdown”
- Formas de parar o reiniciar el sistema
  - # shutdown -r now
  - # shutdown -h +5 “paramos en 5 minutos”
  - # telinit 0
  - En el escritorio: Sistema -> Salir
  - En consola de texto: Ctrl-Alt-Supr
  - Pulsando el botón de encendido/apagado

# Anexo: control de procesos

---

- Proceso: programa en ejecución
- Trabajo (job): proceso ejecutándose bajo el control de un terminal
- Estados de un proceso:
  - ejecutándose, durmiendo, detenido, zombie
- Acciones sobre procesos:
  - envío de señales
  - cambiar la prioridad

# Anexo: control de procesos

---

- Control de procesos gráfico
  - gnome-system-monitor
- Comando para el control de procesos
  - ps: listar procesos
  - nice: cambiar prioridad
  - kill/killall: enviar señales
  - top/htop: control interactivo
- Control de trabajos
  - jobs: lista de trabajos del terminal
  - bg/fg: enviar a segundo/primer plano