

# SysV

El sistema [SysV](#) controla la gestión de los servicios de un sistema durante su puesta en marcha. [SysV](#) analiza el estado del sistema y decide si se deben iniciar o parar cada uno de los servicios configurados.

[SysV](#) se basa en el concepto de niveles de ejecución para determinar los servicios y procesos que se deben ejecutar en cada uno de ellos. En total, [SysV](#) gestiona 7 niveles de ejecución diferentes, que van desde el 0 hasta el 6 y que tienen las siguientes funciones:

| Nivel | Definición  |
|-------|---|
| 0     | es un nivel transitorio. Se utiliza para pasar de un estado iniciado a uno de parada. Es decir, para apagar el sistema  |
| 1     | “single user mode”. En este nivel el sistema trabaja en modo monousuario, sin conexión de red y ejecutando los servicios mínimos indispensables para el arranque del sistema. También se conoce como “rescue mode”, y está reservado para este uso en todas las distribuciones de GNU/Linux |
| 2     | Multiusuario sin red. Este nivel se suele caracterizar por permitir que varios usuarios puedan conectar con el sistema, pero sin funcionalidades de red   |
| 3     | Multiusuario con red. Proporciona las mismas funcionalidades que el nivel 2 más el soporte de red   |
| 4     | Normalmente es un nivel de ejecución que no viene definido, si bien el administrador del sistema puede configurarlo para dotarlo de funcionalidad   |
| 5     | Multiusuario gráfico: en sistemas con entorno gráfico este nivel proporciona la posibilidad de iniciar sesión en él a múltiples usuarios, con soporte de red, y con acceso al gestor de ventanas gráfico del sistema  |
| 6     | estado transitorio para pasar del nivel de ejecución actual al nivel por defecto (reinicio del sistema)   |

La definición de los diferentes niveles así como su uso se puede consultar, para todas las distribuciones que trabajen con [SysV](#) en el fichero [/etc/inittab](#)

## inittab

El fichero [/etc/inittab](#) contiene información sobre los procesos que se inician en el arranque del sistema, así como la configuración de cada uno de los niveles de ejecución. Este fichero tiene una sintaxis definida por 4 campos fundamentales:

```
# <id>:<runlevels>:<action>:<process>
```

- <id>: de 1 a 4 caracteres que identifican el tipo de entrada
- <runlevels>: niveles de ejecución donde se aplica la acción indicada.
- <action>: describe la acción que se va a producir.
- <process>: proceso a ejecutar.

Algunas de las acciones que se pueden definir en inittab son:

boot

el proceso se ejecuta en el inicio del sistema. El campo runlevel se ignora.

initdefault

indica cuál es el nivel de ejecución por defecto (al inicio).

once

el proceso se ejecuta solamente una vez cuando se accede al nivel indicado.

respawn

el proceso se lanza cada vez que se finaliza

sysinit

el proceso se ejecuta durante el inicio del sistema, antes de las acciones boot. El campo runlevel se ignora.

wait

el proceso se ejecuta solamente una vez cuando se accede al nivel indicado, y init espera a que acabe para continuar.

Si miramos el contenido del fichero inittab podemos ver algunas de las configuraciones que comentamos.

```
$ cat /etc/inittab
```

```
### Aquí podemos ver los diferentes niveles de ejecución de este sistema, y las funciones que ofrecen cada uno de ellos.
```

```
# Default runlevel. The runlevels used by RHS are:
```

```
# 0 - halt (Do NOT set initdefault to this)
```

```
# 1 - Single user mode
```

```
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
```

```
# 3 - Full multiuser mode
```

```
# 4 - unused
```

```
# 5 - X11
```

```
# 6 - reboot (Do NOT set initdefault to this)
```

```
#
```

```
### Este es el nivel de ejecución que se inicia por defecto.
```

```
id:3:initdefault:
```

```
### La siguiente línea nos indica el script que se va a cargar en el proceso de arranque del sistema.
```

```
# System initialization.
```

```
si::sysinit:/etc/rc.d/rc.sysinit
```

```
### Scripts de arranque para cada uno de los niveles de ejecución del sistema
```

```
l0:0:wait:/etc/rc.d/rc 0
```

```
l1:1:wait:/etc/rc.d/rc 1
```

```
l2:2:wait:/etc/rc.d/rc 2
```

```
l3:3:wait:/etc/rc.d/rc 3
```

```
l4:4:wait:/etc/rc.d/rc 4
```

```
l5:5:wait:/etc/rc.d/rc 5
```

```
l6:6:wait:/etc/rc.d/rc 6
```

```
### Comportamiento del sistema ante la combinación de teclas Ctrl+Alt+Del
```

```
# Trap CTRL-ALT-DELETE
```

```
ca::ctrlaltdel:/sbin/shutdown -t3 -r now

### Comportamiento del sistema ante problemas de tensión
# When our UPS tells us power has failed, assume we have a few minutes
# of power left. Schedule a shutdown for 2 minutes from now.
# This does, of course, assume you have powerd installed and your
# UPS connected and working correctly.
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"
# If power was restored before the shutdown kicked in, cancel it.
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"

### Configuración de los tty que arrancan en el inicio.
# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6

### Con esta configuración iniciamos el entorno gráfico cuando escogemos el
nivel 5
# Run xdm in runlevel 5
x:5:respawn:/etc/X11/prefdm -nodaemon
```

## **/etc/init.d/**

El directorio de trabajo [/etc/init.d/](#) (/etc/rc.d/init.d/ en algunas distribuciones) es el encargado de almacenar todos los scripts de inicialización o parada de los servicios del sistema. Por lo tanto, si queremos construir nuestro propio servicio, y definir su comportamiento en el inicio del sistema, hemos de crear en este directorio el script que nos lo permita gestionar. Si queremos poder gestionar estos scripts con herramientas del sistema como service o chkconfig es importante que en la cabecera del script aparezcan las siguientes líneas comentadas.

```
# chkconfig: 2345 12 88
```

También sería importante que los scripts que construyamos cumplan el LSB (Linux Standard Base), para intentar reducir las diferencias en el modo de iniciar servicios. Según el LSB nuestro script debería tener una cabecera con esta información:

```
# BEGIN INIT INFO
# Provides: my_daemon
# Required-Start: postgresql networking
# Required-Stop: postgresql networking
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: This is a test daemon
# Description: This is a test daemon
# This provides example about how to
```

```
# write a Init script.
# END INIT INFO
```

De esta manera estamos indicando el nombre de nuestro servicio (para poder ser llamado por otros), así como sus posibles dependencias tanto en el inicio como en la parada, además de su comportamiento para cada uno de los niveles de ejecución del sistema.

## /etc/rcN.d

Como hemos comentado, [/etc/init.d](#) contiene los scripts que se ejecutan para el inicio o parada de los servicios, pero no contiene suficiente información sobre como actuar en cada nivel de ejecución. Para ello tenemos los directorios rcN.d. Existen un total de 7 directorios “rcN.d”, uno para cada nivel de ejecución existente. En ellos lo que tenemos son enlaces simbólicos a los scripts que se encuentran en [/etc/init.d](#), pero con una nomenclatura particular.

```
$ ls -l /etc/rc3.d/
lrwxrwxrwx 1 root root 18 Feb  4 2020 K89netplugd -> ../init.d/netplugd
lrwxrwxrwx 1 root root 15 Feb  4 2020 K89rdisc -> ../init.d/rdisc
lrwxrwxrwx 1 root root 16 Feb  4 2020 S07iscsid -> ../init.d/iscsid
lrwxrwxrwx 1 root root 19 Feb  4 2020 S08ip6tables -> ../init.d/ip6tables
```

Si nos fijamos, todos los enlaces tienen la misma estructura en el nombre:

- S/K: Start/Kill. Nos indica si el servicio se debe iniciar o parar
- NN: Valor numérico que nos indica la posición en la que se va a ejecutar ese script (recordamos que [SysV](#) trabaja de manera secuencial)
- [NAME]: nombre del servicio al que hace referencia el enlace.

## chkconfig

El comando [chkconfig](#) nos permite ver y manipular la información sobre la activación y desactivación de los servicios del sistema en cada uno de sus niveles de ejecución.

Por ejemplo, si queremos tener una vista general del estado de los servicios podemos ejecutar el comando:

```
$ chkconfig --list
acpid                0:off  1:off  2:on   3:on   4:on   5:on   6:off
bootlogd             0:off  1:off  2:off  3:off  4:off  5:off  6:off
S:on
bootlogs             0:off  1:on   2:on   3:on   4:on   5:on   6:off
bootmisc.sh          0:off  1:off  2:off  3:off  4:off  5:off  6:off
S:on
checkfs.sh           0:off  1:off  2:off  3:off  4:off  5:off  6:off
S:on
...
```

Si queremos ver la configuración de un servicio en concreto lo podemos hacer con:

```
$ chkconfig --list sshd
sshd          0:off  1:off  2:on   3:on   4:on   5:on   6:off
```

El comando `chkconfig` también nos permite modificar el comportamiento de un servicio. Para ellos podemos hacer uso de la instrucción:

```
# chkconfig -level <RUNLEVELS> <PROCESS> <ACTION>
```

donde:

- RUNLEVELS: niveles en los que queremos aplicar la modificación.
- PROCESS: servicio sobre el que queremos actuar
- ACTION: que queremos hacer
- on: activar el proceso
- off: desactivar el proceso
- reset: volver al valor por defecto

Si queremos desactivar el servicio SSH en los niveles 3 y 5 ejecutaríamos:

```
# chkconfig -level 35 ssh off
```

Para comprobar que la configuración se realizó correctamente:

```
$ chkconfig --list sshd
sshd          0:off  1:off  2:on   3:off  4:on   5:off  6:off
```

## service

Para poder gestionar los servicios de forma manual tenemos 2 opciones. Una es pasar al script del servicio la acción a realizar (start/stop/restart/status). La otra es utilizar el comando `service`. Este comando nos permite indicar al sistema el tipo de acción a realizar sobre un servicio indicado.

```
# service [SERVICE] [ACTION]
```

donde las acciones son las mismas que le podíamos pasar directamente al script que se encuentra en </etc/init.d>. En cambio, el comando `service` no necesita la ruta completa del script, siempre y cuando éste se encuentre en alguna ubicación reconocible.

From:  
<https://wiki.deceroauno.net/> - **DE 0 A 1**

Permanent link:  
<https://wiki.deceroauno.net/doku.php?id=basics:sysv>

Last update: **2021/01/11 05:45**

