Explicando BSD

Resumen

En el mundo del código abierto, la palabra "Linux" es casi un sinónimo de "Sistema Operativo", pero no es el único sistema operativo de código abierto UNIX®.

¿Así que cuál es el secreto? ¿Por qué BSD no es más conocido? Este artículo trata sobre esa y otras cuestiones.

A lo largo de este documento, las diferencias entre BSD y Linux se *mostrarán en cursiva*.

Tabla de contenidos

1. ¿Qué es BSD? 1
2. ¿Qué, un verdadero UNIX®? 2
3. ¿Por qué BSD no es más conocido? 3
4. Comparando BSD y Linux 3

1. ¿Qué es BSD?

BSD significa "Berkeley Software Distribution". Es el nombre de las distribuciones de código fuente de la Universidad de California, Berkeley, que originalmente eran extensiones del sistema operativo UNIX® de AT&T Research. Varios proyectos de sistemas operativos de código abierto tienen su origen en una distribución de éste código conocida como 4.4BSD-Lite. Además, comprenden una serie de paquetes de otros proyectos de código abierto, incluido especialmente el proyecto GNU. El sistema operativo completo incluye:

- El kernel BSD, que se encarga de la programación de procesos, la gestión de la memoria, el multiprocesamiento simétrico (SMP), los controladores de dispositivos, etc.
- La biblioteca C, la API base del sistema.

  La biblioteca C de BSD está basada en código procedente de Berkeley, no del proyecto GNU.

- Utilidades como shells, utilidades de archivos, compiladores y enlazadores.

  Algunas de las utilidades derivan del proyecto GNU, otras no.

- El sistema X Window, que gestiona el entorno gráfico.

  El sistema X Window utilizado en la mayoría de las versiones de BSD es mantenido por el proyecto X.Org. FreeBSD permite al usuario elegir entre una variedad de entornos de escritorio, como GNOME, KDE o Xfce; y administradores de ventanas ligeros como Openbox, Fluxbox o Awesome.

- Muchos otros programas y utilidades.

2. ¿Qué, un verdadero UNIX®?

Los sistemas operativos BSD no son clones, sino derivados de código abierto del sistema operativo AT&T's Research UNIX®, el cual es a su vez ancestro del moderno UNIX® System V. Esto puede sorprenderle. ¿Cómo puede haber sucedido si AT&T jamás liberó su código?

Cierto es que AT&T UNIX® no es código abierto y que en sentido estricto del copyright BSD no es *en absoluto* UNIX®, pero por otra parte AT&T incluyó código fuente de otros proyectos, teniendo como caso notable el Computer Sciences Research Group (CSRG) de la Universidad de California en Berkeley, CA. A partir de 1976, el CSRG comenzó a lanzar cintas de su software, llamándolas *Berkeley Software Distribution* o *BSD*.

Las versiones iniciales de BSD consistían principalmente en programas de usuario, pero cambió drásticamente cuando el CSRG consiguió un contrato con el Defense Advanced Research Projects Agency (DARPA) para actualizar los protocolos de comunicaciones en su red, ARPANET. Los nuevos protocolos fueron conocidos como *Internet Protocols*, más adelante *TCP/IP* y más tarde se covertirían en los protocolos más importantes. La primera implementación ampliamente distribuida fue parte de 4.2BSD, en 1982.

Durante la década de los 80 comienzan a surgir compañías que ofrecían estaciones de trabajo. Muchas preferían obtener licencias de UNIX® en lugar de desarrollar sistemas operativos por sí mismas. En particular, Sun Microsystems adquirió una licencia de UNIX® e implementó una versión de 4.2BSD, que denominaron SunOS™. Cuando la propia AT&T fue autorizada para vender UNIX® comercialmente, iniciaron una implementación un tanto rudimentaria llamada System III, seguida rápidamente por System V. El código base de System V no incluía el soporte de red, por lo que todas las implementaciones incluían software adicional de BSD, incluido el software TCP/IP, pero también utilidades como el shell *csh* y el editor *vi*. En conjunto, estas mejoras se conocían como las *Berkeley Extensions*.

Las cintas BSD contenían código fuente de AT&T y en consecuencia requerían una licencia de código UNIX®. Hacia 1990, la financiación del CSRG se estaba agotando y se enfrentaba al cierre. Algunos miembros del grupo decidieron liberar el código BSD, que era Código Abierto, sin el código propiedad de AT&T. Finalmente esto sucede con la *Networking Tape 2*, más conocida como *Net/2*. Net/2 no era un sistema operativo completo: faltaba aproximadamente un 20% del código del kernel. Uno de los miembros del CSRG, William F. Jolitz, escribió el código restante y lo distribuyó a principios de 1992 como *386BSD*. Al mismo tiempo, otro grupo de antiguos miembros del CSRG fundaron una empresa llamada Berkeley Software Design Inc. y distribuyó una versión beta de un sistema operativo llamado BSD/386, que se basó en las mismas fuentes. El sistema operativo pasó a denominarse BSD/OS.

386BSD jamás llegó a ser un sistema operativo estable. En cambio, otros dos proyectos se separaron de él en 1993: NetBSD y FreeBSD. Ambos proyectos se forman gracias a la falta de paciencia que origina la espera de mejoras en 386BSD: La gente de NetBSD comenzó a principios de año, y la primera versión de FreeBSD no estuvo lista hasta final de año. En ese proceso el código base tomó caminos diferentes hasta tal punto que se hizo difícil de fusionar. Además, los proyectos tenían diferentes objetivos, como veremos más adelante. En 1996, OpenBSD se separó de NetBSD, y en

2003, DragonFlyBSD se separó de FreeBSD.

3. ¿Por qué BSD no es más conocido?

Por varias razones, BSD es relativamente desconocido:

1. Los desarrolladores de BSD a menudo están más interesados en mejorar su código que en comercializarlo.
2. Gran parte de la popularidad de Linux se debe a factores externos a los proyectos de Linux, como la prensa y las empresas formadas para proporcionar servicios de Linux. Hasta hace poco, los BSD de código abierto no tenían tales partidarios.
3. En 1992 AT&T denunció a BSDI, el distribuidor de BSD/386, alegando que el producto contenía código propiedad de AT&T. El caso se resolvió fuera de los tribunales en 1994, pero el espectro del litigio sigue atormentando a las personas. En marzo de 2000, un artículo publicado en la web afirmaba que el caso judicial había sido "resuelto recientemente".

   Un detalle que el proceso judicial aclaró fue el de la nomenclatura: en la década de los 80, BSD era conocido como "BSD UNIX®". Con la eliminación del último vestigio del código AT&T de BSD, también perdió el derecho a llamarse UNIX®. Por lo tanto, es posible encontrar títulos de libros referentes a "the 4.3BSD UNIX® operating system" y "the 4.4BSD operating system".

4. Comparando BSD y Linux

Entonces, ¿cuál es realmente la diferencia entre, digamos, Debian Linux y FreeBSD? Para el usuario promedio, la diferencia es sorprendentemente pequeña: Ambos son sistemas operativos tipo UNIX®. Ambos están desarrollados por proyectos no comerciales (esto no se aplica a muchas otras distribuciones de Linux, por supuesto). En la siguiente sección, veremos BSD y lo compararemos con Linux. La descripción se ajusta más a FreeBSD, que representa aproximadamente el 80% de las instalaciones de BSD, pero las diferencias entre NetBSD, OpenBSD y DragonFlyBSD son pequeñas.

4.1. ¿Quién es dueño de BSD?

Ninguna persona o empresa es dueña de BSD. Es creado y distribuido por una comunidad de colaboradores altamente cualificados y comprometidos a lo largo y ancho del mundo. Algunos de los componentes de BSD son proyectos de código abierto con sus propios derechos que se administran y mantienen por separado.

4.2. ¿Cómo se desarrolla y actualiza BSD?

Los kernel de BSD se desarrollan y actualizan siguiendo el modelo de desarrollo de código abierto. Cada proyecto mantiene un *árbol de código fuente* de acceso público que contiene todos los archivos del proyecto, incluida la documentación y otros ficheros relacionados. Los usuarios pueden obtener una copia completa de cualquier versión.

Un gran número de desarrolladores de todo el mundo contribuyen a mejorar BSD. Se dividen en tres tipos:

- Los *Contributors* escriben código o documentación. No se les permite hacer commit (añadir código) directamente al árbol de fuentes. Para que su código sea incluido en el sistema debe ser revisado y probado por un desarrollador registrado, conocido como *committer*.
- Los *Committers* son desarrolladores que disponen de acceso de escritura en el árbol de fuentes. Para convertirse en committer es necesario demostrar habilidad en el área en la cual él o ella trabaja.

  Depende del criterio individual de cada committer cuándo pedir autorización antes de hacer cambios en el árbol de fuentes. En general, un committer experimentado puede realizar cambios que obviamente son correctos sin obtener consenso. Por ejemplo, un committer que trabaje en un proyecto de documentación puede corregir errores tipográficos o gramaticales sin necesidad de revisión. Por otra parte, se espera que los desarrolladores que pretendan realizar cambios de gran calado o complicados envíen sus cambios para que sean revisados antes de ser incluidos. En casos extremos un miembro del core team con una función como la del Principal Architect puede ordenar que los cambios sean retirados del árbol, es lo que llamamos *backing out*. Todos los committers reciben un correo electrónico que describe cada commit, por lo que no es posible hacer un commit en secreto.

- El *Core team*. Tanto FreeBSD como NetBSD disponen de un core team que administra el proyecto. Los core teams dirigen el rumbo de los proyectos pero sus funciones no siempre están claras. No es necesario ser desarrollador para ser un miembro del core team pero suele ser lo habitual. Las normas del core team varían de un proyecto a otro, pero en general tienen más voz en la dirección del proyecto que los miembros del equipo que no son del Core.

Esta gestión difiere de la de Linux en algunos aspectos:

1. Ninguna persona controla el contenido del sistema. En la práctica eso es muy relativo, ya que el Principal Architect puede requerir que el código sea retirado, e incluso en el proyecto Linux a ciertas personas les está permitido hacer cambios.
2. Por otro lado, *hay* un repositorio central, un único lugar donde se pueden encontrar las fuentes de todo el sistema operativo, incluidas todas las versiones anteriores.
3. Los proyectos BSD mantienen todo el "Sistema Operativo", no solo el kernel. Esta distinción tiene una utilidad marginal: ni BSD ni Linux son útiles sin aplicaciones. Las aplicaciones utilizadas en BSD son con frecuencia las mismas que las utilizadas en Linux.
4. Como resultado del mantenimiento estructurado de un único árbol de fuentes SVN, el desarrollo de BSD es limpio y es posible acceder a cualquier versión del sistema por su número de versión o por la fecha. SVN también permite actualizaciones incrementales del sistema: por ejemplo, el repositorio de FreeBSD se actualiza aproximadamente 100 veces al día. La mayoría de estos cambios son pequeños.

4.3. Versiones de BSD

FreeBSD, NetBSD y OpenBSD proporcionan el sistema en tres "versiones" diferentes. Al igual que Linux, a los lanzamientos se les asigna un número como 1.4.1 ó 3.5. Además, el número de versión tiene un sufijo que indica su propósito

1. La versión de desarrollo del sistema recibe el nombre de *CURRENT*. FreeBSD asigna un número

a CURRENT, por ejemplo FreeBSD 5.0-CURRENT. NetBSD utiliza un sistema ligeramente diferente y añade un sufijo compuesto por una única letra que indica cambios en las interfaces internas, por ejemplo NetBSD 1.4.3G. OpenBSD no asigna ningún número ("OpenBSD-current"). Todo nuevo desarrollo en el sistema entra en esta rama.

2. A intervalos regulares, entre dos y cuatro veces al año, los proyectos liberan una versión *RELEASE* del sistema, que está disponible en CD-ROM y mediante FTP para su descarga gratuita, por ejemplo OpenBSD 2.6-RELEASE o NetBSD 1.4-RELEASE. La versión RELEASE está dirigida al usuario final y es la versión estándar del sistema. NetBSD también dispone de *patch releases* que incluyen un tercer dígito, como por ejemplo NetBSD 1.4.2.
3. A medida que se van encontrando errores en la versión RELEASE, se corrigen y las correcciones se agregan al árbol del SVN. En FreeBSD la versión resultante se denomina versión *STABLE*, mientras que en NetBSD y OpenBSD continúa siendo la versión RELEASE. También se pueden agregar nuevas funcionalidades más pequeñas a esta rama después de un período de prueba en la rama CURRENT. La seguridad y otras correcciones importantes de errores también se aplican en todas las versiones soportadas de RELEASE.

Linux, en cambio, mantiene dos árboles de código separados: la versión estable y la versión de desarrollo. Las versiones estables añaden un número par de versión, como 2.0, 2.2 ó 2.4. Las versiones de desarrollo añaden un número impar, como en 2.1, 2.3 ó 2.5. En ambos casos a ese número se le añade otro más que indica la versión exacta. Además, cada proveedor agrega sus propios programas de usuario y utilidades, por lo que el nombre de la distribución también es importante. Cada distribuidor además asigna números de versión a la distribución, así pues la descripción completa podría ser algo como "TurboLinux 6.0 with kernel 2.2.14"

4.4. ¿Qué versiones de BSD están disponibles?

A diferencia de las numerosas distribuciones de Linux, solo hay cuatro BSDs principales de código abierto. Cada proyecto BSD mantiene su propio árbol de fuentes y su propio kernel. En la práctica, sin embargo, parece haber menos diferencias entre el código de usuario de los proyectos que en Linux.

Es difícil categorizar los objetivos de cada proyecto: las diferencias son muy subjetivas. Básicamente,

- FreeBSD tiene como meta ofrecer un alto rendimiento y facilidad de uso al usuario final, y es uno de los favoritos de los proveedores de contenido web. Se ejecuta en varias plataformas y tiene significativamente más usuarios que los otros proyectos.
- NetBSD tiene como meta la máxima portabilidad: "of course it runs NetBSD". Se ejecuta en máquinas que abarcan desde PDAs hasta grandes servidores, e incluso se ha utilizado en misiones espaciales de la NASA. Es una opción particularmente buena para ejecutar en hardware antiguo que no sea Intel®.
- OpenBSD tiene como meta la seguridad y la integridad del código: combina del concepto de código abierto y una revisión rigurosa del código que dan como fruto un sistema muy correcto, elegido por instituciones preocupadas por la seguridad como bancos, bolsas de valores y departamentos gubernamentales de los EEUU. Al igual que NetBSD funciona en gran variedad de plataformas.

- DragonFlyBSD tiene como meta ofrecer un alto rendimiento y escalabilidad bajo cualquier entorno, desde computadoras de un solo usuario hasta enormes sistemas de clústeres. DragonFlyBSD tiene varios objetivos técnicos a largo plazo, pero el desarrollo se centra en ofrecer una infraestructura habilitada para SMP que sea fácil de entender, mantener y desarrollar.

También hay dos sistemas operativos BSD UNIX® adicionales que no son de código abierto, BSD/OS y Mac OS® X de Apple:

- BSD/OS fue el derivado más antiguo de 4.4BSD. No era de código abierto, aunque las licencias de código fuente estaban disponibles a un precio relativamente bajo. Se parecía a FreeBSD en muchos aspectos. Dos años después de la adquisición de BSDi por Wind River Systems, BSD/OS no logró sobrevivir como un producto independiente. El soporte y el código fuente podrían estar todavía disponibles por parte de Wind River, pero todos los desarrollos nuevos se centran en el sistema operativo embebido VxWorks.
- Mac OS® X es la última versión del sistema operativo para la línea Mac® de Apple®. El núcleo BSD de este sistema operativo, Darwin, está disponible libremente como sistema operativo de código abierto totalmente funcional para arquitecturas x86 y PPC. Sin embargo, el sistema gráfico Aqua/Quartz y otros aspectos muy característicos de Mac OS® X siguen siendo de código cerrado. Varios desarrolladores de Darwin son también committers de FreeBSD y viceversa.

4.5. ¿En qué se diferencia la licencia BSD de la licencia pública GNU?

Linux está disponible bajo la GNU General Public License (GPL), que está diseñada para eliminar el software de código cerrado. En particular, cualquier trabajo derivado de un producto publicado bajo la GPL también debe suministrarse con el código fuente si se solicita. Por el contrario, la licencia BSD es menos restrictiva: permite la distribución en forma exclusivamente binaria. Esto es particularmente atractivo para aplicaciones encastradas.

4.6. ¿Qué más debo saber?

Como hay menos aplicaciones disponibles para BSD que para Linux, los desarrolladores de BSD crearon un paquete de compatibilidad con Linux, que permite que los programas de Linux se ejecuten bajo BSD. El paquete incluye modificaciones del kernel para realizar correctamente las llamadas al sistema Linux y archivos de compatibilidad con Linux, como la biblioteca de C. No hay una diferencia notable en la velocidad de ejecución entre una aplicación Linux que se ejecuta en una máquina Linux y una aplicación Linux que se ejecuta en una máquina BSD de la misma velocidad.

El modelo "todo del mismo proveedor" de BSD implica que las actualizaciones son mucho más sencillas de gestionar de lo que son con frecuencia en Linux. BSD maneja las actualizaciones de versiones de bibliotecas suministrando módulos de compatibilidad para versiones anteriores, de modo que es posible ejecutar binarios con varios años de antigüedad sin problemas.

4.7. ¿Qué debo usar, BSD o Linux?

¿Qué significa todo esto en la práctica? ¿Quién debería usar BSD, quién debería usar Linux?

Esta es una pregunta muy difícil de responder. Aquí hay algunas pautas:

- "Si no está roto no lo arregles": Si ya usa un sistema operativo de código abierto y está satisfecho con él, probablemente no haya ninguna buena razón para cambiar.
- Los sistemas BSD, especialmente FreeBSD, pueden proporcionar un rendimiento notablemente superior que Linux, pero esto no es una ley inmutable. En muchos casos no hay diferencia de rendimiento o esta es muy pequeña. En algunos casos Linux podría tener un rendimiento mejor que FreeBSD.
- En general los sistemas BSD gozan de una mejor reputación en cuanto a confiabilidad, principalmente por la mayor madurez de su código base.
- Los proyectos BSD tienen una mejor reputación por la calidad y la integridad de su documentación. Los diversos proyectos de documentación tienen como objetivo proporcionar documentación actualizada activamente, en muchos idiomas, y que cubra todos los aspectos del sistema.
- La licencia BSD puede resultar más atractiva que la GPL.
- BSD puede ejecutar la mayoría de los binarios de Linux, mientras que Linux no puede ejecutar los binarios de BSD. Muchas implementaciones de BSD también pueden ejecutar binarios desde otros sistemas similares a UNIX®. Como resultado, BSD puede presentar una ruta de migración más fácil desde otros sistemas que Linux.

4.8. ¿Quién proporciona soporte, servicio y formación orientada a BSD?

BSDi / FreeBSD Mall, Inc. ha estado proporcionando contratos de soporte para FreeBSD durante casi una década.

Además, cada uno de los proyectos tiene una lista de consultores para contratar: FreeBSD, NetBSD y OpenBSD.