

Manejo de servicios Linux:

Linux ofrece multitud de servicios o servidores, estos pueden iniciar o arrancar junto con la carga del sistema o pueden después ser puestos a funcionar cuando se requieran (es lo mejor). Parte esencial de la administración de sistemas Linux es continuamente trabajar con los servicios que este proporciona, cosa que es bastante sencilla. En este tutorial aprenderás todo lo necesario sobre como iniciar/detener/reiniciar etc. los servicios de tu equipo GNU/Linux.

1. Iniciando servicios desde el arranque del sistema

En muchos casos es conveniente que un servidor o servicio inicien junto con el arranque del equipo en si, por ejemplo, el servidor web Apache o alguna base de datos, esto es para que estén disponibles todo el tiempo y no se requiera de intervención del administrador para iniciarlos.

En Linux, a diferencia de otros sistemas operativos, es posible configurarlo en base a niveles de ejecución (run levels), cada nivel de ejecución (en la mayoría de las distribuciones son 7), inicia o detiene (Start o Kill) ciertos servicios. Estos niveles son los siguientes:

- **0** Detener o apagar el sistema
- **1** Modo monousuario, generalmente utilizado para mantenimiento del sistema
- **2** Modo multiusuario, pero sin soporte de red
- **3** Modo multiusuario completo, con servicios de red
- **4** No se usa, puede usarse para un inicio personalizado
- **5** Modo multiusuario completo con inicio gráfico (X Window)
- **6** Modo de reinicio (reset)

Por ejemplo, el nivel 0, que apaga el equipo, mata o detiene a todos los procesos del sistema, todos los servicios, lo mismo hace el 6 con la diferencia que después inicia un script que permite reiniciar el sistema.

El nivel por omisión o por default del sistema está definido en `/etc/inittab`, en la línea init default:

```
$> grep initdefault /etc/inittab id:5:initdefault:  
# runlevel 0 is System halt  
# runlevel 6 is System reboot
```

La salida es de una distro OpenSuse 10.

Existe una línea que comienza con id, seguido por 5 que indicaría el nivel por default del sistema, basta con cambiar este valor con cualquier editor, reiniciar el equipo y se entraría a otro nivel, por ejemplo, el 3 que tiene una funcionalidad completa pero sin sistema de ventanas X Windows.

En el listado previo aparecen dos líneas de comentarios, las que empiezan con #, que hacen la advertencia de no usar nunca los niveles 0 y 6 de init default, ya que jamás arrancaría el equipo. Se tendría que usar un método de rescate.

El nivel de ejecución actual del sistema puede ser consultado con **who -r** y con el comando **runlevel**, este último muestra dos números el primero es el nivel previo en que se estaba y el segundo el nivel actual, si no se ha cambiado de nivel aparece una N.

```
usuario@duarry-ALDA:~$ who -r
      `run-level' 5 2019-03-19 12:04
usuario@duarry-ALDA:~$ runlevel
N 5
usuario@duarry-ALDA:~$
```

2. Comando init

Este comando te permite cambiar el nivel de ejecución actual por otro, es decir puedes estar en el nivel 3 y en cualquier momento pasar al 5 o viceversa, o entrar a un modo monousuario para tareas de mantenimiento. Esto sin necesidad de cambiar la línea init default en /etc/inittab, simplemente usando el comando init:

\$ init 3

Puedes **reiniciar** el equipo con **init 6** o incluso **apagarlo** con **init 0**, de hecho, shutdown y otros comandos como halt, lo que hacen es invocar a init para apagar el equipo más otras tareas como sincronizar buffers, etc.

Si estás en una máquina que tenga particiones con sistemas de archivos ext2 o de windows como FAT, si invocas init 0 o init 6 tal cual, corres el riesgo de perder datos que no estén guardados en disco todavía, para esto combina init 0 con el comando sync que escribe lo que este en buffers de memoria a disco:

\$ sync; sync; init 0

Con un doble sync nos aseguramos que todo se escriba en disco antes de comenzar a apagar.

3. Iniciando servicios manualmente, directorio init.d

Dentro de esta carpeta ubicada en /etc o en /etc/rc.d dependiendo de la distribución, se encuentran una serie de scripts que permiten iniciar/detener la gran mayoría de los servicios/servidores que estén instalados en el equipo.

Estos scripts están programados de tal manera que la mayoría reconoce los siguientes argumentos:

- start
- stop
- restart
- status

Los argumentos son auto descriptivos, y tienen permisos de ejecución, entonces siendo root es posible iniciar un servicio de la siguiente manera, por ejemplo, samba:

#> /etc/init.d/docker start

```
usuario@duarry-ALDA:/etc/init.d$ /etc/init.d/docker start
[ ok ] Starting docker (via systemctl): docker.service.
usuario@duarry-ALDA:/etc/init.d$
```

Solo que hay que cambiar **start** por **stop** | **restart** | **status** para detenerlo, reiniciarlo (releer archivos de configuración) o checar su estatus.

4. El comando Service:

Este comando permite también iniciar y/o detener servicios, de hecho funciona exactamente igual a como si escribiéramos la ruta completa hacía el directorio `init.d`, con `service` se indica de la siguiente manera:

\$ service mysql status

```
usuario@duarry-ALDA:/etc/init.d$ service mysql status
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: en
   Active: active (running) since Tue 2019-03-19 12:04:12 CET; 1h 15min ago
   Main PID: 1578 (mysqld)
     Tasks: 28 (limit: 4915)
    CGroup: /system.slice/mysql.service
            └─1578 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pid

Warning: Journal has been rotated since unit was started. Log output is incomplete.
lines 1-9/9 (END)
```

Si se desea iniciarlo/pararlo:

\$ service mysql start/stop

5. Los directorios rc:

Debajo de `/etc` se encuentra el directorio `rc.d` que a la vez contiene un directorio para cada nivel de ejecución, así tenemos `rc0.d`, `rc1.d`, `rc2.d`, `rc3.d`, etc. Hay algunos SO en los que estos directorios están ubicados directamente en `/etc`.

Cada uno de estos directorios contiene scripts (o más bien enlaces a scripts) que apuntan al directorio `init.d`, entonces el comando `initt` dependiendo del nivel indicado leerá cada uno de los enlaces o accesos directos del directorio respectivo.

Un ejemplo:

```
lrwxrwxrwx 1 root root 7 Oct 20 10:05 K22dbus -> ../init.d/dbus
lrwxrwxrwx 1 root root 9 Oct 20 10:05 K22resmgr -> ../init.d/resmgr
lrwxrwxrwx 1 root root 8 Oct 20 10:05 K24fbset -> ../init.d/fbset
lrwxrwxrwx 1 root root 9 Oct 20 10:05 K24random -> ../init.d/random
lrwxrwxrwx 1 root root 8 Oct 20 18:23 S01fbset -> ../init.d/fbset
lrwxrwxrwx 1 root root 9 Oct 20 18:22 S01random -> ../init.d/random
```

Todos son enlaces al directorio `init.d` que es donde realmente están ubicados los scripts de arranque de los servicios o servidores. También, todos los enlaces comienzan con K (kill) o con S (start), los que comienzan con K son scripts que recibirán el argumento `stop` y los que comienzan con S el de `start`.

Después de la K o S sigue un número consecutivo, seguido generalmente del nombre del servicio que afectan, el número secuencial es simplemente el orden en que se leerán los scripts, primero los K comenzando con el 01 y hacia adelante y después los S. Entonces si por ejemplo no queremos que se inicie samba en el nivel 3 bastaría con borrar su enlace en este directorio:

```
$ pwd /etc/rc.d/rc3.d
$ rm S54smb
```

Si por lo contrario lo que deseamos es iniciar (o apagar) un servicio, basta con crear su enlace en el directorio respectivo:

```
$ pwd /etc/rc.d/rc5.d
$ ln -s /etc/rc.d/init.d/mysql S90mysql
$ ls -l S90mysql
```

Con esto la siguiente vez que iniciemos el equipo o cambiemos a nivel 5 con `init`, también se iniciará el servidor de la base de datos MySQL. El número 90 es escogido al azar entre 01 y 99 es simplemente el orden en que serán iniciados o detenidos los servicios.

6. Comando `chkconfig`

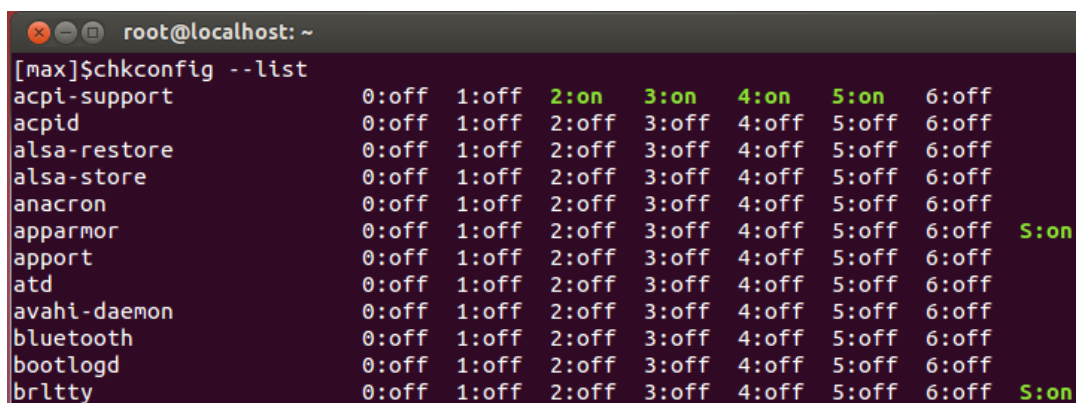
Es una herramienta para configurar los servicios más ágilmente, y que permite configurar que servicios se arrancan/detienen en cada nivel de ejecución.

`Chkconfig` no inicia ni detiene servicios al momento, tan solo crea o elimina precisamente los enlaces. Si lo que se quiere es iniciar o detener el servicio manualmente hay que usar `service` o directamente el script con su argumento conveniente.

- Ejemplos:

a) Con la opción `--list` nos da una lista completa de todos los servicios instalados y para cada nivel si arrancará (on) al entrar a ese nivel o se detendrá (off) o simplemente no se iniciará. Nótese que al final de la lista vienen los servicios que dependen del superservidor `xinetd`.

```
$ chkconfig --list
```



```
[max]$chkconfig --list
acpi-support      0:off  1:off  2:on   3:on   4:on   5:on   6:off
acpid             0:off  1:off  2:off  3:off  4:off  5:off  6:off
alsa-restore      0:off  1:off  2:off  3:off  4:off  5:off  6:off
alsa-store        0:off  1:off  2:off  3:off  4:off  5:off  6:off
anacron           0:off  1:off  2:off  3:off  4:off  5:off  6:off
apparmor          0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
appport           0:off  1:off  2:off  3:off  4:off  5:off  6:off
atd               0:off  1:off  2:off  3:off  4:off  5:off  6:off
avahi-daemon      0:off  1:off  2:off  3:off  4:off  5:off  6:off
bluetooth         0:off  1:off  2:off  3:off  4:off  5:off  6:off
bootlogd          0:off  1:off  2:off  3:off  4:off  5:off  6:off
brltty            0:off  1:off  2:off  3:off  4:off  5:off  6:off  S:on
```

b) Si se indica como argumento el nombre de algún servicio nos regresará su estatus:

`$ chkconfig smb`

c) Sin ningún argumento nos dará una lista con el status de todos los servicios:

`$ chkconfig`

`httpd off`
`mysql on`
`smb on`

d) Si se desea que el servidor web apache (httpd) inicie cuando se entra en el nivel 5, entonces usamos la opción --level:

`$ chkconfig --level 5 httpd on`

e) Si la base de datos MySQL no se desea que inicie en los niveles 3 y 5, solo hasta que el administrador decida arrancarla, se ejecuta:

`$ chkconfig --level 35 mysql off`