

systemd

ENRIQUE CALDERÓN, FRANCISCO GALINDO
Estudiante de Ingeniería en Computación

*Universidad Nacional Autónoma de México
Facultad de Ingeniería*

Curso de SysAdmin, 2025-1

Información del tema

Tiempo estimado

Aproximadamente 60 minutos de clase.

Objetivos

Que los alumnos comprendan como se administran los programas en un sistema Debian GNU/Linux, tanto usando el administrador de paquetes apt como mediante el uso de código fuente.

Que los alumnos sepan qué es un *init system*, qué es *systemd*, así como realizar un manejo básico de las distintas *units* que controla el sistema.

Sistemas de inicialización (*init systems*)

En sistemas tipo *Unix*, el *init system* es el primer programa que se ejecuta en el sistema operativo después del kernel en sí. Es un *daemon*¹ que se ejecuta desde que la computadora arranca hasta que se apaga.

Se encarga de iniciar, ya sea directa o indirectamente, todos los procesos del sistema operativo.

¹Un programa que se ejecuta en segundo plano, en lugar de estar controlado directamente por un usuario

systemd

systemd inició como un sistema de inicialización alternativo para sistemas Linux, pero con el tiempo ha ido evolucionando y retomando más responsabilidades.

systemd no es un sólo programa, sino que se compone de varios programas que realizan tareas más específicas.

Actualmente, *systemd* suele encargarse de las tareas de *init*, manejo de redes y *firewalls*, *bootloaders*, etc.

Unidades

Aquellas entidades que son administradas por *systemd* se les llama unidades (*unit*). Existen los siguientes tipos de *units* dependiendo de su extensión de archivo:

.service	.socket
.device	.device
.mount	.automount
.swap	.target
.path	.timer
.snapshot	.slice
.scope	

Revisando el estado de una unidad

Instala el servidor web apache2 para probar varios de los comandos que vendrán después:

```
sudo apt install apache2
```

El paquete apache2 incluye una unidad para que ejecuta al servidor de apache en segundo plano. Revisa el estado de apache2

```
$ systemctl status apache2
```

```
apache2.service - The Apache HTTP Server
```

```
Loaded: loaded (/lib/systemd/system/apache2.service; .
```

```
Active: active (running) since ...
```

```
...
```

Otros comandos sobre control de unidades

`systemctl start unidad` *# Inicia la unidad*

`systemctl stop unidad` *# Detiene la unidad*

`systemctl restart unidad` *# Detiene e inicia la unidad*

Generalmente recarga ciertas configuraciones

sin detener la unidad por completo

`systemctl reload unidad`

Hará que la unidad inicie al bootear sistema

`systemctl enable unidad`

Ya no iniciará al encender la computadora

`systemctl disable unidad`

Otros comandos útiles

Muestra las unidades

```
systemctl list-units
```

Muestra los timers

```
systemctl list-timers
```

Para ver los logs de las unidades

```
journalctl
```


¿Dónde están las unidades?

Hay diferentes directorios con archivos que describen las diferentes unidades de las que *systemd* estará al tanto, un par de estas son (de menor a mayor prioridad):

- ▶ `/lib/systemd/system`: Aquí se instalan de manera estándar las unidades instaladas por el gestor de paquetes.
- ▶ `/etc/systemd/system`: Se guardan unidades personalizadas

La idea de la separación es que alguna actualización de paquetes no sobrescriba una configuración personalizada ya existente.

Sintaxis de un *unit file*

[Sección]

Directiva1=valor

Directiva2=valor

Hay varias secciones, como [Unit], [Service], [Timer] e [Install], las opciones posibles para cada sección se encuentran en su página de manual correspondiente (systemd.unit, systemd.service, systemd.timer. . .).

Creando un servicio sencillo

Ejemplo:

```
[Unit]
```

```
Description=Mi propio server mío de mí
```

```
After=network.target
```

```
StartLimitIntervalSec=0
```

```
[Service]
```

```
Type=simple
```

```
Restart=always
```

```
RestartSec=1
```

```
User=debian
```

```
ExecStart=/home/debian/bin/miServer
```

```
[Install]
```

```
WantedBy=multi-user.target
```