

upstart

1. Fundamentos

Upstart es el sustituto del servicio init, inicialmente desarrollado para Ubuntu por un empleado de la sociedad Canonical, que funciona de manera asíncrona y basado en eventos. En Ubuntu lo encontramos hasta la versión 14.04, en Debian hasta la versión 6, al igual que Red Hat Enterprise Linux 6 y ciertas versiones de Fedora y las versiones derivadas asociadas (Oracle Linux por ejemplo). De manera más sorprendente, upstart lo usa Google Chrome OS (y Chromium OS) en la base de todos los Chromebooks del mercado.

Para comprender el poder de upstart, puede visitar la página dedicada al arranque y a la gestión de servicios en Chrome OS : <https://www.chromium.org/chromium-os/chromiumos-design-docs/boot-design>

Como puede ver, upstart, sencillo y ligero, es usado por muchos productos, y como el soporte de Red Hat Enterprise 6 (ELS) termina en junio de 2024, hay que conocerlo.

upstart controla el conjunto de servicios, gestiona su arranque y su detención y supervisa también su funcionamiento. Por razones evidentes, mantiene compatibilidad con init System V: cualquier servicio System V funcionará con upstart, sin embargo los servicios upstart no funcionan con System V. upstart es de momento el init predeterminado de muchas distribuciones, y su migración se efectúa con facilidad gracias a esta compatibilidad.

Con upstart, los servicios pueden iniciarse de forma asíncrona y, por lo tanto, pueden paralelizarse y no iniciarse secuencialmente como con init. De este modo, se pueden definir dependencias entre los servicios y optimizar sus inicios y paradas. Basado en eventos, upstart podrá, por ejemplo, iniciar los servicios Bluetooth únicamente si se ha conectado un pendrive de este tipo, y no de forma inútil y sistemática en el arranque de la máquina.

2. Archivos

Los archivos de configuración de upstart están en **/etc/init**. Son archivos de configuración con sintaxis específica, aunque también pueden contener shell script clásico. A continuación tenemos a modo de ejemplo el archivo de configuración que se encarga del servicio sshd:

```
# ssh - OpenBSD Secure Shell Server
#
# The OpenSSH server provides secure shell access to the system.

Description    "OpenSSH server"

start on filesystem or runlevel [2345]
stop on runlevel [!2345]

respawn
respawn limit 10 5
umask 022

pre-start script
test -x /usr/sbin/sshd || { stop; exit 0; }
test -e /etc/ssh/sshd_not_to_be_run && { stop; exit 0; }
test -c /dev/null || { stop; exit 0; }

mkdir -p -m0755 /var/run/sshd
end script

# if you used to set SSHD_OPTS in /etc/default/ssh, you can change the
# 'exec' line here instead
exec /usr/sbin/sshd -D
```

La sintaxis es comprensible para el administrador. Las directivas **start** y **stop** indican con qué eventos se inicia y se detiene el servicio. En este caso, el servicio se iniciará si los sistemas de archivos están montados o si se está en los niveles de 2 a 5. Se detendrá en todos los niveles salvo los que están entre 2 y 5 (ambos incluidos), a saber: 0, 1 y 6. Una parte de la sintaxis retoma la de inittab, especialmente en las acciones (por ejemplo respawn).

3. Nivel por defecto

El archivo **/etc/inittab** ya no es necesario y no se interpretará, pero hay una excepción. Si está presente, es únicamente por la línea **initdefault**. Si no existe, el nivel de ejecución predeterminado se indica en la variable **DEFAULT_RUNLEVEL** del archivo **/etc/init/rc-sysinit.conf**. El nivel de ejecución siempre puede pasarse por parámetro al núcleo en el arranque.

4. Compatibilidad con System V

Los scripts **rc-sysinit.conf**, **rcS.conf** y **rc.conf** siempre están para garantizar la compatibilidad con los scripts System V. El script **rc.conf** es el que va a ejecutar el archivo **/etc/init.d/rc** con el nivel de ejecución pasado por parámetro. A partir de este punto, el método es idéntico al método init System V.

5. Comandos de control

Aunque un script init System V siempre puede controlarse con su nombre o por el comando service, estos métodos no son recomendables. upstart dispone de un cierto número de comandos que los reemplazan.

Iniciar un servicio

Utilice el comando start:

```
# start ufw
ufw start/running
```

Recargar un servicio

Utilice el comando restart:

```
# restart ufw
ufw start/running
```

Parar un servicio

Utilice el comando stop:

```
# stop ufw
ufw stop/waiting
```

Estado de un servicio

Utilice el comando status:

```
# status ufw
ufw start/running
```

initctl

Puede utilizar el comando **initctl** para controlar servicios. Con él se pueden realizar todas las acciones posibles. Recibe como primer parámetro una acción y, si procede, un segundo parámetro con el nombre del servicio asociado. Por ejemplo, puede listar los servicios:

```
# initctl list
avahi-daemon start/running, process 1410
mountall-net stop/waiting
nmbd start/running, process 2030
qemu-kvm start/running
rc stop/waiting
rpc_pipefs (idmapd) start/running
rsyslog start/running, process 1362
screen-cleanup stop/waiting
tty4 start/running, process 1511
udev start/running, process 490
upstart-udev-bridge start/running, process 486
```

```
ureadahead-other stop/waiting
apport stop/waiting
console-setup stop/waiting
...
```

El comando también permite emitir un evento con **emit**. Por ejemplo, para cambiar el nivel de ejecución (lo que siempre se puede hacer con los comandos telinit, halt, reboot o shutdown):

```
# initctl emit runlevel RUNLEVEL=2
```

6. Activación y desactivación de un servicio

No hay sustituto hasta la fecha de **chkconfig** o **update-rc.d** para los servicios upstart. Sin embargo, si la distribución mantiene cierta compatibilidad con System V y upstart sirve básicamente para controlar estos servicios, estos comandos continuarán siendo funcionales para cualquier servicio albergado en **/etc/init.d**.

En caso contrario, necesitará:

- ▾ detener manualmente el servicio,
- ▾ modificar los eventos start y stop en los servicios asociados en /etc/init.