INGENIERÍA DE SERVIDORES (2016-2017)
GRADO EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

Comparativa entre systemd e init.

18 de diciembre de 2016

Resumen

En el mundo de *UNIX* donde el kernel está público y cada vez es más grande la comunidad que lo desarrolla, aparecen nuevas contribuciones para intentar mejorar el sistema. En nuestro caso vamos a analizar una nueva implementación del "proceso 1" que rápidamente ha sido adaptada por las principales distribuciones de *UNIX*, `systemd`, sustituyendo al proceso `init`. Explicaremos en que consiste tanto `systemd` como `init` con sus ventajas e inconvenientes, además examinaremos la implementación de ambos para saber un poco más sobre como funcionan. Por último haremos una conclusión final dando una valoración sobre estos procesos aplicados en el mundo de los servidores.

1. Introducción

Como se indica en la historia de The Open Group [4], los sistemas de tipo *UNIX*, concedidos a principios de la década de los 70, se idearon de forma gratuita y libre, fomentando la idea de software libre y de código abierto. Creando así la comunidad tan grande y amplia que conocemos hoy en día. Gracias a esta comunidad se ha seguido desarrollando y avanzando el sistema *UNIX* hasta el kernel de *Linux* que conocemos y utilizamos a diario. Constantemente se están desarrollando nuevas tecnologías y procesos para mejorar este sistema.

Nosotros nos vamos a centrar en comparar una nueva forma de uno de los procesos principales del sistema con su predecesor que curiosamente se ha mantenido casi intacto desde la década de los noventa, es decir, desde hace más de 20 años no ha sido modificado, hecho sorprendente pensando en la velocidad a la que avanza siempre la tecnología. El proceso en cuestión es el encargado del inicio del sistema, es el proceso con PID 1 (Proccess ID). El arranque del sistema funciona con un proceso que va ejecutando todos los demás procesos, como los drivers del adaptador de red o del ratón, los controladores de pantalla, etc.

Este proceso era `init` que era utilizado por todas las distribuciones basadas en Linux. Pero Lennart Poettering, como redacta en su articulo [1], y Kay Sievers decidieron desarrollar un nuevo proceso de arranque del sistema, desarrollando así `systemd`, el encargado de iniciar prácticamente todas las distribuciones de Linux actuales.

Aunque ambos sistemas son daemons (demonios) hay multitud de diferencias entre ellos, que se explicarán mas adelante y finalmente se hará una conclusión de ambos sistemas de arranque para el mundo de los servidores.

2. Proceso init

El proceso `init` se caracteriza por su simpleza y facilidad de uso. El funcionamiento de `init` consiste en ir iniciando los procesos listados en un archivo de configuración, es decir, inicia el primer proceso del listado y cuando éste se ha iniciado inicia el siguiente y así sucesivamente. De esta forma tan simple es como `init` inicia el sistema. Además utiliza la filosofía del software libre de ser transparente para el usuario y permitirle poder modificar por completo su implementación, ya que basta con modificar el fichero de configuración. De esta forma el proceso `init` siempre es el padre o antecesor de todo proceso existente en el sistema y además adopta cualquier proceso que pudiera quedarse sin un padre por el motivo que fuera.

Sin embargo presenta unos claros inconvenientes derivados de su propia naturaleza simple y sencilla. El primer inconveniente es el control de las dependencias, ya que deben controlarse por parte del programador y tiene que tener conocimientos de las dependencias que tienen los procesos y listar primero los procesos que no tienen dependencias o cuyas dependencias ya se han resuelto, por ejemplo si en el listado de procesos aparece un proceso apache que depende de la configuración de

red este proceso se iniciará con errores o no se iniciará puesto que el controlador de red todavía no se ha iniciado, aún cuando tanto apache como el controlador de red estén perfectamente configurados. El otro inconveniente principal, sobre todo en servidores, es la sobrecarga al inicio del sistema, ya que se inician los procesos de uno en uno y hasta que uno no acabe de arrancar no se inicia el siguiente.

2.1. System V

Tal como se dice en el artículo de Jonas Gorauskas en "Linux Jornal" [5], `System V` fue una versión mejorada del `init` original y es la que ha estado presente todo este tiempo en las distintas distribuciones de `Linux`. Aunque esta versión de `init` sigue utilizando su estructura monolítica y por tanto mantiene el inconveniente del control de dependencia, añadía un sistema de llamadas por niveles de prioridad para la ejecución de los procesos del sistema. Este proceso de llamadas se realiza mediante un esquema de directorios con niveles de ejecución que contienen los scripts de arranque de los servicios. Dependiendo de la distribución de `Linux` se utilizan más o menos niveles de prioridad de ejecución.

2.1.1. Problemas de System V

System V presenta una serie de problemas debido a como fue concebido, ya que se pensó para equipos estáticos y que mantenían su hardware, por ello se desarrollo de forma estática y síncrona para el arranque y apagado del sistema, dando lugar a bloqueos de tareas futuras hasta que las actuales no fueran completadas. Esto deja al sistema sin poder reaccionar ante algunos eventos que no estaban programados para el inicio o apagado del sistema. Además no había un control sobre los demonios una vez ejecutados, si estos tenían cualquier problema durante su ejecución no eran tratados por nadie, como mucho si se quedaban huérfanos los recogía el proceso `init`, pero si se paraban no volvía a lanzarlos aunque fueran procesos importantes o vitales para el sistema.

3. Proceso systemd

Como ya hemos dicho anteriormente `systemd` fue desarrollado por Lennart Poettering y Kay Sievers, ambos empleados de Red Hat. El proceso `systemd` tal como la definen los autores en la web oficial de `systemd` [3] es una suite o conjunto de herramientas diseñadas para ofrecer una funcionalidad específica, en este caso para facilitar y mejorar el arranque del sistema operativo *Linux*. Específicamente es un conjunto de demonios de *Linux*, un demonio [2] es un proceso que funciona en segundo plano en el sistema a la espera de eventos o llamadas que se producen en el sistema y cuando son despertados realizan una tarea concreta.

3.1. Funcionamiento

3.2. Implementación

4. Conclusiones

5. Conclusiones

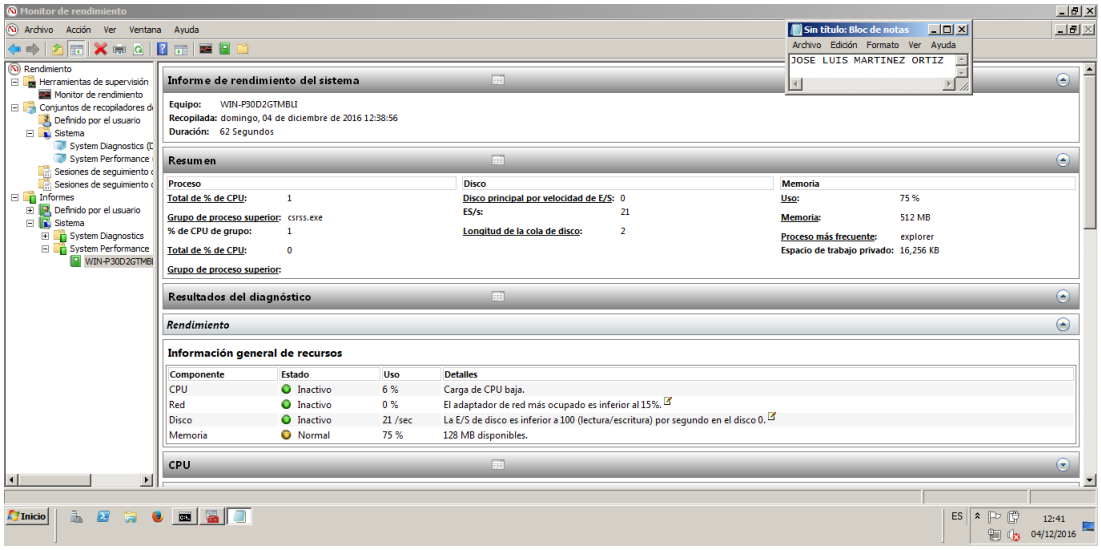

Figura 5.1: Informe del monitor de rendimiento - resumen

Referencias

[1] http://0pointer.de/blog/projects/systemd.html#faqs, consultado el 14 de Diciembre de 2016. Blog de Lennart.

[2] https://wiki.archlinux.org/index.php/daemons, consultado el 14 de Diciembre de 2016. ArchWiki, apartado sobre daemons.

[3] https://www.freedesktop.org/wiki/Software/systemd/, consultado el 14 de Diciembre de 2016. Web oficial de systemd.

[4] http://www.unix.org/what_is_unix/history_timeline.html, consultado el 14 de Diciembre de 2016. The Open Group, UNIX.

[5] Jonas Gorauskas. http://www.linuxjournal.com/content/initializing-and-managing-services-linux-past-present-and-future?page=0,0, consultado el 14 de Diciembre de 2016. Articulo de “Linux Journal”.