

Systemd en Linux: administrar servicios, niveles de ejecución y registros

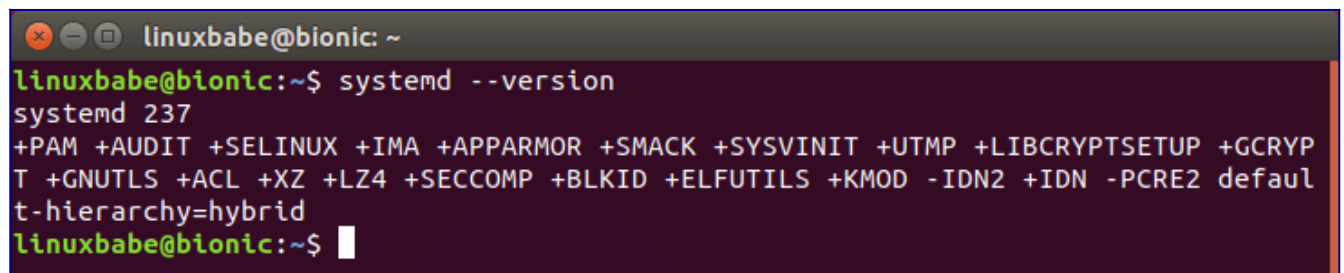
Systemd no es sólo un administrador de servicios sino también un administrador de sistemas para Linux. Está diseñado para ser compatible con versiones anteriores de los scripts de inicio de SysV y lo utilizan muchas distribuciones populares de Linux como Debian 8+, Ubuntu 15.04+, Fedora, Redhat 7+ /CentOS 7+, Arch Linux, OpenSUSE. Por lo que será de gran ayuda saber cómo utilizar systemd.

Este tutorial le muestra algunos comandos básicos que serán útiles para administrar servicios, niveles de ejecución y registros. Primero, es posible que desee conocer la versión de systemd en su distribución de Linux, ejecute

```
systemd --version
```

o

```
systemctl --version
```

A terminal window with a dark background and light-colored text. The prompt is 'linuxbabe@bionic: ~'. The command 'systemd --version' has been entered and executed. The output shows 'systemd 237' followed by a list of features: '+PAM +AUDIT +SELINUX +IMA +APPARMOR +SMACK +SYSVINIT +UTMP +LIBCRYPTSETUP +GCRYPT +GNUTLS +ACL +XZ +LZ4 +SECCOMP +BLKID +ELFUTILS +KMOD -IDN2 +IDN -PCRE2 default-hierarchy=hybrid'. The prompt is now 'linuxbabe@bionic:~\$' with a cursor.

```
linuxbabe@bionic: ~  
linuxbabe@bionic:~$ systemd --version  
systemd 237  
+PAM +AUDIT +SELINUX +IMA +APPARMOR +SMACK +SYSVINIT +UTMP +LIBCRYPTSETUP +GCRYPT  
+GNUTLS +ACL +XZ +LZ4 +SECCOMP +BLKID +ELFUTILS +KMOD -IDN2 +IDN -PCRE2 default  
-hierarchy=hybrid  
linuxbabe@bionic:~$
```

Servicios de gestión

Systemd puede ayudarle a iniciar, detener y reiniciar servicios y también le permite iniciar servicios automáticamente en el momento del arranque.

Para enumerar los servicios systemd activos, ejecute

```
systemctl list-units --type service
```

Verifique el estado de un servicio específico, por ejemplo, el demonio SSH.

```
systemctl status ssh
```

Para iniciar un servicio, utilice el subcomando `start`:

```
sudo systemctl start ssh
```

Comprueba si está activo (en ejecución)

```
systemctl is-active ssh
```

Habilitar un servicio para que se inicie automáticamente en el momento del arranque

```
sudo systemctl enable ssh
```

Comprueba si está habilitado

```
systemctl is-enabled ssh
```

Evitar que un servicio se inicie en el momento del arranque

```
sudo systemctl disable ssh
```

detener un servicio

```
sudo systemctl stop ssh
```

¿Deshabilitaste un servicio systemd, pero de alguna manera comenzó de nuevo? Para evitar que un servicio se inicie manualmente, utilice el subcomando `mask`.

```
sudo systemctl mask ssh
```

Los servicios enmascarados no se pueden iniciar con el comando `systemctl start` hasta que se desenmascaren.

```
sudo systemctl unmask ssh
```

A veces, puede encontrar que se está ejecutando un proceso desconocido con el comando `htop`.

Quiere detenerlo, pero no sabe a qué servicio systemd pertenece. En este caso, puede ejecutar el siguiente comando

```
systemctl status
```

Este comando le dará una lista de unidades de servicio systemd activas y los comandos utilizados para iniciarlas.

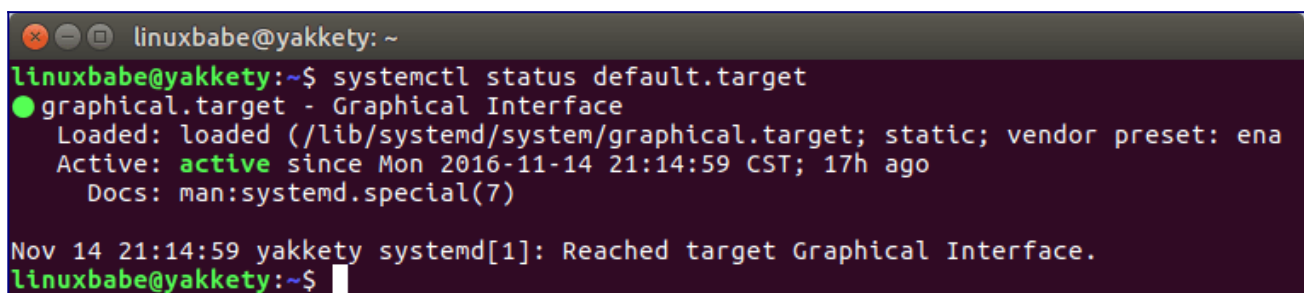
Gestión de niveles de ejecución

El concepto de niveles de ejecución se reemplaza por objetivos en systemd. `multi-user.target` es equivalente al nivel de ejecución 3 y `graphical.target` es equivalente al nivel de ejecución 5. Aún puedes usar el comando `runlevel` para mostrar la ejecución actual nivel.

```
runlevel
```

Utilice el siguiente comando para comprobar el estado del objetivo predeterminado, que suele ser el objetivo gráfico (nivel de ejecución 5).

```
systemctl status default.target
```



```
linuxbabe@yakkety: ~  
linuxbabe@yakkety:~$ systemctl status default.target  
● graphical.target - Graphical Interface  
   Loaded: loaded (/lib/systemd/system/graphical.target; static; vendor preset: ena  
   Active: active since Mon 2016-11-14 21:14:59 CST; 17h ago  
     Docs: man:systemd.special(7)  
  
Nov 14 21:14:59 yakkety systemd[1]: Reached target Graphical Interface.  
linuxbabe@yakkety:~$
```

Utilice el comando `systemctl isolate` para cambiar el objetivo. Por ejemplo, cambie a `multi-user.target` (nivel de ejecución 3):

```
sudo systemctl isolate multi-user.target
```

Y para volver al objetivo gráfico (nivel de ejecución 5):

```
sudo systemctl isolate graphical.target
```

Para establecer `multi-user.target` como destino predeterminado, utilice el siguiente comando:

```
sudo systemctl set-default multi-user.target
```

Este comando crea un enlace simbólico.

```
linuxbabe@yakkety: ~  
linuxbabe@yakkety:~$ sudo systemctl set-default multi-user.target  
Created symlink from /etc/systemd/system/default.target to /lib/systemd/system/multi-user.target.  
linuxbabe@yakkety:~$ ls /etc/systemd/system/default.target -lh  
lrwxrwxrwx 1 root root 37 Nov 15 01:17 /etc/systemd/system/default.target -> /lib/systemd/system/multi-user.target  
linuxbabe@yakkety:~$
```

Si reinicia ahora, se le dirigirá al destino multiusuario.

```
linuxbabe@yakkety: ~  
linuxbabe@yakkety:~$ systemctl status default.target  
● multi-user.target - Multi-User System  
   Loaded: loaded (/lib/systemd/system/multi-user.target; enabled; vendor preset: e  
   Active: active since Tue 2016-11-15 01:23:05 EST; 2min 58s ago  
     Docs: man:systemd.special(7)  
linuxbabe@yakkety:~$
```

Para enumerar objetivos activos, ejecute

```
systemctl list-units --type target
```

Administrar registros

El paquete de software `systemd` incluye una utilidad `journalctl` que se puede utilizar para administrar registros en Linux. Linux tradicionalmente guarda registros en el directorio `/var/log/`. En una distribución de Linux con `systemd`, es posible que no pueda encontrar registros como registros de correo postfix (`/var/log/mail.log`) en ese directorio.

Puede ver registros recientes con el siguiente comando. Puede presionar la tecla `F` para saltar a la página siguiente y presionar la tecla `B` para volver a la página anterior. Presione `Q` para salir.

```
sudo journalctl
```

Utilice `grep` para buscar registros relacionados con su término de búsqueda

```
sudo journalctl | grep <search term>
```

Ver registros desde el inicio actual

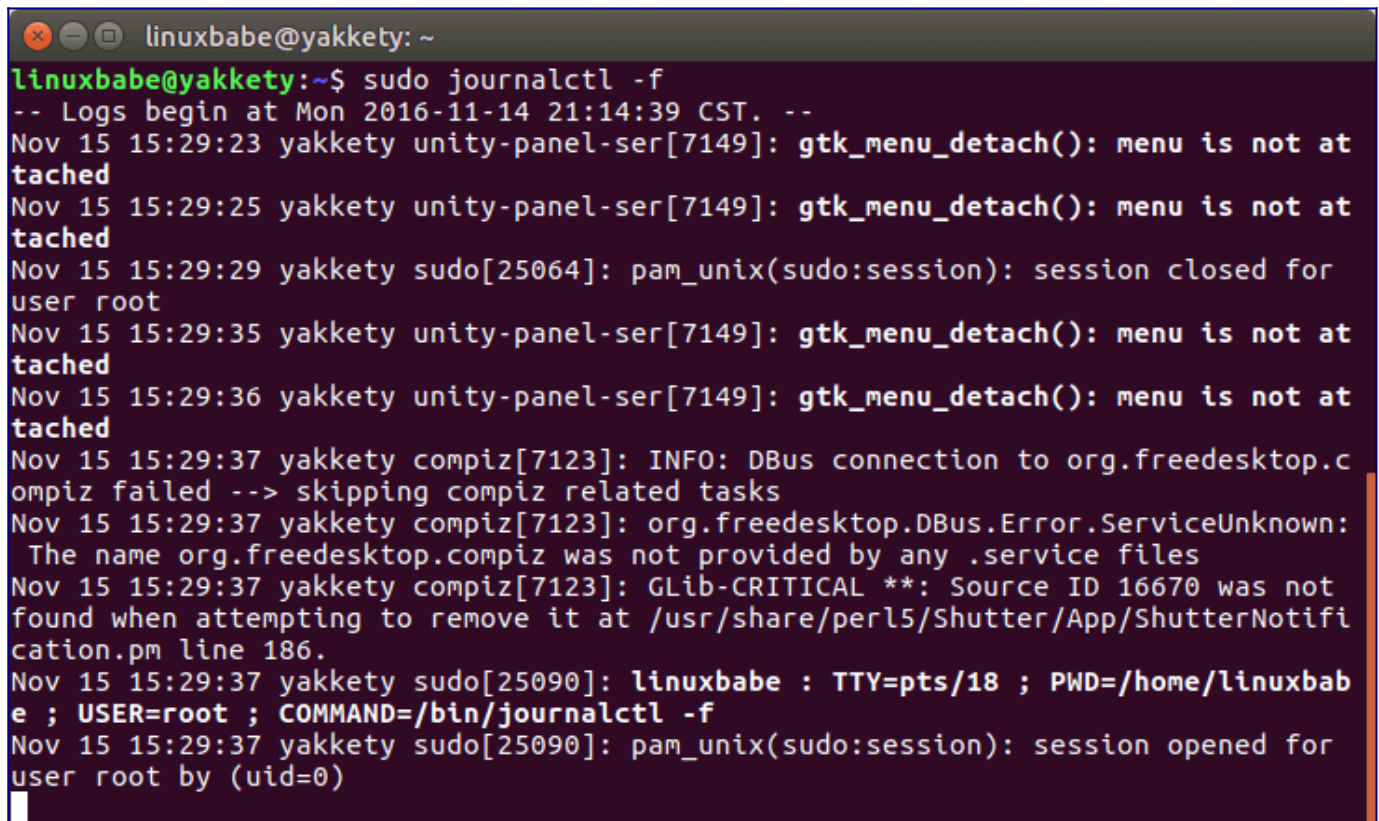
```
sudo journalctl -b
```

Ver registros desde el inicio anterior

```
sudo journalctl -b -1
```

Aquí hay un pequeño truco para mostrar los registros generados en tiempo real.

```
sudo journalctl -f
```

A terminal window titled 'linuxbabe@yakkety: ~' showing the output of 'sudo journalctl -f'. The output displays system logs in real-time, including messages from 'yakkety unity-panel-ser[7149]', 'sudo[25064]', and 'compiz[7123]'. It shows a session being closed for user root and then opened for user root by (uid=0). The terminal has a dark background with light-colored text.

```
linuxbabe@yakkety:~$ sudo journalctl -f
-- Logs begin at Mon 2016-11-14 21:14:39 CST. --
Nov 15 15:29:23 yakkety unity-panel-ser[7149]: gtk_menu_detach(): menu is not at
tached
Nov 15 15:29:25 yakkety unity-panel-ser[7149]: gtk_menu_detach(): menu is not at
tached
Nov 15 15:29:29 yakkety sudo[25064]: pam_unix(sudo:session): session closed for
user root
Nov 15 15:29:35 yakkety unity-panel-ser[7149]: gtk_menu_detach(): menu is not at
tached
Nov 15 15:29:36 yakkety unity-panel-ser[7149]: gtk_menu_detach(): menu is not at
tached
Nov 15 15:29:37 yakkety compiz[7123]: INFO: DBus connection to org.freedesktop.c
ompiz failed --> skipping compiz related tasks
Nov 15 15:29:37 yakkety compiz[7123]: org.freedesktop.DBus.Error.ServiceUnknown:
The name org.freedesktop.compiz was not provided by any .service files
Nov 15 15:29:37 yakkety compiz[7123]: GLib-CRITICAL **: Source ID 16670 was not
found when attempting to remove it at /usr/share/perl5/Shutter/App/ShutterNotifi
cation.pm line 186.
Nov 15 15:29:37 yakkety sudo[25090]: linuxbabe : TTY=pts/18 ; PWD=/home/linuxbab
e ; USER=root ; COMMAND=/bin/journalctl -f
Nov 15 15:29:37 yakkety sudo[25090]: pam_unix(sudo:session): session opened for
user root by (uid=0)
```

Muestra registros de una unidad, por ejemplo, la unidad de servicio SSH.

```
sudo journalctl -u ssh
```

Para ver los registros de Postfix, ejecute

```
sudo journalctl -u postfix
```

Es posible que algunos paquetes aún estén configurados para guardar registros en archivos simples, como los archivos de registro de Postfix en Ubuntu (`/var/log/mail.log`), por lo que puede usar un editor de texto de línea de comandos para leer detenidamente los registros. A veces, un servicio `systemd` puede dejar de generar mensajes de registro; puede intentar reiniciar `rsyslog` para resolver este problema.

```
sudo systemctl restart rsyslog
```

Servicios en conflicto

Cuando dos servicios no deberían ejecutarse al mismo tiempo, puede utilizar el parámetro `Conflicts`. Por ejemplo, siempre uso una VPN en mi computadora de escritorio con Ubuntu. Es una VPN autohospedada que se ejecuta en un VPS. Luego, a veces uso Transmission para descargar torrents. Cuando inicio Transmission, debo cerrar la conexión VPN, porque Vultr cerrará tu cuenta si usas sus servidores para descargar torrents (me refiero a descargar materiales que infringen los derechos de autor a través de BitTorrent).

Puedo editar el `transmission-daemon.service`.

```
sudo nano /lib/systemd/system/transmission-daemon.service
```

Agregue la siguiente línea en la sección `[Unit]`.

```
Conflicts=openconnect.service
```

```
[Unit]
Description=Transmission BitTorrent Daemon
After=network.target
Conflicts=openconnect.service

[Service]
User=debian-transmission
Type=notify
ExecStart=/usr/bin/transmission-daemon -f --log-error
ExecStop=/bin/kill -s STOP $MAINPID
ExecReload=/bin/kill -s HUP $MAINPID
NoNewPrivileges=true

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

Guarde y cierre el archivo. Luego recarga systemd para que los cambios surtan efecto.

```
sudo systemctl daemon-reload
```

`openconnect.service` es el servicio que inicia la conexión VPN en mi computadora. Ahora, si inicio `transmission-daemon.service`, `openconnect.service` se cerrará automáticamente y viceversa.

Quiere

Si desea que el servicio B se inicie cuando se inicie el servicio A, puede utilizar el parámetro `Wants` para que esto suceda. En la sección `[Unit]` de un archivo `systemd.service`, agregue una línea como esta:

```
[Unit]
Wants = b.service
```

También puede utilizar `Before` = o `After` = para especificar el orden.

```
[Unit]
```

```
Wants = b.service
Before = b.service
```

A veces, es posible que desee que el servicio B espere a que se complete el servicio A antes de iniciarse. Luego puede agregar las siguientes líneas en la sección `[Service]` del archivo `b.service`.

```
[Service]
ExecStartPre=/bin/sleep 20
```

Esto le dice a `b.service` que espere 20 segundos.

Recuerde recargar `systemd` para que los cambios surtan efecto.

```
sudo systemctl daemon-reload
```

Ejecute los scripts `/etc/rc.local`

Si está ejecutando una distribución de Linux que usa `Systemd`, es posible que sus comandos en el archivo `/etc/rc.local` no se ejecuten en el momento del inicio del sistema. El siguiente tutorial explica cómo habilitar el script `/etc/rc.local` para que se ejecute al iniciar el sistema.

- Cómo habilitar `/etc/rc.local` con `Systemd`

Contabilidad

`Systemd` le permite comprobar cuánta CPU y memoria utiliza un servicio y cuántas tareas ejecuta el servicio. Si está utilizando una distribución de Linux reciente como **Ubuntu 20.04**, **Debian 10**, **RHEL 8**, la contabilidad está habilitada de forma predeterminada. En distribuciones de Linux antiguas como Ubuntu 18.04, es necesario habilitar la contabilidad manualmente. Supongamos que desea habilitar la contabilidad de Nginx en Ubuntu 18.04, luego siga las instrucciones a continuación.

Primero, creamos un directorio para almacenar configuraciones `systemd` personalizadas para Nginx.

```
sudo mkdir -p /etc/systemd/system/nginx.service.d/
```

Cree el archivo de configuración personalizado en este directorio. La extensión del nombre del archivo debe ser `.conf`.

```
sudo nano /etc/systemd/system/nginx.service.d/custom.conf
```

Luego agregue las siguientes líneas en este archivo.

```
[Service]
CPUAccounting=true
MemoryAccounting=true
TasksAccounting=true
```

Guarde y cierre el archivo. Luego recarga `systemd`.

```
sudo systemctl daemon-reload
```

No es necesario reiniciar Nginx. Ahora puedes verificar el estado de Nginx con:

```
systemctl status nginx
```

```
linuxbabe@nyc3:~$ systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Drop-In: /etc/systemd/system/nginx.service.d
            └─ custom.conf
   Active: active (running) since Wed 2021-05-12 11:06:06 HKT; 2 weeks 5 days ago
     Docs: man:nginx(8)
  Main PID: 1952 (nginx)
    Tasks: 7 (limit: 2360)
   Memory: 92.2M
      CPU: 51min 4.873s
   CGroup: /system.slice/nginx.service
            └─ 1952 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
               └─ 13046 nginx: worker process
                  13047 nginx: worker process
                  13048 nginx: cache manager process
```

Configuración personalizada para el servicio Systemd de plantilla

Es posible que haya visto una plantilla de servicio systemd que contiene el símbolo @ como el servicio Syncthing.

```
/lib/systemd/system/[email
```

Este tipo de servicio le permite ejecutar múltiples instancias. Por ejemplo, puedes ejecutar Syncthing para dos usuarios con:

```
sudo systemctl start syncthing@user1.service
sudo systemctl start syncthing@user2.service
```

Si desea agregar una configuración personalizada a este tipo de servicio, debe crear una carpeta para cada uno de estos servicios.

```
sudo mkdir -p /etc/systemd/system/syncthing@user1.service.d
sudo mkdir -p /etc/systemd/system/syncthing@user2.service.d
```

Luego puede crear el archivo de configuración personalizado.

```
sudo nano /etc/systemd/system/syncthing@user1.service.d/custom.conf
sudo nano /etc/systemd/system/syncthing@user2.service.d/custom.conf
```

Después de guardar los cambios. Recargar systemd.

```
sudo systemctl daemon-reload
```

Temporizadores del sistema

¿Qué sucede si no desea que un servicio systemd se inicie automáticamente en el momento del inicio del sistema, sino que se ejecute en un momento específico? Puede crear un temporizador systemd para el servicio systemd. Por ejemplo, cree un servicio systemd (/etc/systemd/system/iris.service) para el software de protección ocular Iris.

[Unit]

```
Description=Iris Eye Protection Software
After=lightdm.service
```

```
[Service]
Type=oneshot
Environment=DISPLAY=:0
User=your_username
ExecStart=/bin/bash -c 'DISPLAY=:0 xhost 127.0.0.1 && /opt/iris/Iris.sh'
KillSignal=SIGINT
Restart=on-failure
RestartSec=2
```

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

Seguro que puedo hacer que se inicie automáticamente en el momento del arranque con:

```
sudo systemctl enable iris.service
```

¿Qué pasa si quiero que empiece sólo por la noche? Necesito crear un temporizador systemd.

```
sudo nano /etc/systemd/system/iris.timer
```

Agregue las siguientes líneas en este archivo. Este temporizador systemd iniciará `iris.service` a las 18:00:00 todos los días.

```
[Unit]
Description=Iris Night Mode Timer
Requires=iris.service
```

```
[Timer]
Unit=iris.service
OnCalendar=*-*-* 18:00:00
```

```
[Install]
WantedBy=timers.target
```

Guarde y cierre el archivo. Luego habilite este temporizador systemd.

```
sudo systemctl enable iris.timer
```

Verifique el estado de este temporizador.

```
sudo systemctl status iris.timer
```

Salida de muestra:

```
● iris.timer - Iris Night Mode Timer
   Loaded: loaded (/etc/systemd/system/iris.timer; enabled; vendor preset:
enabled)
   Active: active (waiting) since Sat 2022-05-07 19:38:22 +08; 24min ago
   Trigger: Sun 2022-05-08 18:00:00 +08; 22h left
   Triggers: ● iris.service
```