

## SYSTEMD vs INIT

Desde su aparición en el panorama tecnológico (alrededor del año 2010), **systemd** ha ido aumentando su aceptación entre los constructores de distribuciones GNU/Linux en detrimento del histórico **init** y las alternativas a él que han ido apareciendo en el tiempo. Las distribuciones predominantes ya se han pasado a **systemd** adoptándolo como mecanismo de inicialización por defecto, hablamos de [Debian](#), [Fedora](#), Centos, ubuntu, OpenSuSe, Arch, RHEL, etc... Teniendo en cuenta que estas distros son la base de un enorme número de “derivadas”, la presencia de **systemd** es una realidad.

Hay que insistir en que **systemd** no es la única alternativa a **init** pero sí la más aceptada como su relevo. Un administrador de sistemas no debe olvidar que hay distribuciones (pocas) que utilizan algunas de esas alternativas por defecto o que, al menos, están disponibles en sus repositorios de software a disposición del sysadmin para su instalación y uso. La tabla siguiente da una idea de este asunto sin que aparezcan en ella TODAS las posibilidades:

Feature	Init system									
	sysvinit	OpenRC	upstart	systemd	SMF	launchd	Epoch	finit	runit	BSD rc.d
Supported platforms	Linux / BSD	Linux + BSD	Linux	Linux	Solaris	MacOSX	Linux	Linux	Linux / BSD / MacOSX	BSD
Main coding language	C	POSIX shell (+ C)	C	C	C	C	C	C	C	POSIX shell (+ C)
Main dependencies	-	init (sysv or BSD)	D-Bus	D-Bus	init(sysv?)	-	libc, /bin/sh	?	-	rcorder
Init script/service format	single config file	shell scripts	config files + shell fragments	config files (ini)	XML (+ shell scripts)	plist	multiple or single .conf	multiple or single .conf	shell scripts	shell scripts
Per-service configuration	✗ No	✓ Yes (conf.d)	✓ With in-job mechanisms (normal exit, respawn)	✓ Yes	✓ Yes (service instances)	?	✓ Yes (v1.1+)	?	✗ No	✗ No
Running as a daemon	✓ Yes (PID 1)	✓ Yes (PID 1)	✓ Yes (PID 1)	✓ Yes (PID 1)	✓ Yes (invoked)	✓ Yes (PID 1)	✓ Yes (PID 1)	✓ Yes (PID 1)	✓ Yes (PID 1)	✓ Yes (PID 1)
Cross-service dependencies/events	✗ No	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✗ No	✓ Yes (basic support, v1.3+)	?	✓ Yes	✓ Yes
Parallel service startup	✓ Yes	✓ Yes (optional)	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✗ No	✓ Yes	✓ Yes	✓ Yes
Keeping daemons alive	✓ Yes	✓ Yes (optional v0.21+ via supervise-daemon)	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes	✓ Yes
Preferred service file supplier	n/a	Gentoo	Ubuntu	upstream	Solaris	MacOSX	n/a	n/a	Void Linux	NetBSD, FreeBSD, OpenBSD
License	GPL v2+	2-cl. BSD	GPL v2	LGPL v2.1+	?	Apache License 2.0	Unlicense	MIT	BSD	BSD

A partir de aquí, sin entrar en extensos detalles, vamos a explicar las diferencias entre los dos mecanismos más extendidos con el objeto de responder a las siguientes 4 preguntas:

1. ¿Qué es **init**?
2. ¿Qué es **systemd**?
3. ¿Porqué **init** debe ser reemplazado?
4. ¿Cuáles son las características de **systemd**?

## ¿Qué es **init**?

En GNU/Linux, **init** es la abreviatura de inicialización. **init** es el proceso residente (que se ejecuta en segundo plano, en inglés background) que se inicia inmediatamente después del kernel, se mantiene en funcionamiento hasta el apagado de la máquina y es el padre de todos los demás procesos de forma directa o indirecta, típicamente se le asigna el PID=1. Si de alguna manera no se pudo iniciar, no se iniciará ningún proceso y el sistema alcanzará un estado conocido como "Kernel Panic". **init** es la aportación a las distribuciones GNU/Linux del primer sistema operativo UNIX comercial conocido como **System V**, es por ello que el nombre completo de este proceso es **System V init**. El uso de **init** en la mayoría de las distribuciones GNU/Linux es idéntico al sistema operativo del que procede con algunas excepciones, como Slackware, que utiliza el estilo BSD y Gentoo que utiliza un **init** personalizado. La necesidad de reemplazar **init** con algo más avanzado se identificó hace mucho tiempo, se desarrollaron varias alternativas que siguen presentes pero que no han conseguido el grado de implantación de **systemd**.

## ¿Qué es **systemd**?

**systemd** es el demonio administrador del sistema (System Management Daemon) el nombre se le ha otorgado siguiendo las convenciones UNIX de nominación de procesos, añadiendo al final del nombre una **d** de **daemon** con el objeto de identificarlo fácilmente. En un principio fue liberado bajo licencia *GNU General Public License* pero en la actualidad está amparado por una licencia *GNU Lesser General Public License*. Al igual que **init**, **systemd** es el primer proceso que se ejecuta después del kernel, es el padre de todos los demás procesos de forma directa o indirecta y típicamente se le asigna el PID=1.

Además del demonio en si mismo, el término **systemd** puede referirse a todos los paquetes, utilidades y bibliotecas que forman parte del producto completo. Fue diseñado para superar las deficiencias de **init**. En esencia es un proceso que se ejecuta en segundo plano y cuya misión principal es iniciar otros procesos en paralelo, es esto (iniciar en paralelo) lo que le confiere la mayor de las diferencias respecto a **init** ya que proporciona una reducción en el tiempo de arranque y en la sobrecarga computacional. Además de esta, tiene muchas otras características en comparación con **init**.

## ¿Porqué **init** debe ser reemplazado?

**init** arranca los procesos de forma secuencial, osea, un proceso no se iniciará hasta que el proceso anterior haya terminado con éxito sus etapas de arranque y carga en memoria. Esto, a menudo, genera retrasos y un largo tiempo de inicio. **Systemd** no ha sido diseñado para ser rápido en su desempeño pero hace las cosas de manera que no provoca retrasos innecesarios.

## Mejoras de **systemd**

- Diseño limpio, avanzado y eficiente.
- Inicialización mas sencilla.
- inicio de procesos concurrente y en paralelo.
- Mejor API.
- Sintaxis sencilla en el tratamiento de Unidades (Unit).
- Capacidad de eliminar componentes opcionales.
- Bajo impacto en la producción de memoria residual.
- Técnica mejorada para expresar dependencias.

- Instrucciones de inicialización escritas en un fichero de configuración, no en guiones de script.
- Hace uso de Unix Domain Socket.
- Encolado de procesos usando marcas de tiempo propias (systemd Calendar Timers).
- Registro de eventos con **journald**.
- Elección de registro de eventos del sistema tanto con **systemd** como con **syslog**.
- Los logs son almacenados en ficheros binarios.
- Permanencia en el tiempo del estado de **systemd** para ser invocado en el futuro.
- Seguimiento de procesos usando cgroup del kernel y no PID.
- Administración de conexiones de usuarios con systemd-logind.
- Mejor integración con Gnome en aras de la interoperatividad.

#### Inconvenientes de systemd.

- Todo está en el mismo sitio
- No cumple con el standard POSIX

En la página siguiente se proporciona una tabla comparativa de especificaciones técnicas entre **init** y **systemd**.

Features	init	systemd
DBus Dependency Mandatory -	No	Yes
Device based Activation	No	Yes
Device dependency configuration with udev	No	Yes
Timer based Activation	Cron/at	Proprietary
Quota Management	No	Yes
Automatic Service Dependency Handling	No	Yes
Kills users Process at logout	No	Yes
Swap Management	No	Yes
SELinux integration	No	Yes
Support for Encrypted HDD	No	Yes
Static kernel module loading	No	Yes
GUI	No	Yes
List all the child processes	No	Yes
Sysv compatible	Yes	Yes
Interactive booting	No	Yes
Portable to non x86	Yes	No
Adopted on	Several Distro	Several Distro
Parallel service startup	No	Yes
Resource limit per service	No	Yes
Easy extensible startup script	Yes	No
Separate Code and Configuration File	Yes	No
Automatic dependency calculation	No	Yes
Verbose debug	Yes	No
Version	N/A	V44+
Size	560 KB	N/A
Number of Files	75 files	900 files + glib + DBus
Lines of code - LOC	15000 (Approx)	224000 (Approx) (inc Codes, comments and white space) 125000 (Approx) (acctual code)