

---

# Unidad 4: Arranque y Parada. Gestor de Arranque. Ejecución Diferida.

**Centro de Formación Permanente  
Universidad de Sevilla**

Jorge Juan, Abril 2013.

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.

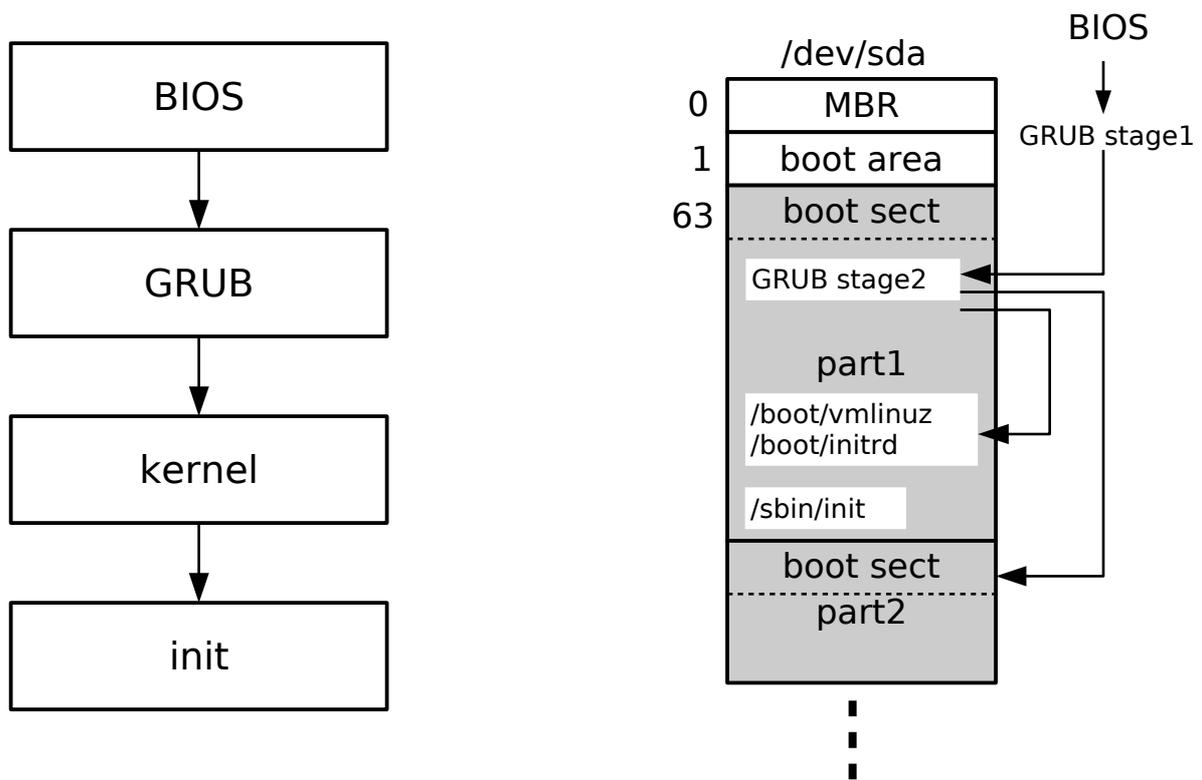
Puede consultar el texto completo de la licencia en <http://creativecommons.org/licenses/by-sa/3.0/>

---

## Contenidos

- Proceso de arranque del sistema
- Gestor de arranque GRUB2
- Arranque del kernel
- Proceso init y *runlevels*
- Tareas típicas
- Parada del sistema
- Ejecución diferida (at)
- Planificador de tareas (cron)

# Proceso de arranque



# 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

# Gestor de arranque Grub2

---

- Grub2 en Ubuntu: <https://wiki.ubuntu.com/Grub2>
- Permite iniciar diversos sistemas operativos desde un menú de arranque
- Se configura e instala durante el proceso de instalación, detectando otros SS.OO.
- La configuración se gestiona automáticamente
- Archivos de configuración
  - /boot/grub/grub.cfg: generado automáticamente. ¡No modificar!
  - /etc/default/grub: opciones generales. Se puede editar.
  - /etc/grub.d/: carpeta con archivos de configuración
    - 40\_custom: configuraciones manuales
- update-grub: tras hacer modificaciones en la conf.

## Problemas con Grub

---

- Tipo 1: Fase 1 no está en el MBR
  - Se ha instalado otro sistema operativo
  - La instalación no ha finalizado correctamente
  - Solución: reinstalar grub en el disco (grub-install)
- Tipo 2: Carga la fase 1 pero no encuentra una configuración correcta
  - Se ha trasladado el disco a otro sistema
  - La actualización de la configuración no funcionó correctamente
  - Se ha intentado (sin éxito) cambiar manualmente la configuración.
  - Solución: actualizar (regenerar) la configuración de Grub

# Problemas con Grub

- Solución genérica
  - Arranque "live" desde una imagen de instalación (CD / USB)
    - ¡Misma arquitectura que el sistema a reparar (32/64 bits)!
  - Localizar disco donde se quiere instalar Grub
    - El disco desde el que vaya a arrancar la BIOS (Ej: /dev/sda)
  - Localizar partición o particiones donde está instalado el sistema
    - Carpeta '/' y '/boot' (si está en partición aparte). Ej: /dev/sda5, /dev/sda6
  - Montar el sistema principal
  - Desde el sistema principal (chroot)
    - Reinstalar Grub en MBR
    - Ejecutar una actualización de la configuración desde el sistema principal (chroot)

# Problemas con Grub Solución genérica

Localizamos particiones  
/dev/sda5 (/),  
/dev/sda6 (/boot)

Montamos el sistema  
en /mnt

Cambiamos al sistema y  
ejecutamos la instalación y  
configuración de Grub

Salimos del chroot,  
desmontamos y reiniciamos

```
# fdisk -l
...

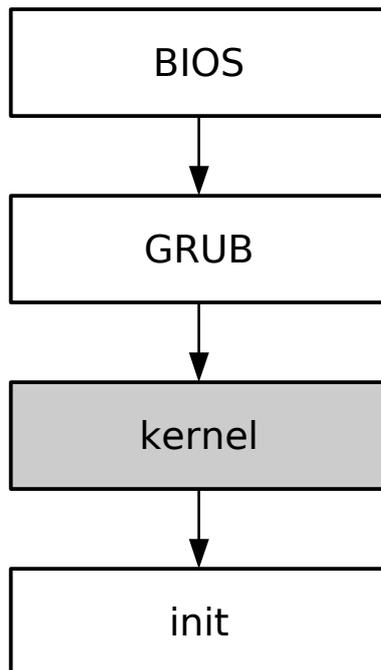
# mount /dev/sda5 /mnt
# mount /dev/sda6 /mnt/boot
# mount --bind /dev /mnt/dev
# mount --bind /proc /mnt/proc
# mount --bind /sys /mnt/sys

# chroot /mnt
(chroot)# grub-install /dev/sda
(chroot)# update-grub
...

(chroot)# exit
# umount /mnt/dev
# umount /mnt/proc
# umount /mnt/sys
# umount /mnt/boot
# umount /mnt
# 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 (dmesg)
  - 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.

# Configuración initrd

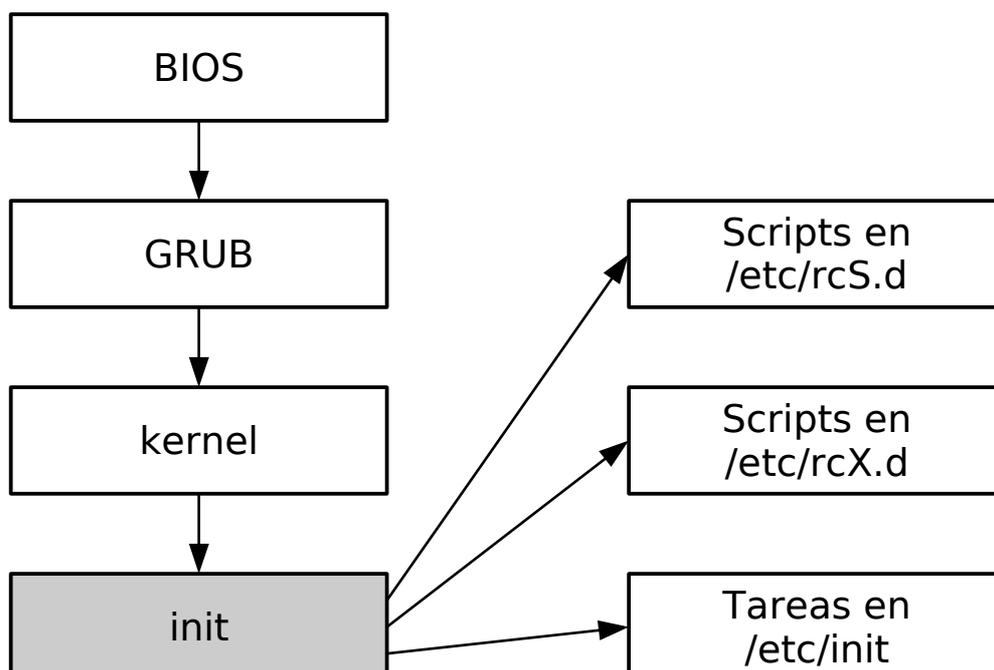
---

- Initrd: sistema de archivos con los módulos/drivers esenciales para iniciar el sistema.
- Es cargado en RAM por el gestor de arranque (Grub)
- Los "initrd" se generan automáticamente cuando se instala/actualiza un nuevo kernel.
- Se pueden controlar algunos aspectos de su configuración: /etc/initramfs-tools
- Se puede generar/actualizar un initrd a mano (ej. tras un cambio de configuración en /etc/initramfs-tools)

```
# update-initramfs -u -k 2.6.31-20-generic  
...
```

# Proceso de arranque

---



# Proceso init y runlevels

---

- Una vez cargado el kernel se ejecuta el programa init que se encarga del resto del arranque.
- Init System V (tradicional)
  - /etc/inittab
    - Configuración básica y runlevel
  - /etc/init.d, /etc/rcS.d, /etc/rc[123456].d
    - Scripts de arranque
- Init Upstart (dirigido por eventos)
  - /etc/init
    - Configuración de tareas (jobs)

# 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

---

- 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

---

- Proceso de inicio System V
  - 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

---

- Scripts de inicio (System V)
  - 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

```
# /etc/init.d/cups stop  
* Stopping Common Unix Printing System: cupsd [ OK ]
```

# Proceso init y runlevels

---

- Proceso de inicio Upstart
  - Controlado por eventos: se ejecutan tareas según eventos generados en el sistema y no en un orden predeterminado.
  - /etc/init: configuración de tareas y los eventos a los que reaccionan.
    - Hay tareas que ejecutan compatibilidad con System V
  - initctl: permite controlar tareas y emitir eventos

# Proceso init y runlevels

---

- “service”: permite controlar tareas gestionadas por scripts System V o por Upstart

```
# service cups start
* Starting Common Unix Printing System: cupsd [ OK ]
```

- /etc/rc.local
  - Script que 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.

---

## 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

# Ejecución diferida: at

---

- Se pueden programar tareas a realizar en una fecha y hora determinada con el comando **at**.
- Ejemplos:
  - at 3am
    - Ejecuta comandos dados en la entrada estándar a las 3:00 horas
  - at -f task 17:25 feb 28
    - Ejecuta comandos listados en el fichero task el 28 de febrero a las 17:25
  - La lista de comandos en la entrada estándar se finaliza con un carácter de fin de fichero (Ctrl-D)

```
$ at 0:0 jan 1 2007
warning: commands will be executed using /bin/sh
at> mpack -s "Feliz año 2007" feliz_2007.html all@us.es
at> <EOT>
job 8 at 2007-01-01 00:00
```

# Ejecución diferida: at

---

- La lista de tareas programadas se puede consultar con **atq**, y pueden cancelarse con **atrm**
- Ejecutados como súper usuario, pueden controlarse los trabajos programados por cualquier usuario.

```
$ atq
7      2007-02-28 17:25 a jjchico
8      2007-01-01 00:00 a jjchico
$ atrm 8
```

# Crontab del sistema

---

- El sistema cron ejecuta periódicamente tareas programadas en tablas de tareas del sistema.
- En Debian y derivados, la configuración por defecto facilita insertar o eliminar nuevas tareas periódicas y automatizar la configuración de cron al instalar o desinstalar paquetes de software
  - /etc/crontab: crontab principal del sistema. En general, no es necesario modificarlo
  - /etc/cron.d/: cada archivo en este directorio se trata como un crontab del sistema. Facilita añadir nuevas tareas
  - /etc/cron.[hourly/daily/weekly/monthly]/: cada carpeta contiene scripts que se ejecutan una vez a la hora, día, semana o mes

# Crontab del sistema

---

- El formato de los crontab del sistema en /etc/crontab y /etc/cron.d/ se describe en la página de manual **crontab(5)**

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts
--report /etc/cron.daily )
47 6 * * 7 * root    test -x /usr/sbin/anacron || ( cd / && run-parts
--report /etc/cron.weekly )
52 6 1 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts
--report /etc/cron.monthly )
```

# Crontab del sistema

---

- En Debian y derivados, puede (y debe) instalarse **anacron** cuando el sistema no está operativo de forma continua. La configuración por defecto del sistema compatibiliza el uso conjunto de cron y anacron
- Anacron ejecuta las tareas pendientes cuando se inicia el sistema, como aquellas tareas en `/etc/cron.daily/weekly/monthly`.

# Crontab de usuarios

---

- Cada usuario posee un fichero de texto donde se indican las tareas programada: crontab
- Este fichero no se edita directamente, sino a través del comando **crontab**
  - `crontab -e`
    - Edita el crontab del usuario
  - `crontab -f file`
    - Instala *file* como crontab del usuario
- El super usuario puede editar el crontab de cualquier usuario usando la opción `-u`
  - `crontab -u jjchico -e`

# Crontab de usuarios

---

- El formato del crontab se describe en la página de manual **crontab(5)**

```
# Ejemplo original (en inglés) de Paul Vixie
# Usar /bin/sh para ejecutar comandos, no configuración en /etc/passwd
SHELL=/bin/sh
# envía salida a `pablo`, no importa de quién sea este crontab
MAILTO=pablo
#
# m h dm mes ds comando
# ejecuta 5 minutos después de medianoche, todos los días
5 0 * * * $HOME/bin/daily.job >> $HOME/tmp/out 2>&1
# ejecuta a las 2:15pm el primero de cada mes – salida enviada a pablo
15 14 1 * * $HOME/bin/monthly
# ejecuta a las 10pm días entre semana – recordar a Juan
0 22 * * 1-5 mail -s "Son las 10pm" juan%Juan,%iRecoje a los niños!%
23 0-23/2 * * * echo "23 minutos después de meianoche, 2am, 4am..."
5 4 * * sun echo "ejecuta a las 4 y 10, todos los domingos"
```

## Crontab de usuarios. Control de acceso

---

- El uso de at y cron puede restringirse mediante los ficheros *at.allow*, *at.deny*, *cron.allow* y *cron.deny* en */etc*
  - Estos ficheros contienen nombres de usuario, uno en cada línea
- Se permite el uso de at a un usuario si y sólo si
  - está listado en *at.allow*, o bien
  - no está listado en *at.deny*, pero el fichero existe (defecto)
- Se permite el uso de cron a un usuario si y sólo si
  - está listado en *cron.allow*, o bien
  - *cron.allow* no existe y no está listado en *cron.deny*, o bien
  - no existen *cron.allow* ni *cron.deny* (defecto)