

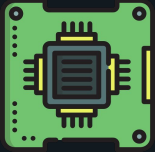
Gestor de Servicios y Sistemas GNU/Linux Systemd

Laboratorio de Sistemas Operativos II

1. Boot o arranque de una computadora



- Proceso que se inicia con el encendido de la computadora y termina cuando el sistema operativo toma el control total de la máquina y está lista para ser utilizada por el usuario.
- Existen 2 tecnologías:
 - **BIOS** Legacy (Basic Input Output System)
 - **UEFI** (Interfaz de Firmware Extensible Unificada)



Firmware del sistema

SW escrito en un lenguaje de bajo nivel, grabado en un chip de memoria no volátil (Flash o ROM) en la motherboard. Actúa como el primer puente de comunicación entre el HW y el SO

01

Forma BIOS Legacy (BIOS + MBR)

- Sencillo y funciona en un modo de compatibilidad antiguo (16 bits)
- Para arrancar el sistema operativo, busca el 1º sector físico del disco duro MBR (Master Boot Record, Sector 0 de 512 bytes).
- Allí ejecuta un código para invocar al cargador de arranque (como GRUB).
- No soporta discos de más de 2 TB ni más de 4 particiones primarias.

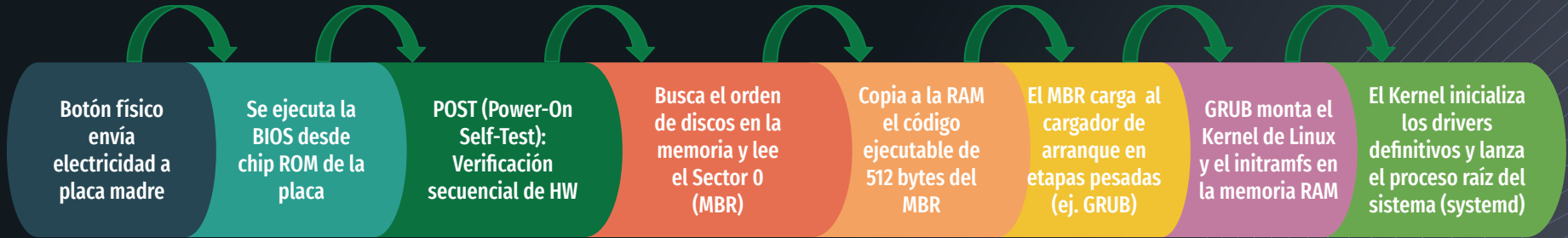
02

Forma Moderna (UEFI + GPT)

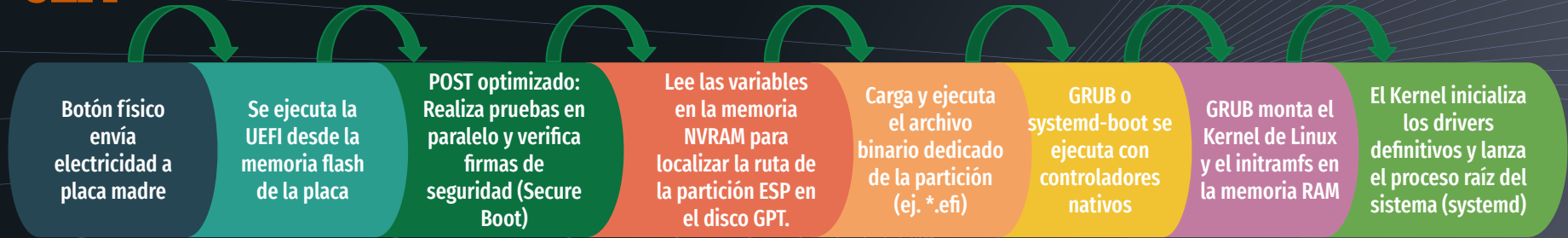
- Reemplazó a la BIOS.
- UEFI es mini SO: corre en 32 o 64 bits,
- Soporta interfaces gráficas, ratón y conexiones de red antes de arrancar el sistema operativo principal.
- No busca un sector oculto en el disco, entiende sistemas de archivos.
- Lee la tabla de particiones GPT y busca una partición especial formateada en FAT32 llamada ESP (EFI System Partition), donde ejecuta directamente un archivo binario (con extensión .efi)..
- Soporta discos gigantescos, inicio seguro (Secure Boot) contra malware y es más rápido.

Pasos de arranque de acuerdo al firmware

BIOS Legacy



UEFI



- **BIOS** (Basic Input Output System) y **UEFI** (Unified Extensible Firmware Interface) son estándares de interfaz de firmware.
- **MBR** (Master Boot Record) y **GPT** (GUID Partition Table) son estándares para la organización de particiones en un disco duro

Cómo saber el tipo de arranque

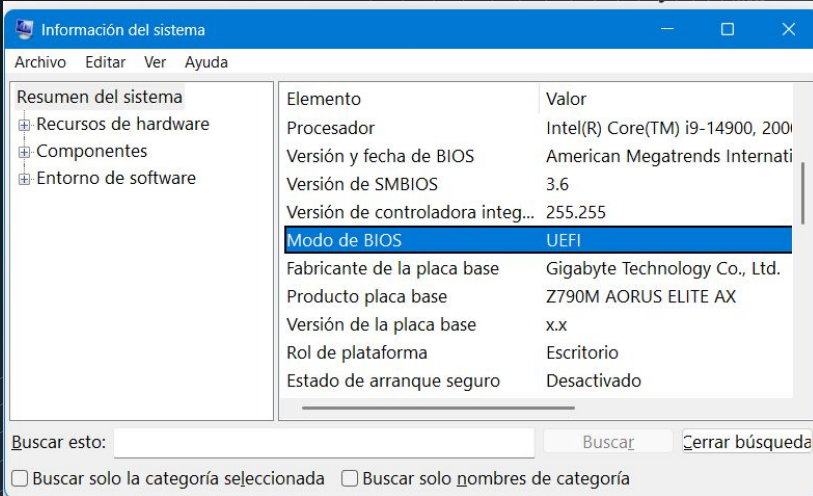
DEBIAN

➤ `#ls /sys/firmware/efi`

Si el directorio `/sys/firmware/efi` existe, significa que el sistema se ha iniciado en modo UEFI. Si el directorio no existe, entonces el sistema inició en modo BIOS.

WINDOWS

➤ `<Windows> +< R>`
➤ `msinfo32`



The screenshot shows the 'Información del sistema' (System Information) window in Windows. The 'Modo de BIOS' (BIOS Mode) is highlighted in blue, showing 'UEFI'. Other system details are visible in the table below.

Elemento	Valor
Procesador	Intel(R) Core(TM) i9-14900, 200
Versión y fecha de BIOS	American Megatrends Internati
Versión de SMBIOS	3.6
Versión de controladora integ...	255.255
Modo de BIOS	UEFI
Fabricante de la placa base	Gigabyte Technology Co., Ltd.
Producto placa base	Z790M AORUS ELITE AX
Versión de la placa base	x.x
Rol de plataforma	Escritorio
Estado de arranque seguro	Desactivado

Inicio en Debian con **systemd**

- “ ● Se ejecuta el primer proceso (systemd)
- El kernel inicia systemd como el proceso con PID 1.
- A partir de este paso, **systemd controla** el arranque del sistema.
- Lee sus archivos de configuración y comienza a iniciar **unidades** (servicios, montajes, temporizadores, etc.) en paralelo y en orden, según sus dependencias.
- Realiza el Montaje del sistema de archivos raíz y otros dispositivos
- Inicia servicios y programas esenciales y críticos como red, log, interfaz gráfica (si aplica), etc.
- Carga del entorno de usuario
- Sistema listo para usar

2. Systemd

Un **gestor de sistema y servicios** es una aplicación o conjunto de componentes de software que se encarga de iniciar, detener, administrar, supervisar y coordinar los recursos, procesos, servicios y dispositivos del sistema operativo, asegurando la correcta operación y disponibilidad del sistema completo

The logo for Systemd, featuring a green circle and a green triangle pointing left, enclosed in white square brackets, followed by the text 'systemd' in white lowercase letters.

[● ◀] systemd

Systemd



- Es el primer proceso que se ejecuta en el espacio de usuario, por lo tanto, también es el **proceso padre** de todos los procesos hijos en el espacio de usuario.
- Es el gestor de sistema y servicios por defecto a partir de Debian 8 Jessie. Versiones anteriores usan SysV.
- Fue diseñado por Lennart Poettering para el núcleo de Linux, es SW libre y de código abierto bajo la GNU General Public License
- Uno de los principales objetivos de systemd es unificar configuraciones básicas de Linux y los comportamientos de servicios en todas las distribuciones.

Características de Systemd



Unidades de sistema:

Systemd utiliza **unidades** para describir el comportamiento del sistema y los servicios, lo que permite un control granular sobre el inicio y el estado de los servicios.

Paralelismo:

Systemd inicia los servicios de forma paralela, lo que reduce significativamente el tiempo de arranque del sistema.

Dependencias transaccionales:

La gestión de dependencias entre servicios garantiza que se inicien en el orden correcto, evitando errores y problemas.

Soporte para instantáneas y restauración:

Permite crear instantáneas del sistema y restaurar a un estado anterior, lo que facilita la gestión y el mantenimiento.

Gestión de servicios:

Systemd ofrece una herramienta de línea de comandos (systemctl) para gestionar todas las unidades y sus configuraciones.

Centralización de registros:

Systemd utiliza un registro centralizado (**journal**) para registrar los eventos del sistema, lo que facilita la resolución de problemas.

Unidades en **Systemd**



- Una unidad es un archivo que describe un servicio, un dispositivo, un punto de montaje, una tarea que systemd debe gestionar. Las dos rutas en Debian para las unidades son:
 - **/lib/systemd/system/**: Archivos distribuidos por el sistema operativo o los paquetes (¡No se tocan!).
 - **/etc/systemd/system/**: Archivos creados o modificados por el administrador (Tienen prioridad).
- Cada unidad tiene un nombre (nombre del demonio) y una configuración específica con una extensión según el tipo de unidad.
- La extensión indica el tipo de unidad que se trata.

Algunas unidades ...



Tipo	Descripción
Servicio (.service)	Gestiona programas o procesos en ejecución.
Montaje (.mount)	Puntos de montaje de archivos y dispositivos.
Socket (.socket)	Gestiona la conexión de comunicación entre servicios.
Target (.target)	Agrupar varias unidades para gestionar grupos.
Device (.device)	Representa dispositivos hardware detectados por el sistema, como discos, interfaces de red, periféricos USB, etc.

- Cuando inicias tu computadora, systemd carga, el servicio de red (network.service).
- Cuando conectas un disco USB, systemd puede crear una unidad .device para ese dispositivo (usb0.device).

Tipo de Unidad	Descripción	Función Principal / Caso de Uso Práctico
.service	Demonios y servicios tradicionales.	Controla procesos en segundo plano. Se encarga de iniciar, detener y monitorear aplicaciones activas (ej. <code>ssh.service</code> , <code>apache2.service</code>).
.target	Grupos lógicos de sincronización.	Define estados del sistema (antiguos runlevels). No ejecuta código directamente, sino que agrupa otras unidades para alcanzar una "meta" (ej. <code>graphical.target</code> para entorno gráfico).
.timer	Temporizadores del sistema.	Automatización basada en tiempo. Despierta a un <code>.service</code> asociado según un calendario o un evento relativo (ej. <i>"ejecutar 5 min después de arrancar"</i>). Reemplazo moderno de <code>cron</code> .
.socket	Sockets de red o comunicación interna (IPC).	Activación bajo demanda. Escucha puertos de red o archivos de socket. Si llega tráfico, <code>systemd</code> arranca el servicio correspondiente al vuelo para ahorrar memoria RAM.
.mount	Puntos de montaje de archivos.	Gestiona el almacenamiento del sistema. Traduce las entradas de <code>/etc/fstab</code> en objetos de <code>systemd</code> , permitiendo que un servicio espere a que un disco esté montado antes de iniciar.
.automount	Montaje automático retardado.	Montaje bajo demanda. Mantiene el punto de acceso listo, pero no monta físicamente el disco o la carpeta de red hasta que un proceso intenta leer o escribir en esa ruta exacta.
.path	Monitoreo de rutas y archivos.	Activación por cambios en archivos. Vigila un archivo o directorio. Si el archivo se modifica, se crea o se borra, dispara automáticamente un servicio (ej. procesar un PDF apenas cae en una carpeta).
.device	Dispositivos de hardware expuestos.	Sincronización con el hardware. Muestra dispositivos controlados por <code>udev</code> como unidades. Útil para que un servicio no arranque si un componente físico (como una tarjeta de red específica) no está listo.
.slice	Contenedores de recursos (Cgroups).	Gestión de rendimiento y límites. Agrupa procesos en "rebanadas" para limitarles jerárquicamente el uso máximo de CPU, memoria RAM y lectura/escritura de disco.
.scope	Gestión de procesos externos.	Monitoreo de procesos ajenos a systemd. Agrupa procesos que fueron creados por vías externas (como una sesión de usuario en un shell interactivo) para poder medir y gestionar sus recursos.
.swap	Espacio de memoria de intercambio.	Gestión de la memoria virtual. Controla la activación y desactivación de las particiones o archivos de intercambio (<i>Swap</i>) del sistema operativo.

Ejemplos de Unidades de Systemd



- **Service** (.service) : Controlan los demonios y procesos del sistema.
 - **cron.service**: El servicio que ejecuta tareas programadas en segundo plano.
 - **networking.service**: El servicio que levanta y configura las interfaces de red.
- **Device** (.device): Representan los dispositivos de hardware reconocidos por el kernel
 - **sys-subsystem-net-devices-eth0.device**: La tarjeta de red Ethernet física.
 - **dev-sda1.device**: La primera partición del primer disco rígido principal.
- **Target** (.target) : Agrupan otras unidades para definir el nivel de ejecución del sistema.
 - **multi-user.target**: Sistema completo en modo consola con red y múltiples usuarios.
 - **reboot.target**: Define las prioridades y servicios que deben cerrar al reiniciar el sistema.
- **Mount** (.mount): Controlan los puntos de montaje de los filesystem (discos y particiones).
 - **-.mount**: Representa la partición raíz (/) del sistema.
 - **home.mount**: El punto de montaje para el directorio de los usuarios (/home).

Unidades de Servicio: Demonios [● ◀]

- El SO tiene procesos independientes del usuario; que no dependen de la conexión o desconexión de ningún usuario. Esos son los Demonios (Daemons).
- Un demonio es un programa que se ejecuta en 2º plano, de forma continua y sin interacción directa del usuario, para realizar tareas del sistema o prestar servicios.
- **Systemd** es el encargado de gestionar los demonios del sistema:
 - Control total: inicia, detiene, reinicia y supervisa los demonios.
 - Archivos de configuración: utiliza archivos llamados unidades de servicio (.service) para saber cómo y cuándo arrancar cada demonio.
 - Paralelismo: puede iniciar varios demonios al mismo tiempo.
 - Gestión de dependencias: resuelve el orden de ejecución automáticamente.
 - Monitoreo: reinicia de forma automática un demonio con error.
- Herramienta de control: El comando `systemctl` es la interfaz para interactuar con estos demonios

Servicio: **cron.service**



- **#cat /lib/systemd/system/cron.service**

```
[Unit]
Description=Regular background program
processing daemon
Documentation=man:cron(8)
After=remote-fs.target
nss-lookup.target
```

```
[Service]
EnvironmentFile=-/etc/default/cron
ExecStart=/usr/sbin/cron -f
$EXTRA_OPTS
IgnoreSIGPIPE=false
KillMode=process
Restart=on-failure
```

```
[Install]
WantedBy=multi-user.target
```

Le dice a systemd que espere a que los discos remotos y la resolución de nombres de red estén listos antes de arrancar cron.

Carga opciones de configuración adicionales desde ese archivo de texto (el signo - significa que no pasa nada si el archivo no existe).

Es el comando real que ejecuta el demonio (/usr/sbin/cron). El parámetro -f le ordena ejecutarse en primer plano para que systemd pueda monitorearlo directamente.

Si el demonio de cron se cierra de forma inesperada por un error, systemd lo vuelve a iniciar automáticamente.

3.

Herramientas para gestionar unidades de `systemd`

Comandos `systemctl` y `journalctl`

Listar Unidades activas

```
#systemctl list-units
```

UNIT	LOAD	ACTIVE	SUB	DESCRIPTION
accounts-daemon.service	loaded	active	running	Accounts Service
acpid.service	loaded	active	running	ACPI event daemon
alsa-restore.service	loaded	active	exited	Save/Restore Sound Card State
apparmor.service	loaded	active	exited	AppArmor initialization

- UNIT: nombre de la unidad.
- LOAD: cargada o no en la memoria.
- ACTIVE: El estado de activación del archivo de la unidad de alto nivel (activo, recargando, inactivo, fallido, activando, desactivando). Generalización de SUB.
- SUB: El estado de la subunidad o proceso asociado (running, failed, waiting, etc.).
- DESCRIPTION: una descripción corta de qué es y qué hace la unidad.

Opciones para **listar** unidades

- `#systemctl list-units --all` • unidades cargadas, incluidas las que están inactivas
- `#systemctl list-units --all --state=inactive` • sólo las unidades inactivas
- `#systemctl list-units --type=service` • sólo las unidades de tipo servicio
- `#ls /etc/systemd/system/*.wants/ssh.service` • Target en el que está un servicio
- `#systemctl list-dependencies nombre-unidad` • Dependencias de una unidad

Gestionar **servicios** durante la sesión actual

```
#systemctl status nombre.service
```

- estado e información de un servicio en ejecución

```
#systemctl start nombre.service
```

- Iniciar un servicio detenido

```
#systemctl stop nombre.service
```

- Detener un servicio en ejecución.

```
#systemctl restart nombre.service
```

- Reiniciar un servicio en ejecución.

```
#systemctl reload nombre.service
```

- Recarga la conf. de un servicio sin reiniciarlo. No apto en todos.

Habilitar y deshabilitar **servicios** en el arranque

`#systemctl enable nombre.service`

`#systemctl disable nombre.service`

`#systemctl is-active nombre.service`

`#systemctl is-enabled nombre.service`

`#systemctl is-failed nombre.service`

- Iniciar automáticamente un servicio en el arranque
- Impedir que un servicio se inicie automáticamente
- Comprobar si una unidad está activa actualmente (ejecutándose)
- Ver si la unidad está habilitada,
- Ver si la unidad está en estado fallido (error al iniciar)

Devolverá active si se está ejecutando adecuadamente o failed si se ha producido un error. Si la unidad se detuvo intencionadamente, puede devolver unknown o inactive

Journal

- Journal es el sistema de registro de systemd , implementado en la unidad systemd-journald.service.
- El archivo de configuración de systemd-journald.service ,se encuentra en /etc/systemd/journald.conf .
- Este archivo contiene las distintas opciones para configurar entre otras cosas , el modo, tipo y tamaño de los logs de registro.

Comando **journalctl**

Comando	Descripción
<code>#journalctl</code>	Ver los logs del demonio journal
<code>#journalctl -r</code>	Mostrar la salida del comando en orden invertido
<code>#journalctl -f</code>	Ver el journal del ordenador en tiempo real, y las nuevas líneas las iremos viendo entrar a las salida del comando.
<code>#journalctl -n12</code>	Permite limitar, en este caso particular, la salida a solamente las últimas 12 líneas
<code>#journalctl -k</code>	Mostrar los registros del kernel

Targets la próxima semana



¡Gracias!

¿Preguntas?