

Explicando BSD

Resumen

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

Entonces, ¿cuál es el secreto? ¿Por qué BSD no es más conocido? Este documento técnico aborda estas y otras preguntas.

A lo largo de este documento, las diferencias entre BSD y Linux serán señaladas *así*.

Table of Contents

1. ¿Qué es BSD?	1
2. Cómo, ¿Un UNIX® de verdad?	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 fueron originalmente extensiones al sistema operativo UNIX® de investigación de AT&T. Varios proyectos de sistemas operativos open source se basan en una versión de este código fuente conocida como 4.4BSD-Lite. Además, constan de cierto número de paquetes de otros proyectos Open Source, incluyendo especialmente el proyecto GNU. El sistema operativo en general se compone de:

- 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 librería C de BSD está basada en código de Berkeley, no del proyecto GNU.

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

Algunas utilidades se derivan del proyecto GNU, otras no.

- El sistema X Ventanas, que maneja la visualización gráfica.

El sistema X Window usado por la mayoría de las versiones de BSD es mantenido por el [Proyecto X.Org](#). FreeBSD permite al usuario escoger de entre una variedad de entornos de escritorio, tales como Gnome, KDE o Xfce; y gestores de ventanas ligeros como Openbox, Fluxbox o Awesome.

- Muchos otros programas y utilidades.

2. Cómo, ¿Un UNIX® de verdad?

Los sistemas operativos BSD no son clones, sino derivados open source del sistema operativo UNIX® de investigación de AT&T que es también el ancestro del moderno UNIX® System V. Esto podría sorprenderte. ¿Cómo pudo ocurrir esto si AT&T nunca ha liberado su código como open source?

Es cierto que UNIX® de AT&T no es open source y en el sentido del copyright, BSD definitivamente *no* es UNIX®, pero por otro lado, AT&T ha importado las fuentes de otros proyectos, principalmente del Computer Sciences Research Group (CSRG) de la Universidad de California en Berkeley, CA. A comienzos de 1976, el CSRG empezó a liberar cintas con 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 convertirían en los protocolos más importantes. La primera implementación ampliamente distribuida fue parte de 4.2BSD, en 1982.

A lo largo de los años 1980, surgieron unas cuantas empresas de estaciones de trabajo. Muchas prefirieron licenciar UNIX® en lugar de desarrollar sistemas operativos ellos mismos. En particular, Sun Microsystems licenció UNIX® e implementó una versión de 4.2BSD que llamaron SunOS™. Cuando a AT&T se le permitió vender UNIX® comercialmente, empezaron con una implementación algo básica llamada System III, que sería seguida rápidamente por System V. El código base de System V no incluyó código de red, así que todas las implementaciones incluyeron software adicional de BSD, incluyendo el software de TCP/IP, pero también utilidades como el *csh* shell y el editor *vi*. De forma colectiva, estas mejoras fueron conocidas como *Berkeley Extensions*.

Las cintas de BSD contenían código fuente de AT&T y por lo tanto requerían una licencia de fuentes de UNIX®. En 1990, la financiación del CSRG se terminó y se enfrentó a su cierre. Algunos miembros del grupo decidieron liberar el código BSD, que era Open Source, sin el código propietario de AT&T. Esto sucedió finalmente con el *Networking Tape 2*, habitualmente conocido como *Net/2*. Net/2 no era un sistema operativo completo: faltaba alrededor del 20% del código del núcleo. Uno de los miembros del CSRG, William F. Jolitz, escribió el código que faltaba y lo liberó a comienzos de 1992 como *386BSD*. Al mismo tiempo, otro grupo de antiguos miembros del CSRG formó una compañía comercial llamada [Berkeley Software Design Inc.](#) y liberó una versión beta de un sistema operativo llamado [BSD/386](#) que estaba basada en las mismas fuentes. El nombre del sistema operativo se cambió posteriormente a BSD/OS.

386BSD nunca se convirtió en un sistema operativo estable. En su lugar, dos proyectos se separaron de él en 1993: [NetBSD](#) y [FreeBSD](#). Los dos proyectos divergieron originalmente debido a diferencias en el nivel de paciencia esperando las mejoras de 386BSD: la gente de NetBSD empezó a comienzos de año y la primera versión de FreeBSD no estuvo lista hasta final de año. Mientras tanto, el código base había divergido lo suficiente como para hacer difícil su unión. Además los proyectos tenían distintos objetivos, como veremos abajo. En 1996, [OpenBSD](#) se escindió de NetBSD, y en 2003,

3. ¿Por qué BSD no es más conocido?

Por varias razones, BSD es relativamente desconocido:

1. Los desarrolladores de BSD suelen estar más interesados en pulir 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 a las empresas formadas para proporcionar servicios de Linux. Hasta hace poco, los BSD de código abierto no tenían tales defensores.
3. En 1992, AT&T demandó a [BSDI](#), el fabricante de BSD/386, alegando que el código contenía código con derechos de AT&T. El caso se resolvió en los tribunales en 1994, pero el espectro del litigio continúa persiguiendo a la gente. En marzo de 2000 un artículo publicado en la web declaraba que el caso judicial se había "resuelto recientemente".

Un detalle que clarificó el proceso judicial fue el nombre: en la década de 1980, BSD era conocido como "BSD UNIX®". Con la eliminación de los últimos vestigios de código de AT&T de BSD, también perdió el derecho al nombre UNIX®. Por lo tanto verás referencias en títulos de libros a "el sistema operativo 4.3BSD UNIX®" y "el sistema operativo 4.4.BSD".

4. Comparando BSD y Linux

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

4.1. ¿Quién es dueño de BSD?

Ninguna persona o corporación es propietaria de BSD. Es creado y distribuido por una comunidad de colaboradores altamente técnicos y comprometidos en todo el mundo. Algunos de los componentes de BSD son proyectos de código abierto por derecho propio y gestionados por diferentes encargados de mantenimiento de proyectos.

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.

Una gran cantidad de desarrolladores en todo el mundo contribuyen a las mejoras de BSD. Se

dividen en tres tipos:

- *Contributors (colaboradores)* escriben código o documentación. No tienen permitido hacer commit (añadir código) directamente al árbol de fuentes. Para que su código sea incluido en el sistema, tiene que ser revisado y comprobado por un desarrollador registrado, conocimo como *committer*.
- *Committers* son desarrolladores con permiso de escritura al árbol de fuentes. Para convertirse en un committer, una persona debe mostrar habilidad en el área en el que está activa.

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 disposición se diferencia de Linux en varias formas:

1. Nadie controla el contenido del sistema. En la práctica, esta diferencia está sobrevalorada, ya que el arquitecto principal puede requerir que el código se restituya, e incluso en el proyecto de Linux se permite a varias personas realizar cambios.
2. Por otro lado, *hay* un repositorio central, un único lugar donde puedes encontrar las fuentes del sistema operativo completo, incluyendo todas las versiones antiguas.
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 formalizado de un único árbol de fuentes SVN, el desarrollo de BSD es claro y es posible acceder a cualquier versión del sistema por número de versión o por fecha. SVN también permite actualizaciones incrementales del sistema: por ejemplo, el repositorio de FreeBSD se actualiza unas 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 del sistema que está en desarrollo se llama *CURRENT*. FreeBSD asigna números a *CURRENT*, por ejemplo FreeBSD 5.0-CURRENT. NetBSD utiliza un esquema de nombrado ligeramente diferente y añade un sufijo de una única letra que indica cambios en los interfaces internos, por ejemplo NetBSD 1.4.3G. OpenBSD no asigna un número ("OpenBSD-current"). Todo el desarrollo nuevo del sistema se realiza en esta rama.
2. A intervalos regulares, entre dos y cuatro veces al año, el proyecto saca una versión *RELEASE* del sistema, que está disponible en CD-ROM y libre para descargar desde los sitios FTP, por ejemplo OpenBSD 2.6-RELEASE o NetBSD 1.4-RELEASE. La versión *RELEASE* está pensada para usuarios finales y es la versión normal del sistema. NetBSD también proporciona *patch releases* con un tercer dígito, 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 BSD 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 divergencias 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 aspira a proporcionar alto rendimiento y facilidad de uso para los usuarios finales y es uno de los favoritos para los proveedores de contenido web. Se puede ejecutar en [diferentes plataformas](#) y tiene un número significativo mayor de usuarios que otros proyectos.
- NetBSD aspira a ofrecer máxima portabilidad: "por supuesto ejecuta NetBSD". Se puede ejecutar en máquinas desde palmtops a grandes servidores y ha sido utilizado en misiones espaciales de la NASA. Es una opción particularmente buena para ejecutar en hardware antiguo que no sea Intel®.
- OpenBSD tiene como objetivo la seguridad y la pureza del código: utiliza una combinación del concepto de código abierto y revisiones rigurosas del código para crear un sistema que sea demostrablemente correcto, lo que lo convierte en la elección de organizaciones preocupadas por la seguridad, como bancos, bolsas de valores y departamentos del gobierno de EE. UU. Como

NetBSD, se ejecuta en varias plataformas.

- DragonFlyBSD tiene como objetivo un alto rendimiento y escalabilidad en todo, desde un sistema UP de un solo nodo hasta un sistema masivamente agrupado. DragonFlyBSD tiene varios objetivos técnicos de largo alcance, pero el enfoque radica en proporcionar una infraestructura compatible con SMP que sea fácil de entender, mantener y desarrollar.

Hay también dos sistemas operativos BSD UNIX® adicionales que no son open source, BSD/OS y el Mac OS® X de Apple:

- BSD/OS fue el más antiguo de los derivados 4.4BSD. No era de código abierto, aunque las licencias de código fuente estaban disponibles a un costo 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 aún pueden estar disponibles en Wind River, pero todo el desarrollo nuevo se centra en el sistema operativo integrado 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 como un sistema operativo open source completamente funcional para ordenadores x86 y PPC. El sistema gráfico Aqua/Quartz y otros muchos aspectos propietarios de Mac OS® X sin embargo permanecen cerrados. Varios desarrolladores de Darwin son también committers de FreeBSD y vice versa.

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 integradas.

4.6. ¿Qué más debo saber?

Dado que hay menos aplicaciones disponibles para BSD que 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 de Linux, como la biblioteca 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, en particular FreeBSD, pueden tener un rendimiento notablemente superior al de Linux. Pero esto no es generalizado. En muchos casos, hay poca o ninguna diferencia en el rendimiento. En algunos casos, Linux puede funcionar mejor que FreeBSD.
- En general, los sistemas BSD tienen una mejor reputación de confiabilidad, principalmente como resultado de una base de código más madura.
- Los proyectos BSD gozan de una mejor reputación por la calidad e 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 licencia GPL.
- BSD puede ejecutar la mayoría de binarios de Linux, mientras que Linux no puede ejecutar binarios de BSD. Muchas implementaciones de BSD también pueden ejecutar binarios de otros sistemas tipo UNIX®. Como resultado, BSD podría presentar una ruta de migración desde otros sistemas más sencilla de lo que sería capaz Linux.

4.8. ¿Quién brinda soporte, servicio y capacitación para 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](#).