

Presentation Title

Subheading goes here



Todo sistema tiene:

- **Proceso de arranque del sistema**

Prepara el sistema para ser usado por el usuario

- **Proceso de parada del server**

Deja el sistema consistente (por ejemplo, los bloques de disco modificados en memoria se guardan en el sistema de ficheros)

El administrador debe conocerlos para modificarlo y facilitar el diagnóstico de errores y resolverlos



Arranque del sistema

- Proceso que inicia el gestor de arranque que es un programa ejecutado por el BIOS cuando se enciende una computadora. Se encarga de la inicialización del sistema operativo y de los dispositivos (boot/booting)



Proceso de arranque

BIOS

Basic **I**nput **O**utput **S**ystem -> comprueba hardware
Busca y carga el MBR

MBR

Master **B**oot **R**ecord (sector de arranque)
Carga en memoria el gestor de arranque (bootloader)

GRUB

GRand Unified Bootloaders -> /boot/grub/grub.conf
Muestra un menú con todos los sistemas operativos encontrados.
Ejecuta el kernel

Kernel

El kernel se encuentra comprimido en un archivo, que se descomprime y carga en memoria.
Se ejecuta y deja operacional el sistema a nivel de procesos.

systemd

Sistema y administrador de servicios.
Es el PID1 (ya no lo es init)
Inicio en paralelo de servicios y procesos. (inicio rápido y flexible)
/usr/lib/systemd/ (no usa /etc/inittab)
/etc/systemd
/lib/systemd

Proceso de arranque: etapas

Primera Etapa: La BIOS

- Al encender el equipo toma el control la BIOS, la cual detecta la configuración del hardware.
- Busca entre los dispositivos de almacenamiento el MBR (Registro Maestro de Arranque) y desde allí carga en memoria el gestor de arranque y se le pasa el control.

Segunda Etapa: El gestor de arranque GRUB

- Se ejecuta el gestor de arranque y muestra el menú de inicio de GRUB.
- Se selecciona el sistema operativo que se quiere arrancar, se carga en memoria y se le pasa el control.

Proceso de arranque: etapas

Tercera Etapa: El Kernel

- Se descomprime el kernel y se carga en memoria.
- Se ejecuta initramfs para cargar módulos necesarios para ejecutar el kernel.
- Pasa el control a systemd

Cuarta Etapa: systemd

- Inicia y supervisa todo el sistema.
- Usa:
 - ▣ paralelización agresiva de servicios
 - ▣ activación D-Bus para iniciar servicios (sistema de comunicación entre aplicaciones de <> origen)
 - ▣ seguimiento de procesos con grupos de control (agrupa procesos para controlarlos)

UNITS

- systemd inicia y supervisa todo el sistema
- se basa en la noción de unidades (**units**)
- Las unidades están compuestas de un nombre (el nombre del demonio) y una extensión.
- La extensión indica el **tipo** de unidad que se trata. Cada unidad tiene su correspondiente archivo de configuración cuyo nombre es idéntico.
- Un ejemplo sería el servicio httpd.service cuyo archivo de configuración es httpd.service.
- Los archivos de unidades disponibles en nuestro sistema podemos encontrarlos en /usr/lib/systemd/system/ y /etc/systemd/system/

7 Tipos de Unidades

- **service**: Demonios que pueden ser iniciados, detenidos, reiniciados o recargados.
- **socket**: Esta unidad encapsula un socket en el sistema de archivos o en Internet. Cada unidad socket tiene una unidad de servicio correspondiente.
- **device**: Esta unidad encapsula un dispositivo en el árbol de dispositivos de Linux.
- **mount**: Esta unidad encapsula un punto de montaje en la jerarquía del sistema de archivos.
- **automount**: Encapsula un punto de montaje automático. Cada unidad automount tiene una unidad mount correspondiente, que se inicia al acceder al directorio de automontaje.
- **target**: Utilizada para la agrupación lógica de unidades. Referencia a otras unidades, que pueden ser controladas conjuntamente, un ejemplo sería multi-user.target, que básicamente desempeña el papel de nivel de ejecución 3 en el sistema clásico SysV.
- **snapshot**: Similar a las unidades target.

NOTA: Entonces los archivos de configuración tendrán los nombres: programa.service, socket.socket, dispositivo.device, puntodemontaje.mount, etc...

DEAMONS o servicios

“Un demonio, daemon o dæmon (Disk And Execution MONitor), es un tipo especial de programa que se ejecuta como un proceso **no interactivo**, es decir, que se ejecuta **en segundo plano** en vez de ser controlado directamente por el usuario. Este tipo de programas se ejecutan de forma continua (infinita); es decir, que aunque se intente cerrar o matar el **proceso**, este continuará en ejecución o se reiniciará automáticamente. Todo esto sin intervención de terceros y sin dependencia de consola alguna.”

Directorios importantes de systemd

/etc/systemd/system

- Contiene la lista de targets (conjunto de servicios). Cada target contiene symlinks a los script de inicio, agrupando en un target varios servicios.

/lib/systemd/system

- Contiene los scripts de inicio de los servicios.

Comando `systemctl`

- Herramienta para administración de unidades.
- Listar todas las unidades de servicio instaladas:

```
#systemctl list-unit-files
```

- Listas sólo unidades activas:

```
#systemctl list-units o #systemctl
```

NOTA de “atajos”:

Si no se especifica el sufijo, `systemctl` asumirá que es `.service`.

Por ejemplo, `netcfg` es igual a `netcfg.service`

En puntos de montaje: por ejemplo, si especifica `/home` será equivalente a `home.mount`.

Los dispositivos por ejemplo `/dev/sda2` es equivalente a `dev-sda2.device`.

Uso de systemctl y Compatibilidad con SysV

Acción	systemd	SysV
Arrancar un servicio	systemctl start foo	service foo start
Detener un servicio	systemctl stop foo	service foo stop
Reiniciar un servicio	systemctl restart foo	service foo restart
Recargar el archivo de configuración de un servicio (en systemd no todos los servicios lo soportan)	systemctl reload foo	service foo reload
Rearrancar un servicio que ya se encuentra en ejecución	systemctl condrestart foo	service foo condrestart
Mostrar el estado del servicio	systemctl status foo	service foo status
Activar un servicio para que sea ejecutado durante el arranque	systemctl enable foo	chkconfig foo on
Desactivar un servicio para que no sea ejecutado durante el arranque	systemctl disable foo	chkconfig foo off
Muestra el estado de un servicio durante el arranque	systemctl is-enable foo (1*)	chkconfig --list foo
Crear o modificar el archivo de configuración de un servicio	systemctl daemon-reload	chkconfig --add foo
Listar los modos de ejecución para los que un servicio está activado o desactivado	ls /etc/systemd/system/*.wants/foo.service	chkconfig

Comando systemctl

- Lista completa de targets

systemctl list-units --type=target

- Ubicar el target en el cual se encuentra un servicio

ls /etc/systemd/system/*.wants/ssh.service

- Ver servicios que se ejecutan en un target (conocer el nombre del target)

systemctl show -p "wants" multi-user.target

- Servicios cargados en detalle y si están activos

systemctl -t service list-units -all

¿Por qué automatizar tareas?

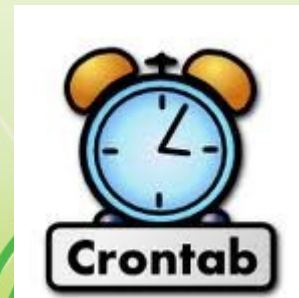
Simplifica y garantiza la ejecución de las mismas.

¿Cuándo?

- Actividades de gestión periódicas y control del sistema (por ejemplo, revisar diariamente el grado de ocupación de los discos, testear cada 30 minutos la ocupación del procesador y poder generar un aviso en caso de ocupación de más de un 70% ...).
- Borrar de forma periódica los ficheros temporales (por ejemplo, el directorio /tmp),
- Realizar copias de seguridad cada día de diversos directorios de datos.
- Realizar algún tipo de informe mensual sobre el funcionamiento del sistema (sobre todo en caso de servidores).
- Programar el apagado del equipo de forma automática, adecuándose al horario de trabajo.
- Programar tareas periódicas desde las necesidades del usuario como un sistema de recordatorios, o la descarga programada de información en horas de poco tráfico (para que sea más rápida).
- ¿ Otras?

Programación de tareas con cron

- **cron** es un administrador regular de procesos en segundo plano (demonio) que ejecuta procesos o scripts a intervalos regulares de tiempo (personalizable por el usuario).
- Utiliza la tabla de tareas `/etc/crontab` que es un archivo de texto donde se guarda una lista de tareas a ejecutar en un tiempo especificado por el usuario.
- Es usado normalmente para tareas administrativas como respaldos o actualizaciones automáticas.



Cron

- Iniciar el servicio

```
# systemctl start cron
```

- Detener el servicio

```
# systemctl stop cron
```

- Reiniciar el servicio

```
# systemctl restart cron
```

- Ver el estado del servicio

```
# systemctl status cron
```

cron

Archivos importantes implicados en el funcionamiento de servicio **cron**:

- El demonio de funcionamiento: **cron**
- La orden para la programación de tareas (disponible para los usuarios con suficientes privilegios): **crontab**
- El sistema de informes (logs) : **/var/log/cron**
- Para entender cómo se deben programar las tareas para que sean ejecutadas, es necesario entender el formato del archivo de configuración (/etc/crontab).
- Está estructurado por líneas, cada una de las cuales contiene una tarea programada, según el siguiente **formato**:

Formato para programar tareas

m h dom m on dow user com m and

m :Corresponde al minuto en que se va a ejecutar la tarea, el valor va de 0 a 59.

h :La hora exacta, los valores van de 0 a 23.

dom : Hace referencia al día del mes que se va a ejecutar la tarea.

m on : Significa el mes, puede ser numérico (1 a 12) o las 3 primeras letras del mes en inglés.

dow : Significa el día de la semana, puede ser numérico (0 a 7, donde 0 y 7 son domingo) o las 3 primeras letras del día en inglés.

user: Define el usuario que va a ejecutar la tarea, puede ser root u otro usuario siempre y cuando tenga permisos de ejecución sobre la tarea.

com m and : Refiere al comando o a la ruta absoluta de la tarea a ejecutar.

Ejemplo

55	23	*	*	0	root	/usr/local/sbin/copiasemanal.sh
Rango 0 - 59	Rango 0 - 23	Rango 1 - 31	Rango 1 - 12	Rango 0 - 6		Comando
					Usuario	
				Día de la semana	Lunes = 1, Martes = 2, Miércoles = 3 Jueves = 4, Viernes = 5, Sábado = 6, Domingo = 0	
		Mes	Enero = 1, Febrero = 2, Marzo = 3, Abril = 4, Mayo = 5, Junio = 6, Julio = 7 Agosto = 8, Septiembre = 9, Octubre = 10, Noviembre = 10, Diciembre = 12			
		Día del mes				
	Hora					
Minuto						

Ejecuta *copiasemanal.sh* cada domingo a las 23:55

Ejemplos de tareas cron

- Ejecutará el script actualizarDB.sh a las 10:15 a.m. todos los días

15 10 * * * usuario /home/usuario/scripts/actualizarDB.sh

- Ejecutará una actualización todos los domingos a las 22:00 p.m.

00 22 * * 0 root apt-get -y update

- El día 21 de noviembre a las 6:30 el usuario correrá el script backup.sh

30 6 21 11 * usuario /home/usuario/scripts/backup.sh

- A las 21:30 de la tarde todos los días de lunes a viernes

30 21 * * 1,2,3,4,5 root /opt/limpiarCache.sh

- A las 12:00 del día todos los días primero, quince y 28 de cada mes

00 12 1,15,28 * * root /opt/generarNoticias.sh

- Cada quince minutos de las 11:00a.m. a las 14:00p.m.

*/15 11-14 * * * root /opt/checkDNS.sh

- Cada 6 horas de los días 1 al 10 de cada mes y que el día sea miércoles

0 */6 1-10 * 3 root /opt/checkDNS.sh

Un * indicará inicio y fin del campo, es decir todo. Por ejemplo un * en el campo de minuto indicará todos los minutos.

APAGADO DEL SISTEMA

Finalizar sesión cuando termina de trabajar.

- Para que el sistema libere memoria que ocupaban sus aplicaciones, y otros usuarios puedan aprovecharla.
- Para evitar que otra persona utilice el sistema con su nombre de usuario y comprometer la integridad de los datos o la disponibilidad del sistema
- Comandos: **logout** **exit** **<ctrl> + <D>**
-

Detener el sistema: las aplicaciones guardan sus datos de estado, cierran sus archivos en disco y realizan todas las operaciones necesarias para continuar funcionando normalmente la próxima vez que se inicie el equipo.

- Comandos apagar: **poweroff shutdown halt**
- Comandos reiniciar: **reboot**

Ejercicio: buscar man shutdown