

# Sistemas de inicio Unix

Comprender los distintos tipos de arranque en los sistemas Unix y sus posteriores derivaciones nos acerca a muchos conceptos vistos a lo largo del curso, principalmente al de procesos y al desarrollo histórico de la computación. No solo es importante por proporcionarnos una visión más amplia del funcionamiento de un sistema operativo, sino también por su vigencia en los años recientes y al papel que han tomado en muchas de las principales distribuciones de Linux.

## Sistemas de inicio

Un sistema de inicio es el software que, cuando arrancas tu sistema, es responsable de cargar los drivers apropiados, activar tu conexión de red, ejecutar varios servicios, y finalmente brindar la pantalla gráfica en la que accedes al sistema.

## Las dos tradiciones de Unix

Desde los inicios de Unix, este se conoció por ser un sistema altamente portable y modificable. Esto provocó que cualquier persona pudiera desarrollar su propia versión del sistema. Este hecho fue el principal propulsor de versiones como System V o BSD. Cada uno basado en un mismo núcleo pero con distintas características como la localización de los binarios, comandos, o el sistema de inicio, del cual se tratará a continuación.

## El inicio al estilo SysV vs el inicio al estilo BSD

El sistema de SysV se basa en niveles de ejecución (runlevels). Sus archivos de configuración se encuentran en `/etc/rc.d/`, este contiene los scripts utilizados por el comando `/sbin/init`. Junto a `rc.d` se pueden ver directorios como `rc0.d`, `rc1.d`, `rc2.d`, ..., `rc6.d`. Cada uno correspondiente a un nivel de ejecución configurado por defecto, los cuales son:

- 0 — Halt
- 1 — Single-user text mode
- 2 — Not used (user-definable)
- 3 — Full multi-user text mode
- 4 — Not used (user-definable)
- 5 — Full multi-user graphical mode (with an X-based login screen)
- 6 — Reboot

Los niveles de ejecución determinan qué servicios son detenidos o iniciados por el comando `init`, por ejemplo el nivel 1, detiene cualquiera de los servicios de red, mientras que el nivel 3 los inicia.

El sistema de inicio al estilo BSD es mucho más sencillo, simplemente consiste en ejecutar el script de inicialización `/etc/rc`, de una manera muy similar a como se hacía en el Unix original de Bell Labs, no existen los niveles de ejecución.

## Upstart

Es el remplazo del tradicional “`init`” en System V. Provee muchas de las facilidades que el sistema “`init`”, pero tiene varias características en las que intentó mejorarlo.

Upstart fue creado debido a limitaciones fundamentales en dos sistemas existentes:

- Sistema `init` de System V
- Sistemas `init` basados en dependencias

## Las razones históricas

### SysV, lo bueno:

- Simplicidad: crear archivos de servicio es fácil con el `init` de SystemV, ya que simplemente son *shell scripts*. Para habilitar o deshabilitar un servicio en algún nivel de arranque en particular, solo es necesario crear o eliminar una liga simbólica en un directorio en particular o un conjunto de directorios.
- Pedido de servicios garantizado: Esto lo logra el `init` ejecutando los scripts a los que hacen referencia las ligas simbólicas secuencialmente. El orden en el cual `init` invoca estos scripts está determinado por un elemento numérico en el nombre: los servicios con números más pequeños se ejecutan antes que los servicios con números altos.

### SysV, las limitaciones:

- Sin un rendimiento óptimo: El inicio del sistema secuencial era apropiado para los tiempos en que fue inventado, pero para los estándares modernos es “lento” en el sentido de que no usa el paralelismo.
- Enfocado al servidor: En los días en que un sistema Unix con cientos de usuarios concurrentes, donde los reinicios eran raros, la propuesta de SysV era perfecta. Si un hardware necesitaba ser remplazado se programaba un apagado del sistema, se llevaba a cabo, el nuevo hardware era instalado y el sistema era nuevamente puesto en línea. Sin embargo, el mundo ha cambiado. Desde la perspectiva de Ubuntu una gran cantidad de usuarios ejecuta la versión de escritorio en dispositivos portátiles que podrían ser reiniciados varias veces al día.

- Asume un hardware estático todo el tiempo: El tradicional init de SysV es incapaz de manejar un cambio dinámico en el hardware del sistema.

## El diseño de Upstart

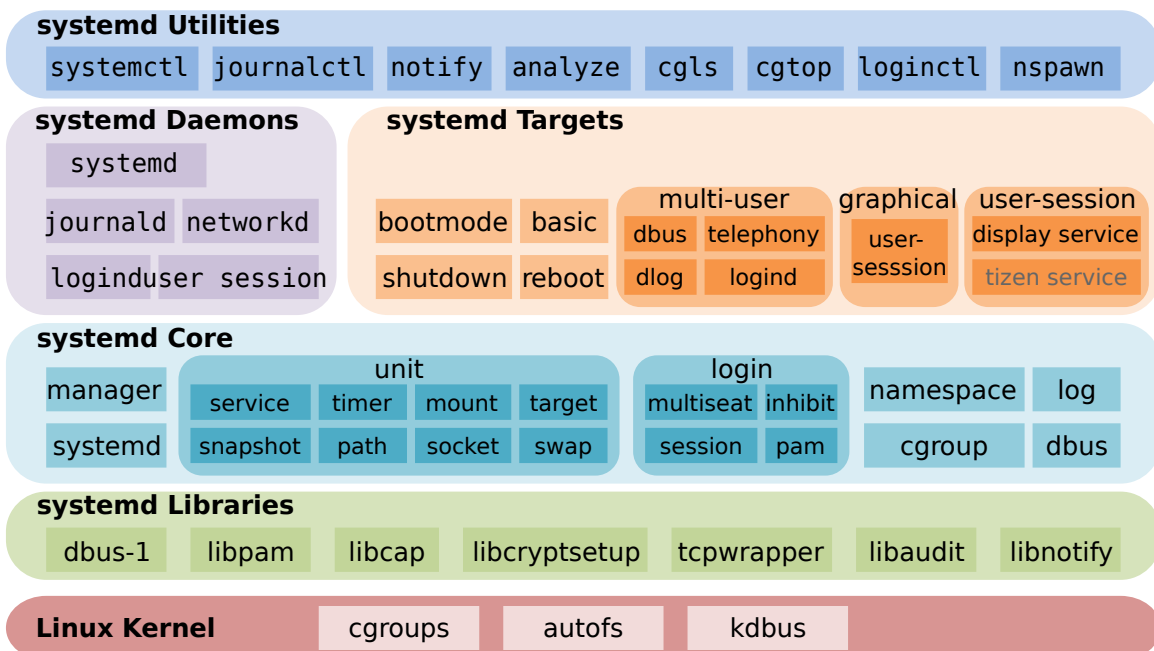
Upstart maneja la asincronicidad emitiendo “eventos” que pueden ser registrados por servicios. Cuando un evento –o combinación de eventos– que satisface los requerimientos de algunos servicios es emitido, Upstart automáticamente iniciará o detendrá ese servicio. Si múltiples trabajos tienen la misma condición de “inicio”, Upstart iniciará esos trabajos en “paralelo”. Wikipedia dice acerca del diseño de Upstart:

Upstart trabaja de forma asíncrona supervisando las tareas mientras el sistema esta arrancado. También gestiona las tareas y servicios de inicio cuando el sistema arranca y los detiene cuando el sistema se apaga. La fácil transición y la perfecta retrocompatibilidad con sysvinit fueron objetivos explícitos en el diseño. Por lo tanto, Upstart es capaz de ejecutar scripts de sysvinit sin modificaciones. De esta manera se diferencia de la mayoría de reemplazos de init, que normalmente requieren una transición completa para funcionar correctamente y no son compatibles con un entorno mixto formado por métodos de arranque tradicionales y nuevos.

Upstart fue propuesto originalmente hacia Ubuntu, varios sistemas lo adoptaron, pero en los últimos años se ha observado una migración, incluso de Ubuntu, hacia Systemd.

## Systemd

La mayoría de las principales distribuciones de Linux han adoptado como sistema de inicio por defecto a Systemd, este más que ser solamente un sistema de inicio es un conjunto de demonios que se encargan de una gran variedad de características del sistema. Se describe a sí mismo como “un



conjunto de demonios de administración de sistema, bibliotecas y herramientas diseñados como una plataforma de administración y configuración central para interactuar con el núcleo del Sistema operativo GNU/Linux.” En el proceso de arranque es el proceso con el PID 1 y de él dependen todos los demás procesos. En 2011 Fedora fue la primer distribución de Linux en adoptarlo como sistema de inicio por defecto.

## Controversias

Amancio, en el sitio Lignux, escribió:

SystemD no es simple ni cumple una única función. Es un programa muy complejo formado por diversos demonios que, lejos de limitarse a las funciones de init, está absorbiendo cada vez más funciones, procesos y demonios. Por ejemplo, el control de energía, manejo de puntos de montaje, encriptación de discos, configuración de red, etc. Miremos más atentamente el manejo de puntos de montaje. Éste proceso está controlado por udev que es el gestor de dispositivos del kernel Linux Su función es controlar los ficheros de dispositivo en /dev. Sin embargo, udev ha pasado a estar integrado en SystemD. Todos los programas que necesiten a udev como dependencia, ahora también dependen de Systemd.

Otro problema es el manejo de journal, un sistema de registro de incidencias que permite solucionar fácilmente fallos del sistema. SystemD ha modificado la forma en que trabaja respecto a opciones anteriores, haciéndola más compleja. Maneja journal mediante código binario fácilmente corruptible, además de otras problemáticas.

Todo esto tiene implicaciones preocupantes. Como el programa es cada vez más complejo, adquiere cada vez más código que puede fallar y colapsar al propio SystemD. Pero, recordemos, SystemD tiene PID 1. Si SystemD colapsa, también colapsa todo el sistema operativo. Peor aún es si SystemD colapsa debido a procesos que si fuesen independientes, no tendrían jamás PID 1 (y al colapsar no causaría daños tan graves). Por ejemplo, que por fallar udev colapse SystemD, y por tanto también el sistema entero.

De igual forma es cada vez más difícil o complejo poder revisar el programa para pulirlo y detectar deficiencias.

Pero, decíamos que SystemD tiene una perspectiva de “control unificado” por así decir, sobre el sistema (como acabamos de ver), abarcando cada vez más y más procesos. Pero por otro lado, también tiene una perspectiva unificadora sobre el ecosistema GNU/Linux: el objetivo es eliminar “diferencias sin sentido entre distribuciones”. Esto es repulsivo para muchas personas que precisamente, algo que aman del software libre en general y de GNU y Linux en particular, es que puede modificarse tanto como sea necesario. De hecho, no sólo diferentes pueden querer sistemas operativos que trabajen de diferentes maneras según los intereses de esas personas; incluso una misma persona puede querer usar diversos sistemas operativos que se ajusten a necesidades o gustos diversos.

Ante esto y otras muchas críticas hacía Systemd, ha seguido avanzando y posicionándose cada vez mejor, tanto en decisiones unilaterales como con Canonical, así como por parte de la comunidad, como en Arch Linux.

Los sistemas de inicio son hoy en día no un mero conjunto de scripts que configuran de una sola vez el sistema, sino todo un conjunto de herramientas que reaccionan de manera dinámica a las acciones del usuario, las cuales vale la pena estudiar y comprender, ya que pueden tener una gran repercusión en los temas de estabilidad del sistema operativo, seguridad, sistemas de archivos, puntos de montaje, privacidad e incluso a la cultura y filosofía misma de Unix y los sistemas con los que interactuamos todo el tiempo.

## Fuentes

---

1. <https://danielmiessler.com/blog/the-differences-between-bsd-and-system-v-unix/>
2. <http://nereida.deioc.ull.es/~pcgull/ihiu01/cdrom/unix/unix1/contenido/node3.html>
3. <http://www.faqs.org/docs/artu/historychapter.html>
4. <http://so-grupo3.blogspot.mx/2007/05/historia-de-unix.html>
5. <https://unix.stackexchange.com/questions/5877/what-are-the-pros-cons-of-upstart-and-systemd>
6. <http://upstart.ubuntu.com/cookbook/#what-is-upstart>
7. Gran parte de las secciones “Razones históricas” y “El diseño de Upstart” fueron tomados, traducidos y adaptados de <http://upstart.ubuntu.com/cookbook/#what-is-upstart>
8. <http://0pointer.de/blog/projects/why.html>
9. <https://www.pcworld.com/article/2841873/meet-systemd-the-controversial-project-taking-over-a-linux-distro-near-you.html>
10. <http://blog.jorgenschaef.de/2014/07/why-systemd.html>
11. <https://www.centos.org/docs/rhel-rg-en-3/s1-boot-init-shutdown-sysv.html>
12. <https://lignux.com/la-polemica-systemd/>
13. <https://www.linuxito.com/gnu-linux/nivel-alto/431-por-que-systemd-es-una-mierda>