Tútulo: El software libre en las grandes organizaciones. Distribuciones: mantenimiento, soporte y seguridad. Casos reales.

Tema:

Autores:
Rubén Rubio de la Oliva

Rafael Grimán Canto

Empresa: Novell

Contenido:

Introducción
Concepto de distribución
- Qué es una distribución.
- Tipos de distribuciones (criterios: licencias, soporte o no por empresas, etc.)
- Valor añadido de una distribución
Soporte y mantenimiento
Modelos de soporte y mantenimiento del software libre.
- Servicios de soporte y de mantenimiento disponibles en el mercado
-
Seguridad
Gestión de puestos
Colaboración

Más otros temas relativos a virus, parches, actualizaciones, versiones de alto nivel de seguridad (SEL), la certificación de la seguridad del software libre, productos de swl estrella en seguridad, etc.

El software libre en las grandes organizaciones. Distribuciones: mantenimiento, soporte y seguridad. Casos reales

Introducción

En general se reconoce que el movimiento del código abierto y Linux ha cambiado para siempre el paisaje de la industria de TI. En muy poco tiempo Linux se ha convertido en uno de los sistemas operativos con mayor crecimiento en la industria de TI.

- Según la empresa de investigación de mercado IDC, se prevé que el empleo de Linux crezca a un índice anual compuesto del 14% entre 2002 y 2007.
- Esta tendencia es confirmada por Forrester Research, quien afirma que el 72% de 50 empresas entrevistadas con un valor de más de mil millones de dólares prevé aumentar su utilización de Linux en los próximos dos años. La encuesta de Forrester también muestra que más de la mitad de las empresas entrevistadas proyecta reemplazar otros sistemas operativos con Linux.
- Según el CIO Research, el 64% de 375 empresas entrevistadas está utilizando código abierto y una de las razones clave de esto es el menor coste total de propiedad, TCO. Los responsables de IT dicen que los principales beneficios de usar código abierto son un menor importe total de propiedad, menor inversión de capital y mayor fiabilidad y tiempo de servicio en comparación con los sistemas existentes

Existe una larga lista de beneficios asociada a la utilización de software de código abierto, como Linux:

- Capacidad de ejecución en una gran variedad de plataformas hardware
- Reducción importante en los costes
- Alta calidad, escalabilidad, fiabilidad y disponibilidad
- Reducción en los costes de operación
- Abundancia de personal con experiencia y formación
- Conjunto de aplicaciones y herramientas en continuo crecimiento

EL modelo de desarrollo seguido para desarrollar de código abierto supone un cambio muy grande en la cadena de valor total del software. El principio y fin de esta cadena de valor pueden ser el código de más bajo nivel asociado a los microprocesadores y la adaptación o configuración que se pueda realizar de cualquier aplicación para el cliente final. En medio ha aparecido un elemento productivo llamado “la comunidad”, que hasta el momento ha producido unos 133 millones de líneas de código. Una comunidad no es más que un grupo de personas que se “juntan” (puede ser físicamente o mediante Internet) para desarrollar un software determinado. Existen comunidades de desarrollo de drivers, entornos gráficos, juegos, emuladores, iconos, sonidos, etc. Los integrantes de las comunidades de desarrollo tienen una serie de características comunes:

- una meta clara
- desarrollan porque les gusta
- suelen ser muy jóvenes
- desarrollan sin un horario definido
- desarrollan lo que necesitan, es software hecho por la comunidad para la comunidad, es decir, no crean la necesidad, desarrollan lo que necesitan en ese momento
- desarrollan para aprender y mejorar

- son MUY buenos desarrolladores
- tienen una inquietud muy grande por avanzar
- valoran el trabajo bien hecho más que “la fecha de entrega”

El modelo de desarrollo de una comunidad puede parecer anárquico, caótico o desordenado. Parece que esto no puede llevar a buen puerto ningún desarrollo de software y que no puede haber calidad en el propio software, pero esto no es cierto ¿Por qué no es cierto? Por varias razones:

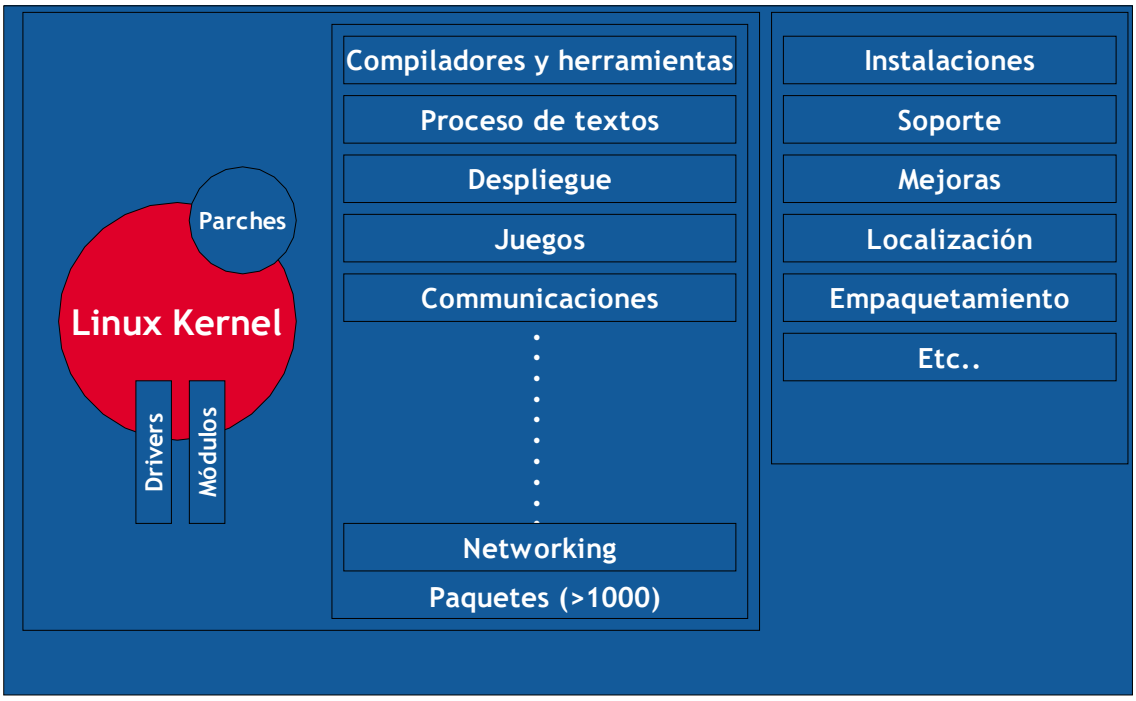
- no es desordenado ya que generalmente el que empieza un proyecto lo lidera, por lo que no hay anarquía. Si el líder se cansa, se aburre o se va, pronto sale alguien que retoma ese liderazgo o bien se vota un nuevo líder
- el líder generalmente lo que hace es decidir qué mejoras o cambios se introducen en el software que está desarrollando la comunidad. ¿Qué criterio se sigue para esto? Depende de la comunidad: calidad de las mejoras, problemas que resuelve, prioridades de la comunidad, etc.
- las tareas se desarrollan por iniciativa por lo que se trabaja en lo que se cree y en lo que gusta, de ahí que algunos consideren a las comunidades como sectas
- Existe por tanto un control de configuración y camino de desarrollo aunque no está sometido a las reglas de desarrollo del software tradicional, como mercado, rentabilidad, intereses corporativos, etc.

Distribución

Las comunidades no suelen tener relación entre ellas y desarrollan componentes de muy distinto tipo, como drivers, iconos, utilidades de línea de comandos, herramientas de desarrollo, editores o sistemas operativos como Linux. Si se quiere tener un conjunto homogéneo de componentes que aporten un valor completo a usuarios finales se hace necesaria una labor de recopilación y empaquetado. En lo que se refiere a sistemas operativos existen principalmente dos tipos:

- si se basan en un kernel Linux y usan principalmente herramientas GNU es una distribución de Linux
- si se basan en el kernel de BSD y usan herramientas principalmente bajo licencia BSD lo que desarrollan es un sistema operativo BSD

Ejemplo de componentes en una distribución



Tipos de Distribuciones

Como siempre, al ser humano le da por clasificar las cosas, como las distribuciones de Linux no podían ser menos, nos hemos dedicado a clasificarlas. Se pueden clasificar de diversas formas:

- comerciales y de comunidad
- específicas y generalistas
- ...

Nos quedaremos con la clasificación *comercial* y de *comunidad*.

Distribuciones de Comunidad

Las distribuciones de comunidad son aquellas que se desarrollan, mantienen y gestionan mediante una comunidad sin que haya una empresa que lo organice todo. En estas distribuciones puede haber alguna empresa implicada, pero simplemente colabora con:

- dinero
- software
- hardware
- patrocinan eventos
- ...

pero no gestionan nada ni tienen el liderazgo. Es la comunidad la que decide qué se hace y qué no, qué entra y qué no, cuándo sale una versión nueva y cuándo no.

Posiblemente la distribución de comunidad más famosa y conocida es Debian, aunque no la única. Otras, también muy conocidas, como Linex o Knoppix son derivadas de Debian.

Distribuciones “hechas”

Debian es una distribución muy organizada y estructurada en cuanto a sus integrantes. No es sencillo entrar a formar parte de la comunidad de desarrolladores de Debian, se vota cualquier decisión y tienen unas guías muy estrictas en cuanto a las licencias del código que forma parte de la distribución.

Slackware fue de las primeras distribuciones en aparecer y sigue muy de cerca la filosofía Unix, de hecho el sistema para administrar el software es mediante paquetes tgz o tar.gz, igual que en los Unix comerciales. Tiene muchos menos seguidores, pero son muy leales.

Distribuciones “do it yourself”

Otra distribución de comunidad muy respetada es **Gentoo**. Muy buena y sumamente configurable debido a que se compila todo desde cero de forma que el usuario puede decidir desde qué opciones vamos a pasarle al compilador para obtener el máximo rendimiento de nuestro equipo hasta qué opciones queremos que tenga una aplicación determinada. Una ventaja es que trae una serie de herramientas que permiten automatizar esta tarea de creación de nuestra propia distribución.

LFS (Linux From Scratch) es muy similar a la anterior ya que se compila desde cero, pero se diferencia en que no hay herramientas que automatiza la tarea de creación de nuestra propia distribución por lo que es completamente manual, y en que necesitamos un Linux previamente instalado para poder desarrollar nuestro propio LFS.

Como principales ventajas podemos resaltar

- muy estables, la calidad del software es muy alta
- muy personalizables, se pueden afinar hasta límites casi insospechados
- los seguidores son muy fieles por lo que desarrollan un software de muy buena calidad
- consumen pocos recursos
- desarrollo más rápido
- requiere tener grandes conocimientos

Algunos inconvenientes pueden ser

- no tienen un roadmap definido
- no están certificadas por fabricantes de hardware y/o software
- los seguidores son muy fieles por lo que llegan a ser intolerantes
- se requiere mucho tiempo para tenerla instalada tal y como queremos
- no tienen soporte garantizado

Distribuciones Comerciales

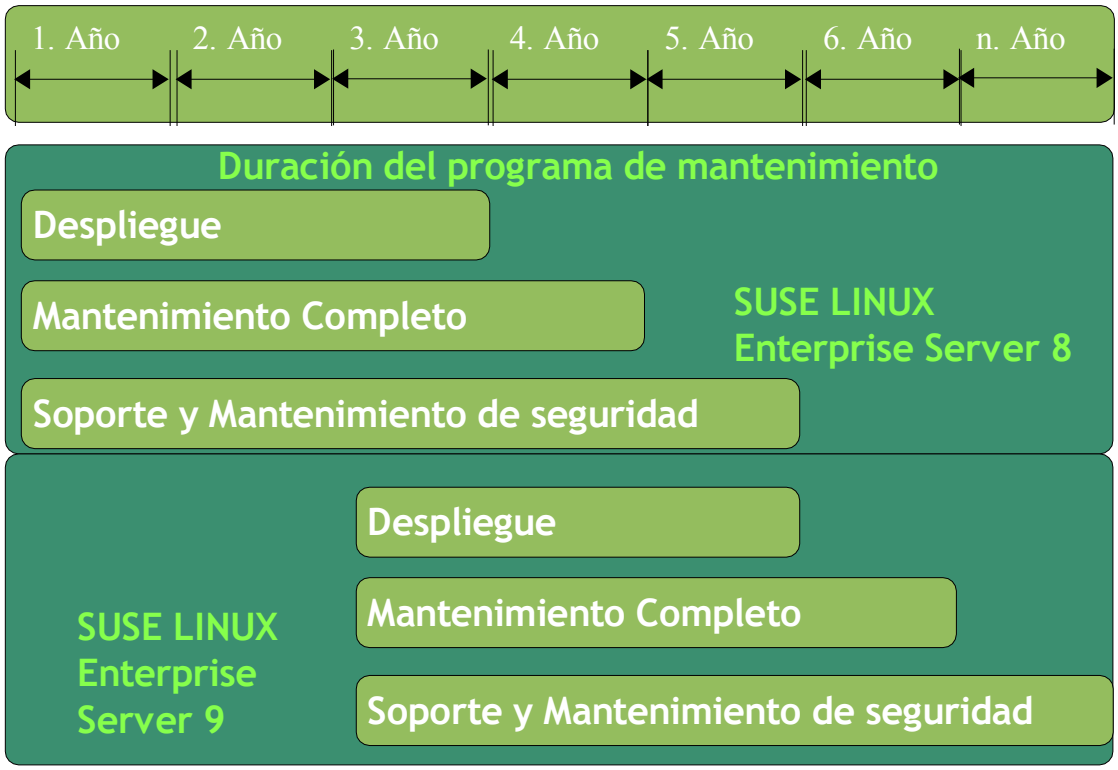
Una distribución comercial es aquella que se gestiona, mantiene y desarrolla enteramente con una empresa detrás. El desarrollo, la gestión, el mantenimiento y las decisiones se llevan a cabo en y desde la propia empresa.

En este caso existe una comunidad que es la comunidad de usuarios que usan dicha distribución. Estos usuarios no suelen intervenir en el desarrollo ni en las decisiones de sobre la distribución. Algunos usuarios sí desarrollan algunos paquetes y participan activamente en las listas de correo dando su opinión sobre la distribución.

Las distribuciones comerciales más famosas son:

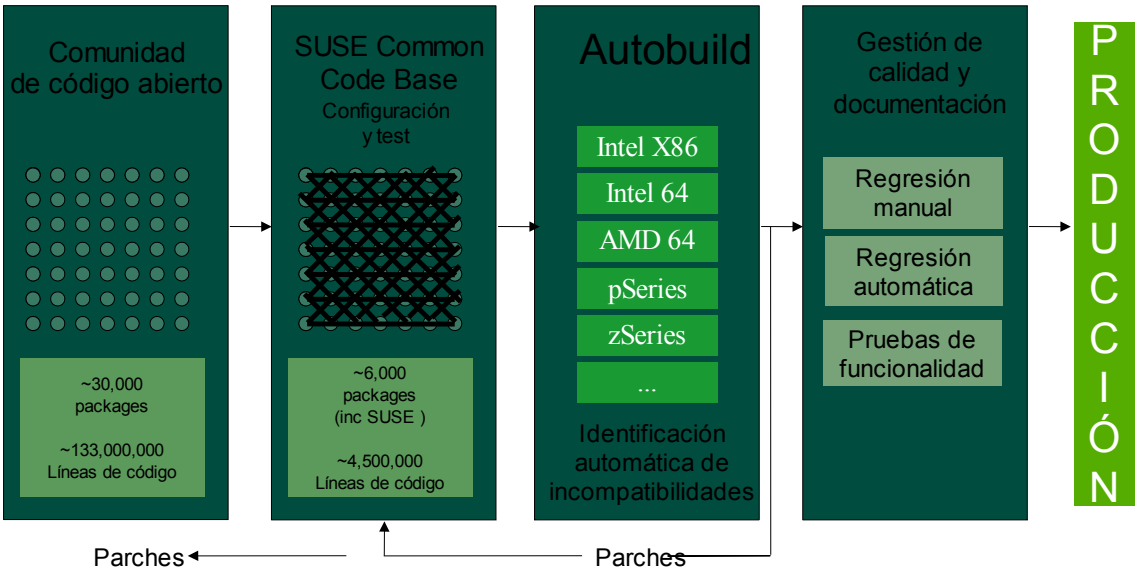
- **SUSE**, distribución alemana en origen, de muy buena calidad. La empresa que la empezó a desarrollar (SUSE Linux AG) ha sido adquirida recientemente por Novell. Muy estable y fiable, con una herramienta (YaST) que nos permite administrar el equipo, desde el software hasta el hardware, pasando por los usuarios mediante una interfaz gráfica. Sigue muy de cerca los estándares definidos: LSB, FHS, ...
- **Red Hat**, distribución de origen Norte Americano, con mucha difusión y muy conocida. Presenta herramientas para administrar el sistema, pero no están agrupadas bajo una interfaz gráfica.
- **Mandrake**, distribución de origen francés, basada en RedHat. Muy orientada al usuario final por lo que presenta una herramienta de administración muy sencilla de utilizar llamada DrakConf y es muy sencilla de utilizar. Tiene poca difusión.

La siguiente figura muestra los ciclos de vida y evolución de las versiones de la distribución SUSE.



La distribuciones comerciales suelen estar orientadas al mercado empresarial que tiene ciertas necesidades específicas, como mantenimientos, soporte, certificaciones de hardware y software de uso común compatibilidad hacia atrás, estabilidad de las versiones y presencia en el mercado, roadmaps definidos, etc. Las distribuciones comerciales aportan a las organizaciones el valor de extraer y mantener el software producido por la comunidad y hacerlo utilizable por la organización adaptándolo a sus necesidades.

La siguiente figura muestra la cadena de producción de la distribución SUSE



Elegir una distribución u otra depende de una serie de factores:

- conocimientos que tengamos
- tiempo del que disponemos
- necesidad de soporte
- necesidad de certificaciones
- flexibilidad y personalización buscada
- si somos organización o particular
- necesidad de roadmaps definidos
- velocidad de desarrollo
- recursos de los que disponemos
- ganas de intervenir en el desarrollo de la distribución

Una organización siempre se verá beneficiada de una distribución comercial, más que nada por el soporte, las certificaciones y los roadmaps bien definidos. También hay que tener en cuenta que puede intervenir en el desarrollo de distribuciones comerciales gracias a acuerdos comunes.

Seguridad

El mundo del SW es un mundo que crece muy deprisa y además es uno de los niveles a tener en cuanta en el mundo de la seguridad. Igual que hay compañias de desarrollo de SW ofimático que no dejan ver su código fuente para evitar que la gente “les robe sus ideas”, existen empresas de desarrollo de SW de seguirdad (cortafuegos, antivirus, ...) que opinan igual.

La opinión de estas empresas de SW de seguridad es igualmente respetable que la de cualquier otra empresa o individuo y tiene una postura lógica. Es decir, si yo oculto el código fuente:

- la gente que quiera ver ese código fuente para poder buscar agujeros de seguridad, no lo podrá hacer

Este tipo de pensamiento se conoce como seguridad mediante oscuridad. Pero esto trae una serie de dudas:

- ¿cómo sé yo, como usuario, que dicha empresa no ha incluido código malicioso?
- ¿cómo sé yo que el SW que desarrolla esa empresa es realmente seguro? O, lo que es lo mismo, ¿cómo puedo auditar yo el SW de seguridad que voy a usar?
- ¿qué ocurre si yo encuentro un problema de seguridad al usar ese SW?
- ¿qué ocurre si otra persona encuentra un fallo de seguridad?

Como podemos ver, el ocultar el código fuente no supone más seguridad. Muchos critican que si se muestra el código fuente, efectivamente hay mucha gente auditando ese código para hacerlo más seguro pero que también hay gente auditándolo para aprovecharse de sus vulnerabilidades. La cuestión es, ¿no lo hacen ya, aunque ese código esté oculto?

Tengamos en cuenta que ocultar el código no impide que alguien sepa cómo funciona ese SW, hay muchas técnicas que le permitirán saber cómo funciona:

- ingeniería inversa
- ensayo y error
- ...

Es más, se lleva ocultando el código fuente durante mucho tiempo y esto no ha impedido que el SW siga siendo vulnerable y que haya ataques informáticos.

Seguridad en el SW abierto

Abrir el código fuente no sólo permite dotar al SW de mayor seguridad sino que ofrece una tranquilidad mayor al usuario ya que puede:

- auditar el código (o pedir que lo auditen) para ver si realmente es seguro: al disponer del código fuente uno puede auditar el SW que va a utilizar y decidir si está a la altura de sus expectativas. En el caso de no saber auditar código, siempre puede acudir a otra persona, colectivo o empresa para que lo hagan ellos bien sea pagando o no
- añadir mejoras y mejorar el SW: si un matemático desarrolla una ecuación nueva que mejora un algoritmo de cifrado, siempre podrá añadirlo al código fuente y mejorar el SW que está utilizando. Dichas mejoras ayudarán al resto de usuarios también ya que el código se incluiría en el SW original
- eliminar problemas que haya encontrado: ya que podemos acceder al código fuente, siempre podremos revisarlo/auditarlo para detectar problemas y corregirlos
- aumentar el rendimiento: si se desarrolla un nuevo algoritmo más rápido o eficiente, siempre se podrá incluir en el código fuente del SW para que todo el mundo que use dicho SW salga beneficiado
- velocidad de desarrollo: tenemos que tener en cuenta, además, que no hay un sólo matemático, criptógrafo, ... que está mirano el código sino muchos, recordemos que el SW abierto se basa en el trabajo en equipo. Esto da lugar a un desarrollo mucho más rápido y eficiente

En cuanto a la tranquilidad que se le ofrece al usuario, debemos hacer hincapié en que el usuario podrá saber si el desarrollador (sea particular o empresa) ha incluido código malicioso que, por ejemplo, le puede:

- robar datos sensibles: número de la tarjeta de crédito, número de la cuenta bancaria, ...
- conocer información personal y hacer mal uso de ella: si un personaje ilustre tienen alguna enfermedad, alguien puede robar su historial médico, publicar dicha información y arruinarle la vida a dica persona
- molestar: enviar correo basura, pornografía, ...
- ...

Gestión de recursos en red

La creciente presencia de recursos de IT, hardware y software, en toda la organización y en la mayoría de los casos en localizaciones dispersas, genera una dificultad en la gestión de los mismos, que normalmente se resuelve con aplicaciones específicas de gestión de puestos de trabajo o de gestión de servidores. La inclusión de sistemas de código abierto y más concretamente de Linux complica más la gestión de recursos, ya que introduce un elemento de complejidad adicional, con una nueva manera de escribir, probar y empaquetar el software.

Los factores adicionales de complejidad son:

- Resolución de dependencias: Casi todos los paquetes que se instalan tiene dependencias de otros componentes dentro del sistema.
- Ancho de banda: las organizaciones con mucha distribución geográfica dispondrán en muchos casos de conexiones con poco ancho de banda.
- Diversidad de aplicaciones y frecuencia de la actualizaciones, especialmente en parches de seguridad y actualizaciones.

La solución a esta problemática está en una solución integrada que permita la gestión de una plataforma heterogénea, incluido Linux, de una manera centralizada.
El entorno Linux es muy dinámico. La posibilidad de revisión del código y la comunicación que existe entre todos los miembros de la comunidad en el mundo hace que cualquier incidencia de seguridad o de otro tipo sea abiertamente expuesta y revisada por todos, comunidad y también usuarios, generándose actualizaciones y parches de manera continua.
Las distribuciones son un primer escalón de aprovisionamiento de estas mejoras a las organizaciones, proporcionando componentes testados y asegurando que pueden ser distribuidos y utilizados de manera segura.
En otros casos se pueden tener varias distribuciones en un mismo entorno y además paquetes procedentes de desarrollo interno.
El proceso completo de actualización de software consta normalmente, de las siguientes fases:

- Obtención de actualizaciones, tanto externas como internas
- Instalar estas actualizaciones en un entorno de prueba
- Organización y gestión del software durante y después del proceso de pruebas
- Distribuir el software ya probado al entorno de producción

Novell Red Carpet es un producto especializado en la gestión de entornos Linux, desarrollado por Ximian, que ha sido integrado en el entorno ZENworks. El producto resultante reúne la funcionalidad necesaria para gestionar un entorno heterogéneo, de software propietario y de software libre que incluye servidores, puestos de trabajo y dispositivos móviles. Algunas de las funcionalidades con que cuenta en lo relativo al software libre son resolución de dependencias de paquetes, resolución de conflictos en la aplicación de parches y gestión automática de estos, permitiendo la integración con

productos como SUSE Linux YAST o Red Hat Nerwork, rollback de parches e instalaciones y distribución e instalación.