

# Arranque Rápido



Uzin, José Maria (joseuzin@hotmail.com)

Copyright (C) 2012 JOSE MARIA UZIN.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

## Índice de contenido

Introducción.....	4
Proceso de Arranque Linux.....	5
Niveles de Arranque o Ejecución.....	6
Proceso Init.....	6
<i>Directorios de Niveles de Arranque.....</i>	<i>8</i>
<i>Lanzando los niveles de arranque.....</i>	<i>8</i>
Sistema de Arranque Upstart.....	9
<i>Conceptos y Terminología Upstart.....</i>	<i>10</i>
<i>Proceso de Inicio Upstart.....</i>	<i>12</i>
<i>Directorios y Configuraciones Upstart.....</i>	<i>13</i>
<i>Estructura de Configuración de un Trabajo (Job).....</i>	<i>15</i>
<i>Ejemplos de Archivos de Configuración.....</i>	<i>16</i>
<i>Comandos Usados para interactuar con Upstart.....</i>	<i>18</i>
<i>Deshabilitar Un Trabajo del Inicio Automático.....</i>	<i>18</i>
Diferencias Demonio SystemV Init y Upstart.....	18
<i>Beneficios del Sistema SystemV Init.....</i>	<i>18</i>
<i>Limitaciones del Sistema System V.....</i>	<i>18</i>
<i>Beneficios del Sistema de Arranque Upstart.....</i>	<i>20</i>
Conclusión.....	21
Bibliografía.....	22

# Introducción

Habiendo estudiado como inicia un sistema operativo GNU/Linux en Distribución Debian, quise buscar los mismos archivos de configuración en la Distribución Ubuntu 12.04, llevándome la sorpresa de que algunos archivos no existían directamente, por ejemplo el */etc/inittab* (que definía como iniciaba el sistema en Debian), esto me llevo a investigar en Internet porque no estaban en el sistema, descubriendo que Ubuntu usaba una filosofía de inicio totalmente distinta a la de Debian, basada en eventos y trabajos, llamada Upstart, que funciona asincrónicamente y no de manera sincrónica como el System V Init de Debian. Esto me llamo la atención y motivó el siguiente trabajo, cuya fuente de información fueron la Web y el Curso de Administración GNU/Linux Nivel I de Gugler.

Aunque han aparecido diversos trucos para acelerar el proceso de Inicio de System V Init a lo largo de estos años, la mayoría de las reformas del sistema de arranque han resultado ser inviables en la práctica, y muchos de los turbo cargadores empleados no están al alcance de los usuarios corriente.

Upstart adopta una solución diferente para acelerar el proceso de arranque de Linux. El Proyecto Upstart presenta un servicio Init genérico que es la base de muchos de los desarrollos de los sistemas Linux modernos. Upstart inteligentemente evita las esperas innecesarias, ejecuta los scripts de arranque de forma simultánea y reduce el tiempo de arranque al mínimo. El plan es reemplazar los servicios del sistema que se ejecutan en segundo plano, como el servicio at, cron y otros por Upstart. La distribución Ubuntu 6.10 (Edgy Eft) muestra el primer efecto de esta herramienta.

# Proceso de Arranque Linux

Antes de entender las diferencias entre el inicio clásico según el demonio System V Init y el sistema Upstart debemos entender como es el proceso de arranque del Sistema Operativo.

1. Encendemos el equipo. Se realiza una comprobación del hardware.
2. El sistema Básico de Entrada y Salida (BIOS) se carga y busca un sector de arranque válido.
3. Se lee el Master Boot Record (M.B.R), ubicado en los primeros 512 Bytes del primer sector (sector 0) del primer dispositivo activo.
4. Cualquier código de cargador de arranque encontrado es ejecutado. En este punto la BIOS deja de tener el control de la secuencia de arranque.
5. Si seleccionamos una imagen de Linux, se lee la ruta para cargar el kernel, se descomprime el kernel y se carga en memoria.
6. A medida que el kernel se va cargando inicializa los dispositivos y carga los módulos (drivers) necesarios. A continuación monta el sistema raíz (/), tal y como está definido en la entrada *root=/dev/XXX* del gestor de arranque.
7. Se carga el programa */sbin/init* y se convierte en el proceso con PID 1, el abuelo del resto de procesos del sistema.
8. El proceso *Init* lee el fichero */etc/inittab* y ejecuta el script */etc/init.d/rcS*.
9. El script de inicialización *rcS* carga los módulos necesarios, comprueba el sistema de ficheros raíz, monta los sistemas de archivos locales, inicia los dispositivos de red y luego, si están configurados, monta los sistemas de archivos remotos.
10. El proceso *Init* vuelve a leer */etc/inittab* y cambia el sistema al nivel de arranque definido por defecto ejecutando los scripts especificados en el directorio apropiado.
11. Los scripts del nivel de arranque del tipo *XXservicio* son iniciados, Start, en el orden especificado por los números *XX*.
12. Debian utiliza enlaces en los directorios */etc/rc0.d* a */etc/rc6.d* que apuntan al directorio */etc/init.d* que es dónde se encuentran los scripts ejecutables.

13. Se cargan las sesiones de terminal especificadas en */etc/inittab* y la utilidad *mingetty* muestra el prompt de *login* y queda a la espera de que el usuario se valide en el sistema (mediante usuario y contraseña).

## Niveles de Arranque o Ejecución

Un nivel de arranque o nivel de ejecución es la definición de un estado del sistema, en el que un número (por defecto) representa un conjunto de servicios y algunas aplicaciones que se ejecutan.

Existen 6 niveles de ejecución

Nivel de ejecución	Descripción	Ubuntu
0	Modo Apagado Sistema Finaliza servicios y programas activos, así como desmonta filesystems activos y para la CPU.	Modo Apagado Sistema Finaliza servicios y programas activos, así como desmonta filesystems activos y para la CPU.
1	Modo Mantenimiento, no se puede cambiar de usuario, interprete de comandos y algunas utilidades de rescate.	Modo Mantenimiento, no se puede cambiar de usuario, interprete de comandos y algunas utilidades de rescate
2	Multiusuario sin soporte red	En Ubuntu, son todos iguales, sistema completo multiusuario, soporte de red y modo gráfico por defecto.
3	Multiusuario con soporte red	
4	Igual a nivel 3 no suele usarse	
5	Multiusuario con entorno gráfico	
6	Modo Reinicio Sistema	Modo Reinicio Sistema

Para determinar el Nivel de Ejecución actual ejecutamos el comando ***runlevel***



## Proceso Init

Cuando se ejecuta, el programa */sbin/init* se convierte en el proceso número 1 o padre y se le asigna como PID 1. El resto de procesos son hijos o descendientes de este y, por defecto, heredan el entorno y los atributos del proceso *Init*. El proceso *Init* se configura en el archivo */etc/inittab*.

### Ejemplo de Configuración de Inittab

```
# /etc/inittab: init
```

```
# Nivel de arranque por defecto.
```

```
id:2:initdefault:
```

```
# script inicialización/configuración de booteo.
```

```
# Es ejecutado primero excepto en modo de recuperación.
```

```
si::sysinit:/etc/init.d/rcS
```

```
# /etc/init.d executes the S and K scripts upon change  
# of runlevel.
```

```
# Runlevel 0 is halt.
```

```
# Runlevel 1 is single-user.
```

```
# Runlevels 2-5 are multi-user.
```

```
# Runlevel 6 is reboot.
```

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

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

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

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

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

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

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

```
# /sbin/getty invocations for the runlevels.
```

```
1:2345:respawn:/sbin/mingetty --noclear --autologin matt tty1
```

```
2:23:respawn:/sbin/mingetty tty2
```

```
3:23:respawn:/sbin/mingetty tty3
```

```
4:23:respawn:/sbin/mingetty tty4
```

```
5:23:respawn:/sbin/mingetty tty5
```

```
6:23:respawn:/sbin/mingetty tty6
```

Las entradas en el archivo inittab tienen la siguiente sintaxis

id:niveles:accion:proceso

El sistema, y en particular el proceso *Init*, obtienen su comportamiento de este archivo; es el responsable de los niveles de arranque, de cuantos programas se ejecutan y responden a señales y de como el sistema es iniciado. Después, el sistema, según el nivel de ejecución elegido, consulta los ficheros contenidos en */etc/rcn.d*, donde n es el número asociado al nivel de ejecución, en el que se encuentra una lista de servicios por activar o parar en caso de que arranquemos en el nivel de ejecución, o lo abandonemos. Dentro del directorio encontraremos una serie de enlaces simbólicos a los scripts que controlan el servicio. Los scripts de inicio son scripts ubicados en */etc/init.d*. Cada script posee un nombre relacionado con el servicio, una S o K inicial que indica si es el script para iniciar (S) o matar (K) el servicio, y un número que refleja el orden en que se ejecutarán los servicios. El nivel de arranque por defecto permite al sistema saber a que nivel de arranque queremos entrar cuando los scripts de inicio del sistema */etc/init.d/rcS* terminan. Esta entrada es importante porque define que ejecutará el sistema al arrancar o reiniciarse.

*id:2:initdefault ( nivel de inicio por defecto, en este caso nivel 2)*

La entrada de inicio del sistema establece los scripts que se ejecutaran al arrancar y antes de entrar en algún nivel de arranque. No hay niveles de arranque para esta entrada ya que estos tienen lugar después. Por lo tanto las acciones son comunes y previas a los niveles de arranque.

`si::sysinit:/etc/init.d/rcS`

## Directorios de Niveles de Arranque

Las dos variantes principales de sistemas de inicio son BSD/Slackware que contiene unos pocos y largos scripts de inicio y SysV que integra el script `sysinit` y un conjunto de directorios que contienen enlaces a los servicios del sistema. Cada directorio representa un nivel de ejecución. Red Hat y Debian utilizan una estructura similar para esos directorios, con una diferencia fundamental. Las dos estructuras de directorios se muestran a continuación:

Red Hat	Debian
<code>/etc/rc.d</code>	<code>/etc/</code>
<code> -- init.d</code>	<code> -- init.d</code>
<code> -- rc0.d</code>	<code> -- rc0.d</code>
<code> -- rc1.d</code>	<code> -- rc1.d</code>
<code> -- rc2.d</code>	<code> -- rc2.d</code>
<code> -- rc3.d</code>	<code> -- rc3.d</code>
<code> -- rc4.d</code>	<code> -- rc4.d</code>
<code> -- rc5.d</code>	<code> -- rc5.d</code>
<code> -- rc6.d</code>	<code> -- rc6.d</code>
<code> -- rcS.d</code>	<code> -- rcS.d</code>

En ambos casos el directorio `Init.d` contiene todos los ejecutables de los servicios y demonios. Estos son enlazados desde los directorios de los niveles de arranque `rcn.d` de dos maneras : de inicio o de parada. Un nivel de arranque, por tanto, no deja de ser un conjunto de servicios y programas que se lanzan o se detienen en un orden adecuado para evitar problemas de dependencias.

## Lanzando los niveles de arranque

Una vez lanzado el nivel de arranque S, inicial y común a todos los niveles de arranque ya sólo queda lanzar los servicios y programas correspondientes al nivel de arranque establecido por defecto. De ello se encargan las entradas:

`lx:x:wait:/etc/init.d/rc x`

una para x nivel de arranque ( x va de 0 a 6). Por lo tanto el script que se encarga de recorrer el directorio `/etc/rcx.d` y lanzar los servicios y demonios correspondientes es :

`/etc/init.d/rc`



# Sistema de Arranque Upstart

Upstart es un sustituto basado en eventos para el Demonio Init utilizado en sistemas UNIX, como GNU/Linux, que opera **asíncronamente** y dirige el inicio de las tareas y servicios durante el arranque del sistema, los supervisa mientras éste está encendido y los detiene durante su apagado. Upstart ha sido programado por Scott James Remnant, trabajador de Canonical Ltd., está escrito en lenguaje C con licencia pública GNU y a pesar de haber sido desarrollado originariamente para la distribución Ubuntu, está diseñado con la intención de que pueda adaptarse para su uso en todas las distribuciones de GNU/Linux como sustituto de sysvinit (System-V init). La versión inicial fue desarrollada el 24 de Agosto de 2006 y estaba incluida en la versión 6.10 de Ubuntu “Edgy Eft” como reemplazo de sysvinit y en 7.04 “Feisty Fawn.” Actualmente la versión estable es la 1.5 (15 de Marzo de 2012). En Agosto de 2011, Red Hat decidió también adoptar Upstart, dejando de lado SystemV init.

Versión Ubuntu	Nombre Clave	Fecha Salida	Versión Upstart Estables
6.10	Edgy Eft	26-10-2006	Trunk serie (primeras líneas de desarrollo no versión estable)
7.04	Feisty Fawn	19-04-2007	Trunk serie (primeras líneas de desarrollo no versión estable)
7.10	Gutsy Gibbon	18-10-2007	Trunk serie (primeras líneas de desarrollo no versión estable)
8.04 LTS	Hardy Heron	24-04-2008	0.3.9-2
8.10	Intrepid Ibex	30-10-2008	0.3.9-8
9.04	Jaunty Jackalope	23-04-2009	0.3.9-8
9.10	Karmic Koala	29-10-2009	0.6.3-11
10.04 LTS	Lucid Lynx	29-04-2010	0.6.5-8
10.10	Maverick	10-10-2010	0.6.6-4

	Meerkat		
11.04	Natty Narwhal	28-04-2011	0.9.7-1.1
11.10	Oneiric Ocelot	13-10-2011	1.3-0ubuntu12
12.04 LTS	Precise Pangolin	26-04-2012	1.5-0ubuntu7

### *Evolución Temporal Upstart*



## Conceptos y Terminología Upstart

El principal concepto en Upstart son eventos y trabajos.

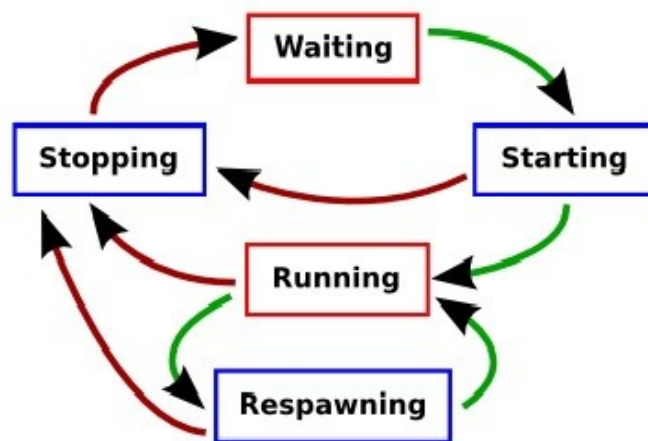
Trabajos (Jobs) : Generalmente una tarea o un servicio. Cada trabajo es definido en su archivo de configuración de trabajo localizado en /etc/init, con extensión (\*.conf). Para que un trabajo pueda ponerse en marcha o pararse, deben darse ciertas condiciones, normalmente estas condiciones vienen dadas por la detección o no de ciertos eventos.

Trabajo Tipo Tarea : Es aquel que ejecuta un proceso de corta duración, es decir que tiene un tiempo de vida definido y un estado de finalización. Son simples scripts que se ejecutan y

terminan su trabajo en un breve lapso de tiempo. Por ejemplo borrar un archivo puede ser una tarea ya que el comando comienza, borra el archivo en cuestión y luego finaliza.

Trabajo Tipo Servicio : Son procesos que se quedan en ejecución (demonios). Es lo opuesto a una tarea ya que un servicio puede nunca terminar por su propia cuenta. Por ejemplo entidades como bases de datos, servidores webs o ftp.

Estados de los Trabajos



### *Estados de los Trabajos Upstart*

- waiting : Estado Inicial.
- starting : Trabajo esta por iniciar.
- spawned : Ejecutando script o sección exec .
- running : Trabajo en ejecución.
- stopping : Deteniendo Trabajo.
- killed : Trabajo detenido.

Los dos estados en rojo waiting (en espera) y running (en ejecución) son estados de reposo, normalmente esperamos que un trabajo se encuentre en esos estados hasta la llegada de un evento. Los otros estados son estados temporales. Los trabajos dejan un estado debido a que el proceso asociado a ellos termina y se mueven al próximo estado apropiado, siguiendo la flecha verde si el trabajo es iniciado o la flecha roja si el trabajo es terminado.

Eventos: Notificaciones enviadas por Upstart a todas las partes interesadas ( trabajos u otros eventos). Los eventos son emitidos al sistema Upstart completo. Si no hay trabajos que registren interés en un evento particular en su condición de inicio (start-on) o detención (stop-on) el evento no tiene efecto en el sistema. Upstart tiene un conjunto especial de eventos que son emitidos para anunciar las transiciones de estado de los trabajos.

Eventos Especiales Upstart emitidos por los Trabajos (Jobs)	Descripción
Iniciando (starting)	Este evento es emitido por Upstart cuando un trabajo esta por ejecutarse.
Iniciado (started)	Este evento es emitido por Upstart cuando un trabajo esta ejecutándose
Deteniendo (stopping)	Evento emitido por Upstart cuando un trabajo va a ser detenido.
Detenido (stopped)	Evento emitido por Upstart cuando el trabajo ha sido completado.

Los eventos son comunicados al demonio Upstart mediante las sentencias *start on* y *stop on* (para inicio o fin de un trabajo)

*start on* <evento>

*stop on* <evento>

## Proceso de Inicio Upstart

1. Upstart realiza su inicialización interna.
2. Upstart emite un evento simple llamado *startup*. Este evento lleva al resto del sistema a inicializar.
3. *Init* ejecuta un número pequeño de trabajos que especifica el evento *startup* en su condición de iniciado. El más notable de estos trabajos es *mountall* que monta discos y sistemas de archivos.
4. El trabajo *mountall* emite un número de eventos. Estos incluyen local-file systems, virtual-filesystems y all-swaps.
5. El evento *virtual-filesystems* causa que comience el trabajo *udev*.

6. El trabajo *udev* causa que el trabajo *upstart-udev-bridge* inicie
7. El trabajo *upstart-udev-brdige* en algún momento emitirá el evento *net-device-up IFACE=lo* significando que la red local está disponible.
8. Después que el último sistema de archivos es montado, *mountall* emitirá el evento *filesystem*.
9. Ya que la condición de inicio del trabajo *rc-sysinit* es *start on filesystem y net-device-up IFACE=lo* Upstart entonces ejecutará el trabajo *rc-sysinit*.
10. El trabajo *rc-sysinit* llama al comando *telinit* pasándole el nivel de ejecución por defecto, *telinit 2*.
11. El comando *telinit* emite el evento *runlevel*.
12. El evento *runlevel* causa que otros trabajos inicien, incluyendo */etc/init/rc.conf* que inicia el sistema de compatibilidad *Init System V (sysvinit)*, ejecuta el script */etc/init.d/rc <nivel de ejecución>*.
13. Los scripts de nivel de arranque son iniciados en el orden especificado por el numero de servicio.
14. Se cargan las sesiones de terminal especificadas y la utilidad *mingetty* muestra el prompt de login y queda a la espera que el usuario se valide en el sistema mediante el nombre de usuario y contraseña.

## Directorios y Configuraciones Upstart

1. */etc/init* : donde se encuentran los archivos de configuración de upstart. Ejecutan lo que sea necesario para reemplazar los scripts sysvinit. En las primeras versiones de Upstart, la definición de Jobs ( trabajos a realizar cuando se detecten ciertos eventos) se hacía en el directorio */etc/event.d*. En Ubuntu Karmic Koala (9.10), Canonical decidió cambiar esa localización a */etc/init*.
2. */etc/init.d* : es donde se encuentran los scripts tradicionales sysvinit y los scripts compatibles de Upstart.
3. */etc/init/rc-sysinit.conf* : controla la ejecución de los scripts tradicionales añadidos manualmente . También controla el nivel de arranque por defecto
4. */etc/default* : archivos de configuración que permiten el control del comportamiento de los scripts sysvinit y upstart.

Si en el arranque SysVInit se configura en */etc/inittab*, Upstart se configura en una serie de ficheros incluidos en la carpeta */etc/init*.

```
$ ls /etc/init
acpid.conf          hwclock.conf        network-manager.conf
alsa-mixer-save.conf hwclock-save.conf   nmbd.conf
anacron.conf        irqbalance.conf     plymouth.conf
apport.conf         module-init-tools.conf plymouth-log.conf
atd.conf            mountall.conf        plymouth-splash.conf
avahi-daemon.conf   mountall-net.conf    plymouth-stop.conf
console-setup.conf  mountall-reboot.conf procps.conf
control-alt-delete.conf mountall-shell.conf  rc.conf
cron.conf           mounted-dev.conf     rcs.conf
cups.conf           mounted-tmp.conf     rc-sysinit.conf
dbus.conf           mounted-varrun.conf  rsyslog.conf
dmesg.conf          mysql.conf           screen-cleanup.conf
failsafe-x.conf     networking.conf      smbd.conf
gdm.conf            network-interface.conf tty1.conf
hostname.conf       network-interface-security.conf tty2.conf
```

El nivel de arranque por defecto se encuentra en *rc-sysinit.conf*

```
$ cat /etc/init/rc-sysinit.conf
# rc-sysinit - System V initialisation compatibility
#This task runs the old System V-style system initialisation scripts,
# and enters the default runlevel when finished.
Description "System V initialisation compatibility"
author "Scott James Remnant <scott@netsplit.com>"
start on filesystem and net-device-up IFACE=lo
stop on runlevel
env DEFAULT_RUNLEVEL=2
console output
env INIT_VERBOSE
# Switch into the default runlevel
telinit "${DEFAULT_RUNLEVEL}"
end script
```

Lo primero que se define es el evento que hace que se lance este trabajo mediante *start on*, en este caso que se monte el sistema de archivos y que se cree el dispositivo virtual de red de loopback. A continuación se incluye el script que se ejecutará para este evento, en este caso primero se comprueba si existe el fichero *etc/inittab* en cuyo caso se busca la línea *initdefault* en la que se establece el nivel de arranque por defecto y este es el que se aplica (se modifica la variable *DEFAULT\_RUNLEVEL*). En los ficheros *ttyX* se definen los terminales que se lanzarán, y dentro de cada uno de ellos se especifica, entre otras cosas en que nivel de arranque se ejecutarán. Otro de los ficheros presentes es *control-alt-delete* en el que se especifica que se ejecuta al recibir la señal generada por la pulsación de esas tres teclas.

Archivo de Trabajo *rc.conf* : En esencia nos dice que ejecute el script SystemV Init */etc/init.d/rc \$RUNLEVEL* cuando el comando *telinit* emita el evento *RUNLEVEL* para cualquier nivel de ejecución.

```
# /etc/init/rc.conf
```

```
start on runlevel [0123456]

stop on runlevel [!$RUNLEVEL]

export RUNLEVEL

export PREVLEVEL

console output env

INIT_VERBOSE

task exec /etc/init.d/rc $RUNLEVEL
```

## Estructura de Configuración de un Trabajo (Job)

Los trabajos serán definidos en archivos de texto plano sin permisos de ejecución que serán colocados en el directorio `/etc/init` y cuyo nombre será el nombre del fichero sin la extensión `.conf`

Description : Descripción del Archivo de configuración de Trabajo

*description <cadena>*

Exec & Script : todos los archivos de trabajo deben tener una instancia de `exec` o `script` en su estructura. Esto especificará que tarea va a ejecutar. `Exec` indica la ruta en el sistema de archivos a un ejecutable.

*Exec /usr/sbin/mysqld*

En cambio, `script` permite escribir código que será ejecutado usando `/bin/sh`. Siempre es usada la opción `-e`, por lo que si un comando del script falla provocará la finalización del mismo. La instancia del script finalizará con la línea “end script”

*script*

*#hacer algo*

*if [...] ; then*

*...*

*fi*

*end script*

Start On y Stop On ( Eventos que iniciarán los trabajos) :El primer evento generado por Upstart es *startup* el cual se produce cuando la máquina es iniciada. También tendremos eventos `runlevel x`, donde `x` es un número entre 0-6 o `s`. Los eventos que iniciarán nuestros trabajos debemos indicarlos con `start on` y los eventos que detendrán los mismos con `stop on`.

*Start on startup*  
*start on runlevel [23]*  
*start on stopped rcS*  
*start on started tty1*  
*stop on runlevel [016]*

Emits ( Eventos que emitirán los trabajos ) : Especifica los eventos que el archivo de trabajo generará.

*Emits virtual-fileystems*  
*Emits filesystem*  
*Emits mounted*

Console ( Salida por Consola ) : Podemos inidicar por donde queremos que salga la salida de nuestros trabajos con la instancia *console*. Output ( entrada y salida desde /dev/console) o none (opción por defecto, entrada y salida a /dev/null)

*exec echo example*  
*console output*

## Ejemplos de Archivos de Configuración

Archivo hostname.conf (configurar el nombre del equipo)

```
$ cat /etc/event.d/hostname.conf
# hostname - set system hostname
#This task is run on startup to set the system hostname from /etc/hostname,
# falling back to "localhost" if that file is not readable or is empty and
# no hostname has yet been set.
description    "set system hostname"
start on startup
task
exec hostname -b -F /etc/hostname
```

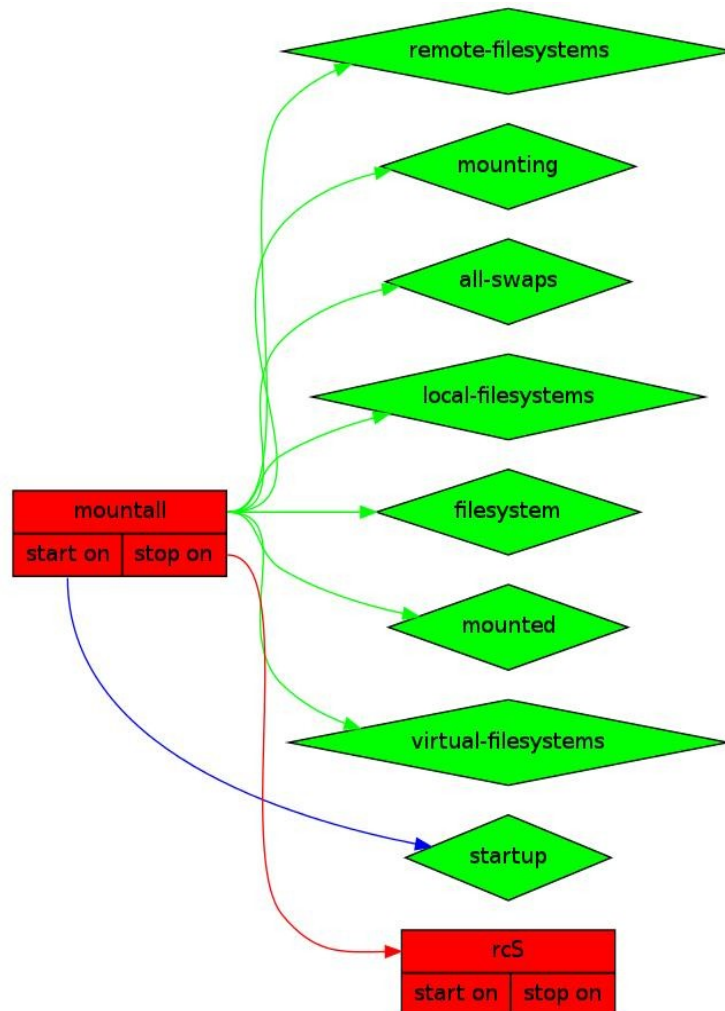
Archivo Mountall.conf ( monta sistemas de archivo y dispositivos)

```
$ cat mountall.conf
# mountall - Mount filesystems on boot
# monta sistemas de archivos en el orden correcto cuando los
# dispositivos y puntos de montaje están disponibles.
description    "Mount filesystems on boot"
start on startup
stop on starting rcS
expect daemon
emits virtual-fileystems
emits local-fileystems
emits remote-fileystems
emits all-swaps
emits filesystem
```



emits mounting  
emits mounted  
[ ... rest of script ]

En Ubuntu Natty (11.04) , Upstart introdujo el comando **initctl2dot**, que permite ver gráficamente estas dependencias.



Generated on 2012-06-22 19:05:55.175183 by initctl2dot  
(subset, from 'initctl --system show-config -e' on host jose-N150-N210-N220).

Boxes of color red denote jobs.  
Solid diamonds of color green denote events.  
Dotted diamonds denote 'glob' events.  
Emits denoted by green lines.  
Start on denoted by blue lines.  
Stop on denoted by red lines.

Las cajas rojas indican trabajos, *start on* es el evento que inicia el trabajo indicado con línea azul (en este caso *startup*), *stop on* en este caso es otro trabajo que detiene al trabajo (*rcS*), *mountall* también emite eventos que pueden o no iniciar otros trabajos, los eventos generados por este trabajo se indican por las líneas verdes.

## Comandos Usados para interactuar con Upstart

<i>Initctl start &lt;trabajo&gt;</i>	Inicia un Trabajo
<i>Initctl stop &lt;trabajo&gt;</i>	Detiene un Trabajo
<i>Initctl emit &lt;evento&gt;</i>	Emite un evento
<i>Initctl list</i>	Lista trabajos Conocidos
<i>Initctl show-config &lt;trabajo&gt;</i>	Muestra Eventos emitidos, y condiciones de inicio y parada del trabajo
<i>Initctl status &lt;trabajo&gt;</i>	Muestra Estado Trabajo
<i>Initctl version</i>	Muestra Versión Demonio Init

### Deshabilitar Un Trabajo del Inicio Automático:

Desde Upstart 0.6.7 , para impedir que Upstart inicie automáticamente un trabajo se puede

- Renombrar el archivo de configuración de trabajo de manera tal que no finalice en la extensión “.conf”.
- Editar el archivo de configuración de trabajo y comentar la sentencia de inicio de evento *start on* usando el carácter “#”.

## Diferencias Demonio SystemV Init y Upstart

### Beneficios del Sistema SystemV Init:

- Simplicidad: Crear archivos de servicios es fácil ya que son simples scripts de shell. Para habilitar/deshabilitar un servicio en un nivel de ejecución, solo se tiene que crear/quitar un enlace simbólico en un directorio particular o conjunto de directorios (rcn.d).
- Orden Garantizado de Servicios: Ejecutando los scripts apuntados por los enlaces simbólicos en secuencia numérica. Servicios numerados bajos se ejecutan antes que aquellos con números altos.

## Limitaciones del Sistema System V:

- **Performance No Optimizada:** El sistema de inicio secuencial tradicional fue apropiado para el tiempo que se inventó, pero para los estándares modernos es “lento” en el sentido que no hace uso de paralelismo sino de sincronismo. Fue diseñado para ser simple y eficiente para el manejo de los administradores. Sin embargo este modelo no hace un uso completo de los recursos de los sistemas modernos, particularmente el que múltiples servicios pueden ejecutarse simultáneamente.

Como hemos visto, El demonio `init` actual (primer proceso del sistema y padre de todos los procesos) se basa en el del UNIX System V, por lo que se le conoce como `sysvinit`. Los sistemas operativos que siguen este esquema clasifican las tareas en diferentes “run levels” o niveles de ejecución, y se lanzan una serie de servicios y demonios dependiendo del nivel de ejecución (o arranque) que hayamos seleccionado. Ejecutar una serie fijada de scripts, uno tras otro, en un determinado orden ha funcionado razonablemente bien hasta ahora. No obstante, conforme Linux ha ido mejorando y adaptándose a los sistemas modernos este enfoque ha empezado a plantear problemas.

La antigua solución (demonio `Init`) sólo funcionaba si garantizábamos que en determinados instantes de la secuencia de arranque están disponibles ciertos recursos, por ello para que los scripts de `init` funcionen se deben ejecutar en puntos concretos de dicha secuencia.

Normalmente se ordenan teniendo en cuenta que

- Los discos duros deben haberse encontrado, iniciado y haber detectado sus particiones antes de que intentemos montarlas desde `/etc/fstab`.
- Los dispositivos de red deben haberse detectado e iniciado antes de que activemos los servicios de red

Esto funcionaba hace diez años, ¿por qué ahora no? La respuesta breve es que nuestras computadoras se han vuelto mucho más flexibles:

- Los dispositivos pueden enchufarse y desenchufarse en cualquier momento, e.j. dispositivos USB.
- Los buses de almacenamiento aceptan un número variable de dispositivos, por lo que hay que explorar el bus. Esta operación no debe ser bloqueante.
- Para reducir el consumo eléctrico, las unidades de disco duro pueden dejar de girar hasta que se explore el bus, por lo que tardarán un tiempo mayor en aparecer.
- Los dispositivos de red pueden enchufarse y desenchufarse en cualquier instante.

Se necesitaba un sistema `init` que pudiera re ordenar dinámicamente la secuencia de inicio basándose en la configuración y el hardware disponible en cada situación.

### Beneficios del Sistema de Arranque Upstart :

- Arranque de servicios en paralelo (asíncrono)
- Guiado por eventos
- Servicios que se arrancan de forma automática si mueren
- Reducción del tiempo de arranque y apagado
- Reducción tiempo de carga del entorno gráfico

## Conclusión

Hemos visto una pequeña introducción a Upstart, sobre todo lo que tiene que ver con la compatibilidad con el sistema Sys V Init, lo que nos lleva a pensar que la migración todavía no ha sido completa, Upstart emplea los scripts de niveles de ejecución rc <nivel de ejecución>. Respecto al tiempo de inicio de un equipo la diferencia no es muy significativa, quizás porque la migración entre un sistema y otro no ha sido completada. Se podrán esperar mayores ganancias de velocidad tan pronto los scripts de arranque individuales se hayan adaptado al nuevo sistema. Pero las posibilidades son mucho más extensas que el solo hecho de un aumento de velocidad en el inicio, por ejemplo en lo que a sistema de gestión dinámica se refiere para gestionar que ha de pasar cuando una portátil está funcionando con batería o cuando se conecta a la corriente, simplemente habría que añadir archivos de configuración de trabajos en /etc/init que gestionasen dichos eventos. La filosofía orientada a eventos puede aplicarse a cambios en el hardware o en el software del equipo lo que llevará a un dinamismo de los sistemas que reaccionarán a estos cambios con algún suceso o trabajo, estos trabajos podrán llevarse a cabo en paralelismo a otros lo que supondrá un ahorro de tiempo y un sistema mas veloz y dinámico. Habrá que esperar en un futuro no muy lejano a que Upstart migre completamente y no dependa mas del antiguo sistema para ver su potencialidad.

## Bibliografía

1. Upstart Wikipedia <http://en.wikipedia.org/wiki/Upstart>
2. Upstart Cookbook [http://upstart.ubuntu.com/cookbook/upstart\\_cookbook.pdf](http://upstart.ubuntu.com/cookbook/upstart_cookbook.pdf)
3. Upstart Home Page <http://upstart.ubuntu.com/>
4. Como Inicia Ubuntu <https://help.ubuntu.com/community/UbuntuBootupHowto>
5. Upstart Linked Packages, <https://launchpad.net/upstart/+packages>
6. Proceso de Arranque en Linux,  
[http://es.wikipedia.org/wiki/Proceso\\_de\\_arranque\\_en\\_Linux](http://es.wikipedia.org/wiki/Proceso_de_arranque_en_Linux)
7. Linux Boot Camp, <http://www.linuxplanet.com/linuxplanet/tutorials/7019/1>
8. Faster Booting With Upstart, <http://www.h-online.com/open/features/Faster-booting-with-Upstart-848690.html>
9. The Upstart Startup Manager <http://www.linuxplanet.com/linuxplanet/tutorials/7033/2>
10. Where to find the inittab in ubuntu Linux <http://backdrift.org/where-to-find-the-inittab-in-ubuntu-linux>
11. Curso Administración GNU/Linux Nivel I <http://cursos.gugler.com.ar/>