Proceso de generación de releases en FreeBSD

Murray Stokely <murray@FreeBSD.org >
Revisión: 43184

FreeBSD is a registered trademark of the FreeBSD Foundation.

CVSup is a registered trademark of John D. Polstra.

Intel, Celeron, EtherExpress, i386, i486, Itanium, Pentium, and Xeon are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

XFree86 is a trademark of The XFree86 Project, Inc.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this document, and the FreeBSD Project was aware of the trademark claim, the designations have been followed by the "™" or the "®" symbol.

2013-11-13 por hrs.

Resumen

Este artículo describe la aproximación utilizada por el equipo de ingeniería de productos de FreeBSD para generar releases de calidad y listas para utilizar en entornos de producción. Se detalla la metodología utilizada para generar la release oficial de FreeBSD y se describen las herramientas disponibles para aquellas personas interesadas en generar sus propias releases a medida de sus necesidades, en particular para demostraciones de empresa o para comercializar el producto.

Traducción de Juan F. Rodriguez <jrh@it.uc3m.es >.

Tabla de contenidos

1. Introducción 2
2. Proceso de ingeniería de releases 3
3. Construcción de la Release 8
4. Distribución 12
5. Extensibilidad 13

6. Lecciones aprendidas a partir de FreeBSD 4.4 14
7. Directrices para el futuro 14
8. Agradecimientos 15
9. Lecturas recomendadas 15

1. Introducción

El desarrollo de FreeBSD es un proceso realmente abierto al público. FreeBSD se alimenta de contribuciones de miles de personas del mundo entero. El Proyecto FreeBSD proporciona acceso público a través de CVS[1] de tal forma que cualquiera puede acceder a los mensajes de log y a los archivos de diferencias (también conocidos como "diffs" o parches) aplicados a distintas ramas de desarrollo, junto con el resto de funcionalidad que el gestor de código fuente pone a nuestra disposición. Este hecho, aunque muchas veces pasa inadvertido, ha constituido uno de los más importantes recursos de la comunidad y ha servido para captar y motivar a muchos desarrolladores con talento. No obstante, y creo que todo el mundo está de acuerdo con lo que voy a decir, sería un completo caos proporcionar acceso de escritura a todo el que pueda conectarse a Internet. Es por esto que existe sólo un "selecto" grupo de en torno a 300 personas que poseen dicho acceso de escritura en el repositorio de CVS. Estos *committers[6]* se responsabilizan del desarrollo del corazón de FreeBSD. Un *core-team[7]* compuesto por desarrolladores muy experimentados proporciona ciertas directrices a la dirección que va a tomar el proyecto.

El rápido ritmo de desarrollo de `FreeBSD` deja poco tiempo para pulir el sistema y proporcionar una release de calidad equivalente a las releases de sistemas comerciales. Para resolver este problema, se continúa el desarrollo en dos caminos paralelos. La rama de desarrollo principal se denomina *HEAD* o *trunk* (tronco) y constituye el punto de desarrollo más avanzado del árbol CVS. Esta rama consituye lo que llamamos "FreeBSD-CURRENT" o simplemente "-CURRENT" para abreviar.

También se mantiene una rama más estable, conocida como "FreeBSD-STABLE" o "-STABLE". Ambas ramas conviven en el repositorio maestro de CVS localizado en California y dicho repositorio se replica vía CVSup[2] creandose una serie de réplicas (también llamadas espejos o mirrors) por todo el mundo. FreeBSD-CURRENT[8] consituye el límite tecnológico (o "bleeding-edge" en inglés) del desarrollo del sistema FreeBSD y es donde se aplican en primer lugar cualquier cambio realizado al sistema. FreeBSD-STABLE constituye la rama de desarrollo de la cual se generan las releases principales. Los cambios en el sistema se producen a un ritmo variable asumiendose que dichos cambios generalmente se aplican primero a la rama -CURRENT, quedando a disposición de la comunidad de usuarios para que comprueben el correcto funcionamiento global del sistema de una forma exhaustiva antes de aplicarlos a -STABLE, en caso de que fuera necesaria su aplicación.

En el periodo entre releases, se construyen copias del sistema tomadas a determinadas horas de la noche y se ponen a disposición del público en `ftp://stable.FreeBSD.org/` . La amplia disponibilidad de releases de copias binarias actualizadas del sistema ("snaps-

hosts") y la tendencia de nuestra comunidad de usuarios a mantenerse a la última del desarrollo en la rama -STABLE mediante la utilización de CVSup y "`make world`"[8] ayuda a mantener la rama FreeBSD-STABLE en unas condiciones de fiabilidad excelentes que incluso llegan a ralentizar las peticiones de nuevas releases basadas en actividades de depuración de la calidad del software.

Los informes de problemas y las solicitudes de nuevas características no paran de producirse durante el ciclo de vida de una release. Los informes de problemas se almacenan en la base de datos GNATS[9] utilizando el correo eletrónico, la aplicación send-pr(1) o vía la interfaz web proporcionada en `http://www.FreeBSD.org/send-pr.html` . Además de la multitud de listas de correo de carácter técnico que FreeBSD pone a nuestra disposición, el lista de correo sobre 'Quality Assurance' en FreeBSD proporciona un foro de discusión sobre aspectos "a pulir" del sistema antes de su salida.

Para dar servicio a nuestro usuarios más conservadores, con la aparición de FreeBSDD 4.3 se introdujeron ramas individuales dentro del árbol CVS. Estas ramas de releases se crean poco tiempo después de la generación de una release final. Una vez generada la última release (la más actual o más reciente), sólo se aplican a esta release las modificaciones más críticas o necesarias, normalmente aquellas que provienen de fallos de seguridad. Además de las actualizaciones del código fuente a través de CVS, existen paquetes de parches binarios para mantener las releases *RELENG_X_Y* actualizadas.

La Sección 2, "Proceso de ingeniería de releases" describe las distintas fases del proceso de ingeniería de releases que se utiliza para construir el sistema real mientras que Sección 3, "Construcción de la Release" describe el proceso de contrucción en sí mismo. Sección 5, "Extensibilidad" describe cómo la release base puede ser ampliada por terceras partes y Sección 6, "Lecciones aprendidas a partir de FreeBSD 4.4" detalla algunas de las lecciones aprendidas durante la generación de la release FreeBSD 4.4. Por último, Sección 7, "Directrices para el futuro" presenta caminos futuros de desarrollo.

2. Proceso de ingeniería de releases

Las nuevas release de FreeBSD se generan a partir de la rama -STABLE en intervalos de aproximadamente cuatro meses. El proceso comienza a ejecutarse 45 días antes de la fecha de salida, cuando el ingeniero de releases envía un correo eletrónico a las listas de desarrollo de FreeBSD para recordar a los desarrolladores que disponen de tan solo 15 días para integrar nuevos cambios antes de la fase de congelación de código fuente. Durante este periodo de tiempo, muchos desarrolladores realizan lo que se ha dado en llamar "barrido MFC". MFC significa en inglés "Merge From CURRENT" (Integración desde CURRENT) y describe el proceso de unificación de los cambios aplicados en la rama de desarrollo -CURRENT a nuestra rama -STABLE.

2.1. Revisión de Código

Treinta días antes del lanzamiento de una release dada el repositorio de código fuente entra en una fase de "code slush" ("código aguanieve", en el sentido de no estar aún congelado y ser por tanto ligeramente moldeable). Durante este período todos los commits de la rama -STABLE deben ser aprobados por el Grupo de ingeniería de releases <re@FreeBSD.org >. Los cambios permitidos en esta fase de 15 días de duración son los siguientes:

- Arreglo de bugs o errores.
- Actualizaciones de la documentación.
- Parches relacionados con cualquier tipo de fallo en la seguridad.
- Cambios pequeños en controladores de dispositivos, tales como la adición de identificadores de dispositivo.
- Cualquier cambio adicional que el equipo de ingeniería de releases considere justificado, teniendo siempre en cuenta el riesgo potencial que puede conllevar.

Después de los primeros 15 días de código "slush", se genera una *release candidate* (candidata a release) o "RC" para su testeo exhaustivo por parte de la comunidad de usuarios y el código fuente entra en la fase de "code freeze" o congelamiento. En este punto resulta mucho más difícil aceptar cambios a menos que se descubran serios fallos de seguridad o bugs importantes. Durante esta fase, se genera al menos una RC cada semana, hasta que la release final ve la luz. Durante el periodo de tiempo comprendido desde el congelamiento del código hasta la generación de la release final, el equipo de ingeniería de releases se comunica constantemente con el equipo del "security officer", los equipos encargados de mantener la documentación y los mantenedores de ports, para asegurarse de que los distintos componentes necesarios para obtener una release existosa se encuentran disponibles y listos para ser construidos.

2.2. Lista de tareas para la release final.

Cuando todos los problemas encontrados en las releases candidatas se han corregido, se puede comenzar con el procedimiento de "pulimiento o enbellecimiento" de la release final.

2.2.1. Creación de una Rama Release

Como se describe en la introducción, la rama RELENG_X_Y es una característica relativamente nueva de nuestra metodología de generación de releases. El primer paso para crear esta rama consiste en asegurar que el código fuente utilizado "proviene" de la versión más reciente de RELENG_X .

```
/usr/src# cvs update --rRELENG_4 --P --d
```

El siguiente paso consiste en crear una etiqueta de rama, (*tag*), de esta forma se pueden generar diferencias entre el código actual y la rama de inicio fácilmente, utilizando CVS:

```
/usr/src# cvs rtag --rRELENG_4 RELENG_4_8_BP src
```

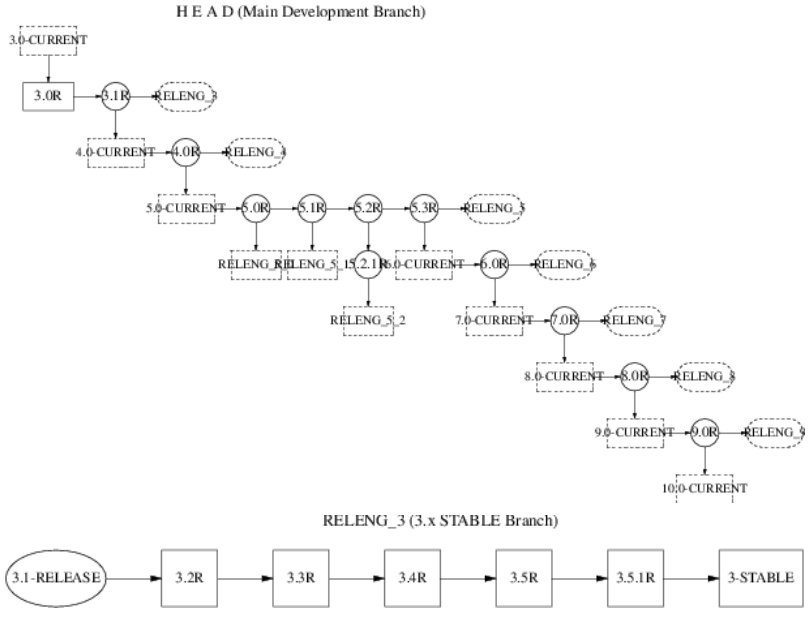
Y a continuación se crea la etiqueta de la rama:

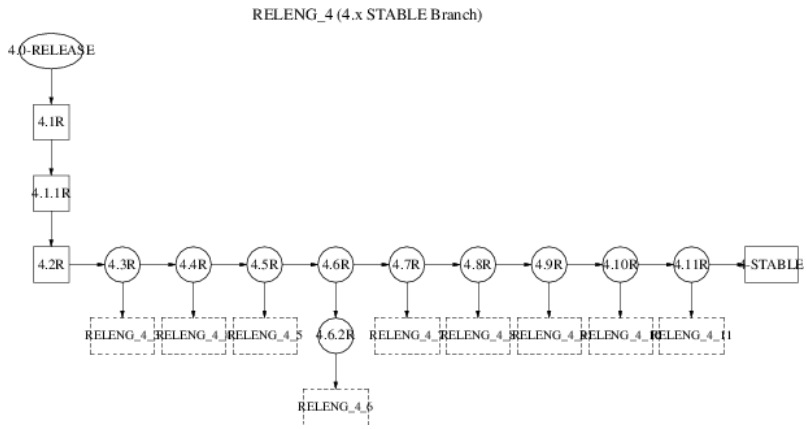
```
/usr/src# cvs rtag --b --rRELENG_4_8_BP RELENG_4_8 src
```

Nota

*Las etiquetas **RELENG_*** sólo pueden ser utilizadas por los CVS-meisters y los ingenieros de releases.*

Una "*etiqueta o tag*" es una característica de CVS que sirve para identificar el código fuente en un determinado instante del tiempo. Mediante el etiquetado del árbol, nos aseguramos de que las futuras releases puedan generar diferencias con respecto al mismo código fuente que se utilizó para generar las releases oficiales anteriores.





2.2.2. Elevación del número de versión

Antes de que la release final se puede etiquetar, construir y antes de que vea la luz, se deben modificar los siguientes ficheros de tal forma que reflejen el número de versión correcto:

- `doc/en_US.ISO8859-1/books/handbook/mirrors/chapter.xml`
- `doc/en_US.ISO8859-1/books/porters-handbook/book.xml`
- `doc/share/xml/freebsd.ent`
- `src/Makefile.inc1`
- `src/UPDATING`
- `src/gnu/usr.bin/groff/tmac/mdoc.local`
- `src/release/Makefile`
- `src/release/doc/en_US.ISO8859-1/share/xml/release.dsl`
- `src/release/doc/share/examples/Makefile.relnotesng`
- `src/release/doc/share/xml/release.ent`
- `src/share/examples/cvsup/standard-supfile`
- `src/sys/conf/newvers.sh`
- `src/sys/sys/param.h`
- `src/usr.sbin/pkg_install/add/main.c`

- `www/en/docs.xml`
- `www/en/cgi/ports.cgi`
- `ports/Tools/scripts/release/config`

El fichero “release notes” y el fichero “errata” también deben ajustarse de acuerdo con la nueva release (en la rama de la release) y deben cortarse adecuadamente en las ramas stable/currrent):

- `src/release/doc/en_US.ISO8859-1/relnotes/common/new.xml`
- `src/release/doc/en_US.ISO8859-1/errata/article.xml`

Sysinstall debe actualizarse para que proporcione el número actual de ports disponibles y la cantidad de espacio de disco requerida para instalar dicha colección de ports. Esta información se encuentra almacenada actualmente en el fichero `src/release/sysinstall/dist.c`.

Después de construir la release se debe actualizar el número almacenado en los siguientes ficheros para anunciar la release al resto del mundo:

- `www/share/xml/includes.release.xml`
- `www/share/xml/includes.release.xsl`
- `www/en/releases/*`
- `www/en/releng/index.xml`
- `www/en/news/news.xml`
- `www/en/search/web.atoz`
- `src/share/misc/bsd-family-tree`

2.2.3. Creación de las etiquetas de release

Cuando la release final se encuentra preparada se utiliza el siguiente comando para crear la etiqueta (a modo de ejemplo) `RELENG_4_8_0_RELEASE` .

```
/usr/src# cvs rtag --rRELENG_4_8 RELENG_4_8_0_RELEASE src
```

Los gestores de los proyectos de Documentación y de los Ports se responsabilizan del correcto etiquetado de sus respectivos árboles utilizando `RELEASE_4_8_0`.

Ocasionalmente se puede presentar un apaño o arreglo de última hora justo *después* de la creación de las etiquetas finales. En la práctica esto no constituye un problema ya que

CVS permite cierta manipulación de etiquetados mediante `cvs tag -d nombredeetiqueta nombredefichero .` Es muy importante que dichos cambios de última hora se etiqueten adecuadamente para que pasen a formar parte de la nueva release. Las releases de FreeBSD deben ser siempre "reproducibles". Los "hacks" locales dentro del entorno de ingeniería de releases no están permitidos salvo que se efectúen mediante una correcta manipulación y notificación.

3. Construcción de la Release

Cualquier persona dueña de una potente máquina y con acceso de lectura al repositorio de código fuente puede "construir" las "releases" de FreeBSD. En la práctica esto significa que cualquiera puede generar el proceso de construcción de releases, ya que, como se comentó con anterioridad, FreeBSD ofrece acceso CVS anónimo a todo el mundo (consulte el Handbook para más detalles). El único requisito imprescindible para realizar este proceso es la existencia del dispositivo vn(4). (En -CURRENT, este dispositivo ha sido reemplazado por el nuevo driver de discos en memoria denominado md(4).) Si el dispositivo no se encuentra cargado en el kernel, debería cargarse automáticamente al ejecutar el comando vnconfig(8) como parte de la fase de creación del medio de arranque. Todas las herramientas necesarias para construir la release se encuentran disponibles en el repositorio de CVS dentro del directorio `src/release`. Estas herramientas proporcionan una forma consistente y robusta de construir releases de FreeBSD. Una release completa se puede construir utilizando un único comando, incluyendo la creación de las imágenes ISO necesarias para realizar copias en CDROM, junto con disquetes de instalación y un directorio para la instalación por FTP. Este comando fue adecuadamente bautizado como `make release`.

3.1. `make release`

Para poder construir la releases de una forma exitosa se debe rellenar primero el directorio `/usr/obj` ejecutando el comando `make world` o simplemente `make buildworld`. El target release que utiliza el comando make necesita varias variables, tal como se muestra a continuación:

- `CHROOTDIR` - El directorio que se utiliza para el entorno de chroot durante la construcción de la release entera.
- `BUILDNAME` - El nombre de la release que se va a construir.
- `CVSROOT` - La ubicación del repositorio de CVS.
- `RELEASETAG` - La etiqueta CVS correspondiente con la release que se quiere construir.

Si no se dispone de acceso a un repositorio de CVS local, se puede realizar una copia espejo (un mirror) con CVSup. El fichero `/usr/share/examples/cvsup/cvs-supfile` , sirve como buen punto de partida para realizar un mirror del repositorio de CVS.

Si se omite `RELEASETAG`, la release se construirá a partir de la rama `HEAD` (también conocida como -CURRENT). Las releases que se construyen desde el principio se conocen normalmente con el nombre de "-CURRENT snapshots".

Existen otras variables que se pueden editar para adaptar el proceso de construcción de la release. La mayoría de estas variables se encuentran documentadas al comienzo de `src/release/Makefile`. El comando exacto para contruir la release oficial de FreeBSD 4.7 (x86) fue:

```
make release CHROOTDIR=/local3/release \
       BUILDNAME=4.7-RELEASE \
       CVSROOT=/host/cvs/usr/home/ncvs \
       RELEASETAG=RELENG_4_7_0_RELEASE
```

El `Makefile` de la release se puede dividir en varios pasos distintos.

- Creación de un entorno de sistema limpio en una jerarquía de directorios separada utilizando "`make installworld`".
- Comprobación de la correcta versión de los ficheros fuentes almacenados en la jerarquía de directorios recién creada, junto con el chequeo de la documentación y de los ports utilizando, todo ello a través de CVS.
- Relleno de los directorios `/etc` y `/dev` dentro del entorno chroot.
- Creación de un "chroot" dentro de la jerarquía de directorios creada, para que resulte más dificil que el entorno de la máquina se vea contaminado por la construcción de la release.
- `make world` dentro del entorno de chroot.
- Contrucción de los binarios relacionados con Kerberos.
- Construcción del kernel `GENERIC`.
- Creación de un esqueleto del árbol de directorios donde se construirán y empaquetarán las distribuciones binarias.
- Construcción e instalación del conjunto de herramientas de documentación necesarias para convertir los fuentes de la documentación (SGML) en los documentos HTML y de texto que pasarán a formar parte de la release.
- Construcción e instalación de la documentación real (manuales de usuario, tutoriales, release notes, listas de compatibilidad de hardware, etc.)
- Construcción de los "decisivos" binarios utilizados en los disquetes de instalación.

- Colocación adecuada de los de los paquetes de distribución de binarios y de fuentes.
- Creación del medio de arranque y del disquete “fixit” o salvamento.
- Creación de la jerarquía de directorios de instalación por FTP.
- Creación de imágenes ISO para CDROM/DVD(opcional).

Para más información sobre la infraestructura involucrada en el proceso de construcción de la release, el lector puede consultar release(7).

3.2. Construcción deXFree86™

XFree86™ es un componente importante para muchos usuarios de entornos gráficos. Antes de la release FreeBSD 4.6 las se usaba XFree86™3.*X* por defecto. La forma más sencilla de construir estas versiones consiste en utilizar el script `src/release/scripts/X11/build_x.sh` . Este script requiere que XFree86™ y Tcl/Tk se encuentren instalados previamente en la máquina donde se realiza la construcción. Después de compilar los servidores X necesarios, el script empaqueta todos los ficheros en “tarballs” que sysinstall(8) sabe cómo localizar utilizando el directorio `XF86336` del medio de instalación.

A partir de FreeBSD 4.6, sysinstall(8) instala XFree86™ 4.*X* por defecto, como cualquier otro conjunto de paquetes. Estos paquetes se pueden construir a partir del “package-building cluster” o a partir de las etiquetas del árbol de ports adecuadas.

Nota

Es importante borrar cualquier configuración particular almacenada en `/etc/make.conf` . Por ejemplo, no sería una idea muy inteligente distribuir binarios que se construyeron en un sistema con la variable `CPUTYPE` asignada a un determinado procesador.

3.3. Software Contribuido (“ports”)

La colección de FreeBSD Ports está compuesta por más de 24,000 paquetes de software de terceras partes que se encuentran disponibles para FreeBSD. El Grupo de administración de ports <portmgr@FreeBSD.org > se responsabiliza de mantener un árbol de ports consistente que se pueda utilizar para crear paquetes binarios, los cuales se añaden a las releases oficiales de FreeBSD.

Las actividades de ingeniería de releases para nuestra colección de paquetes software de terceras partes se encuentra más allá del objetivo de este documento. Otro artículo, `http://www.freebsd.org/doc/en_US.ISO8859-1/articles/releng-packages/`, cubre este tema en profundidad.

3.4. ISOs de la release

A partir de FreeBSD 4.4, el Proyecto FreeBSD decidió lanzar gratuitamente al público las cuatro imágenes ISO que anteriormente se vendían en *BSDi/Wind River Systems/FreeBSD Mall* como distribuciones en CDROM "oficiales". Cada uno de los cuatro discos debe contener un `README.TXT` que explica el contenido de cada disco, un `CDROM.INF` que proporciona metadatos para que sysinstall(8) pueda validar la información en él contenida y un `filename.txt` que proporciona un "manifiesto". Este *manifiesto* se puede crear utilizando un simple comando:

```
/stage/cdrom# find -. --type f -| sed --e -'s/^\.\///' -| sort > ↺
filename.txt
```

Los requisitos concretos de cada CD se resumen a continuación.

3.4.1. Disco 1

El primer disco se crea casi en su totalidad a partir del comando `make release`. Los únicos cambios que se deben realizar dentro del directorio `disc1` son la adición de un directorio `tools`, de XFree86™ y de los paquetes de terceras partes más populares que quepan dentro del espacio remanente de dicho primer disco. El directorio `tools` contiene el software que permite a los usuarios crear disquetes de instalación desde otros sistemas operativos. Este disco debe crearse como autoarrancable para que los usuarios de PCs modernos no necesiten crear disquetes de arranque y puedan utilizar la característica de autoarranque desde CD.

Si se proporciona una versión alternativa de XFree86™, sysinstall(8) debe actualizarse para reflejar la nueva localización y las instrucciones de instalación. El código relevante se encuentra en `src/release/sysinstall` en -STABLE o en `src/usr.sbin/sysinstall` en -CURRENT. Específicamente, se deben actualizar `dist.c`, `menus.c` y `config.c`.

3.4.2. Disco 2

El segundo disco se crea en su mayor parte a partir del comando `make release`. Este disco contiene un "sistema de ficheros vivo", que se puede utilizar a partir de sysinstall(8) para resolver problemas durante el proceso de instalación de FreeBSD. Este disco se debe construir como autoarrancable y debe contener una copia comprimida del repositorio de CVS dentro del directorio `CVSROOT`, junto con demostraciones de software comercial localizadas dentro del directorio `commerce`.

3.4.3. Discos 3 and 4

Los dos discos que quedan contienen paquetes de software para FreeBSD. Estos paquetes deben agruparse de tal forma que un paquete y todas sus *dependencias* quepan en el mismo disco. Se puede obtener más información sobre la creación de estos discos en el artículo `http://www.freebsd.org/doc/en_US.ISO8859-1/articles/releng-packages/` .

4. Distribución

4.1. Servidores de FTP

Cuando se ha probado exhaustivamente la release y se ha empaquetado debidamente para proceder a su distribución, se debe actualizar el sitio maestro de FTP. Los sitios FTP oficiales de FreeBSD son mirrors del sitio FTP maestro, también llamado `ftp-master`. Cuando la release está lista, se deben modificar los siguientes ficheros en el servidor `ftp-master`:

`/pub/FreeBSD/releases/arch/X.Y-RELEASE/`
: El directorio de instalación dde FTP que se crea con la salida del comando `make release`.

`/pub/FreeBSD/ports/arch/packages-X.Y-release/`
: La construcción del paquete completo de la release.

`/pub/FreeBSD/releases/arch/X.Y-RELEASE/tools`
: Un enlace simbólico a `../../../tools` .

`/pub/FreeBSD/releases/arch/X.Y-RELEASE/packages`
: Un enlace simbólico a `../../../ports/arch/packages-X.Y-release` .

`/pub/FreeBSD/releases/arch/ISO-IMAGES/X.Y/X.Y-RELEASE-arch-*.iso`
: Las imágenes ISO. El "*" se sustituye por `disc1`, `disc2`, etc. Solo si existe `disc1` junto con un CD de primera instalación alternativo (por ejemplo una instalación recortada o reducida sin sistema de ventanas) puede existir también un `mini`.

Para obtener más información sobre la arquitectura de mirrors para la distribución del sistema FreeBSD, se ruega al lector que consulte el artículo Mirroring FreeBSD.

Puede que transcurran desde varias horas hasta varios días hasta que la mayoría de los sitios FTP Tier-1 se actualicen con respecto al `ftp-master`, esto depende de si un determinado paquete se cargó o no se cargó en determinado instante. Es imperativo que los ingenieros de releases se coordinen con lista de correo de avisos para administradores de réplicas de FreeBSD antes de anunciar la disponibilidad general del nuevo software en los sitios FTP. Para que todo fuera bien el paquete de la release se debería cargar al menos cuatro días antes del día oficial de lanzamiento de la release. Los permisos para el grupo "other" deben desactivarse completamente para que los sitios espejos puedan descargar la release pero no así los usuarios finales, hasta que llegue el día oficial del lanzamiento. Se debe enviar un correo a lista de correo de avisos para administradores de réplicas de FreeBSD cuando se publican la release con los permisos modificados, diciendo que la release ha sido puesta en escena y proporcionando la fecha a partir de la cual los mirrors deben comenzar a dar permisos de acceso para el público en general. Se debe comprobar que se incluye información relativa a zonas horarias, por ejemplo información relativa a GMT.

4.2. Replicación de CD-ROMs

Dentro de poco tiempo: Consejos para enviar ISOs de FreeBSD a un replicador e información sobre las medidas de aseguramiento de la calidad que se deben tomar.

5. Extensibilidad

Aunque FreeBSD consitituye un sistema operativo "completo", no existe nada que nos obligue a utilizarlo exactamente igual que como se ha empaquetado para crear la distribución. Es decir, el sistema FreeBSD se ha diseñado para ser tan extensible como sea posible de tal forma que se puede utilizar como la base sobre la que se pueden construir productos comerciales. La única "regla" sobre este tema es que si se piensa distribuir FreeBSD con una serie de cambios profundos en él , se anima a que se *documenten adecuadamente dichos mejoras*. La comunidad FreeBSD sólo puede ayudar a los usuarios que utilizan el software que dicha comunidad distribuye. Se anima encarecidamente hacia la innovación tanto en el proceso de instalación como en las herramientas de administración, pero no se puede esperar un respuesta a todas las preguntas que surgan sobre dichos temas.

5.1. Creación de disquetes de arranque a medida

Muchas organizaciones poseen complejos requisitos que pueden consistir en módulos del kernel adicionales o herramientas de entorno de usuario que deben añadirse en los discos de instalación. La forma "rápida y sucia" de añadir estas cosas consiste en modificar el directorio temporal que contiene la estructura de un `make release`:

- Aplicar parches o añadir archivos adicionales dentro del directorio chroot de construcción de la release.
- `rm ${CHROOTDIR}/usr/obj/usr/src/release/release.[59]`
- Reconstruir sysinstall(8), el kernel o cualquier otra parte del sistema que se vea afectada por los cambios.
- `chroot ${CHROOTDIR} ./mk floppies`

Los nuevos disquetes de instalación estarán en `${CHROOTDIR}/R/stage/floppies`.

También se puede llamar el objetivo de make `boot.flp` o directamente al "script" de creación del sistema de ficheros `src/release/scripts/doFS.sh .`

Los parches locales también se pueden proporcionar al proceso de construcción de la release mediante la definición de la variable `LOCAL_PATCH` dentro de `make release`.

5.2. "Scripts" para `sysinstall`

La instalación y configuración del sistema FreeBSD a través de sysinstall(8) se puede modificar mediante "scripts" para que proporcione instalaciones automáticas a grandes or-

ganizaciones. Esta funcionalidad se puede utilizar conjuntamente con Intel® PXE[13] para arrancar sistemas a través de la red, o a través de disquetes de arranque a medida utilizando un "script" de sysinstall. Un ejemplo de guión sysinstall se encuentra disponible en `src/release/sysinstall/install.cfg` .

6. Lecciones aprendidas a partir de FreeBSD 4.4

El proceso de ingeniería de releases de FreeBSD 4.4 comenzó formalmente el 1 de Agosto de 2001. Después de esa fecha todos los "commits" o modificaciones sobre la rama `RELENG_4` de FreeBSD tuvieron que ser aprobados explícitamente por el Grupo de ingeniería de releases <re@FreeBSD.org >. La primera "release candidate" para la arquitectura x86 apareció el 16 de Agosto, seguida por otras cuatro releases candidatas hasta que vió la luz la release final el 18 de Septiembre. El "security officer" estuvo muy involucrado en la última semana del proceso ya que se descubrieron varios problemas de seguridad en las "release candidates" iniciales. Más de *500* correos electrónicos fueron enviados al Grupo de ingeniería de releases <re@FreeBSD.org > en poco más de un mes.

Nuestra comunidad de usuarios ha dejado muy claro que la seguridad y estabilidad de las releases de FreeBSD no pueden sacrificarse por culpa de plazos autoimpuestos o fechas de lanzamiento. El Proyecto FreeBSD ha crecido tremendamente durante su tiempo de vida y se ha visto claramente la necesidad de estandarizar los procedimientos de ingeniería de releases. Este hecho será incluso más importante a medida que FreeBSD vaya estando disponible para más plataformas.

7. Directrices para el futuro

Es de vital importancia para nuestras actividades de ingeniería de releases el ser capaces de crecer al mismo ritmo que nuestra base de usuarios. Junto con estas líneas estamos trabajando duramente en los procedimientos involucrados en la producción de releases de FreeBSD.

- *Paralelismo* - Algunas partes de la construcción de la release son "vergonzosamente paralelas". La mayoría de las tareas que se realizan son intensivas en entrada-salida, de tal forma que resulta más importante poseer varios discos duros de alta velocidad que utilizar varios procesadores a la hora de acelerar el proceso del comando `make release`. Si se utilizan varios discos para las distintas jerarquías de directorios dentro del entorno chroot(2), entonces el "checkout" de los árboles de `ports` y de los `doc` se puede producir al mismo tiempo que la ejecución en otro disco del comando `make world`. Mediante la utilización de un sistema RAID (hardware o software) se puede reducir significativamente el tiempo total de construcción de la release.

- *Releases construidas para otros sistemas finales ("cross building")* : ?Se puede construir una release para IA-64 o Alpha en un hardware x86? `make TARGET=ia64 release` .

- *Tests de Regresión* - Se necesitan mejores herramientas automatizadas para comprobar la corrección del sistema FreeBSD.

- *Herramientas de Instalación* - Nuestro programa de instalación ha sobrepasado su tiempo de vida previsto. Se encuentran en desarrollo varios proyectos para proporcionar un mecanismo de instalación más avanzado. Uno de los más prometedores es el proyecto libh[5] cuyo objetivo consiste en proporcionar un entorno de paquetes nuevo e inteligente junto con un programa de instalación gráfico.

8. Agradecimientos

Me gustaría agradecer a Jordan Hubbard por darme la oportunidad de colaborar en algunas de las responsabilidades del equipo de ingeniería de releases en FreeBSD 4.4 y también me gustaría agradecer públicamente su trabajo y dedicación durante todos estos años para poder situar a FreeBSD en el sitio de honor que le corresponde hoy día. Por supuesto que la release 4.4 no hubiera visto la luz sin el trabajo de Satoshi Asami, Steve Price, Bruce A. Mah, Nik Clayton, David O'Brien, Kris Kennaway, John Baldwin y del resto de la comunidad FreeBSD. También me gustaría agradecer especialmente a Rodney Grimes, Poul-Henning Kamp y muchos otros que trabajaron en las herramientas de ingeniería de releases en los comienzos del sistema FreeBSD. Este artículo está basado en documentos de ingeniería de releases del CSRG[14], el NetBSD Project[11] y en las notas del proceso de ingeniería de releases propuestas por John Baldwin[12].

9. Lecturas recomendadas

[1] CVS - Concurrent Versions System `http://www.cvshome.org`

[2] CVSup - The CVS-Optimized General Purpose Network File Distribution System `http://www.polstra.com/projects/freeware/CVSup`

[3] `http://bento.FreeBSD.org`

[4] FreeBSD Ports Collection `http://www.FreeBSD.org/ports`

[5] The libh Project `http://www.FreeBSD.org/projects/libh.html`

[6] FreeBSD Committers `http://www.FreeBSD.org/doc/en_US.ISO8859-1/articles/contributors/staff-committers.html`

[7] FreeBSD Core-Team `http://www.FreeBSD.org/doc/en_US.ISO8859-1/articles/contributors/staff-core.html`

[8] FreeBSD Handbook `http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/handbook`

[9] GNATS: The GNU Bug Tracking System `http://www.gnu.org/software/gnats`

[10] FreeBSD PR Statistics `http://www.FreeBSD.org/prstats/index.html`

[11] NetBSD Developer Documentation: Release Engineering `http://www.NetBSD.org/developers/releng/index.html`

[12] John Baldwin's FreeBSD Release Engineering Proposal `http://people.FreeBSD.org/~jhb/docs/releng.txt`

[13] PXE Jumpstart Guide `http://www.FreeBSD.org/doc/en_US.ISO8859-1/articles/pxe/index.html`

[14] Marshall Kirk McKusick, Michael J. Karels, and Keith Bostic: *The Release Engineering of 4.3BSD*