

The NetBSD Project

Introducción a NetBSD y pkgsrc

Julio M. Merino Vidal <jmmv@NetBSD.org>

Partyzip@ 2005



Contenido

- ♦ NetBSD vs. Linux.
- ♦ Un poco de historia.
- ♦ Objetivos.
- ♦ Política de versiones.
- ♦ Hilos POSIX.
- ♦ Multiprocesador.
- ♦ Sistemas de archivos.
- ♦ Summer of Code.
- ♦ Cómo obtener NetBSD.
- ♦ Compilación del sistema.
- ♦ pkgsrc y syspkg.
- ♦ Dónde obtener ayuda.
- ♦ Cómo reportar fallos.
- ♦ Preguntas.

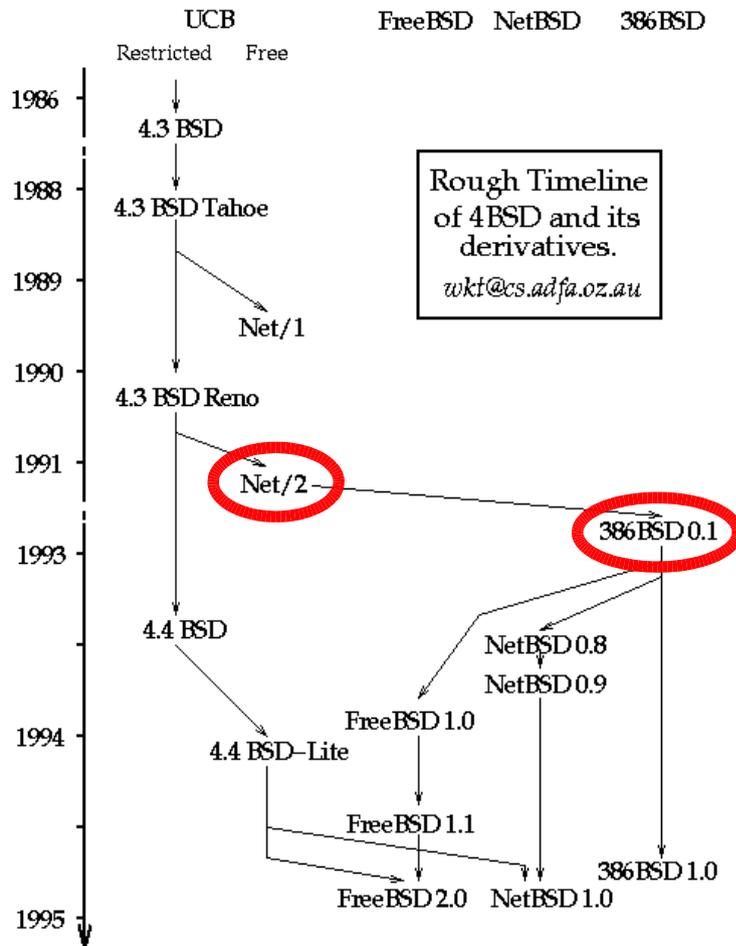


NetBSD vs. Linux

- ♦ Sistema completo.
- ♦ Basado en 4.4BSD.
- ♦ Licencia BSD.
- ♦ Núcleo más base GNU.
- ♦ Escrito desde cero.
- ♦ Licencia GPL.



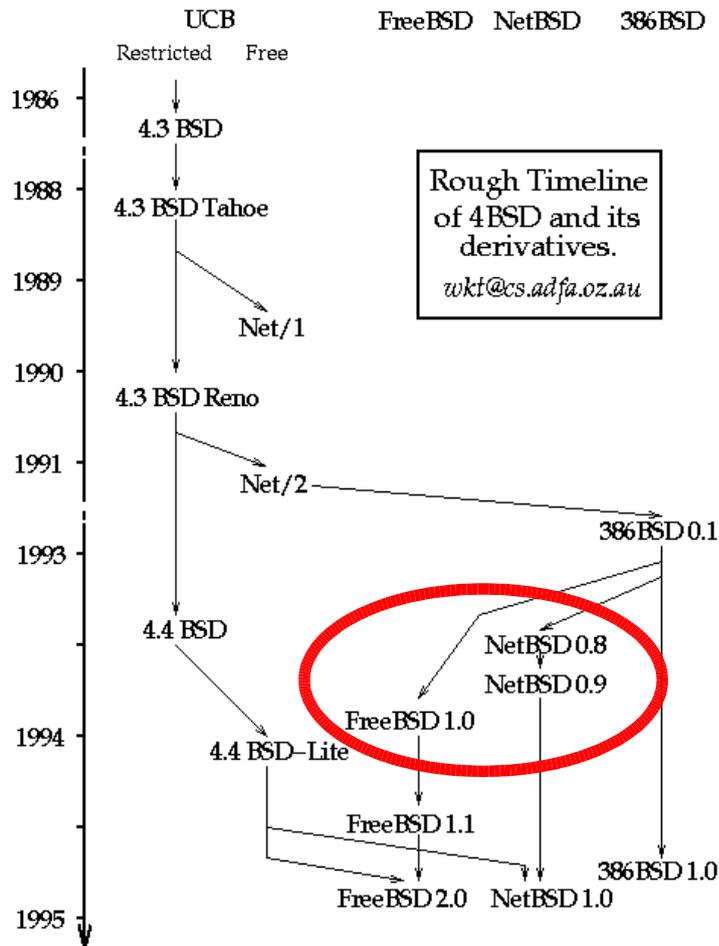
Un poco de historia (1/4)



- ◆ *Fork* de 4.3BSD Networking/2: 386BSD.
- ◆ Motivo:
 - Frustración en la integración de parches.



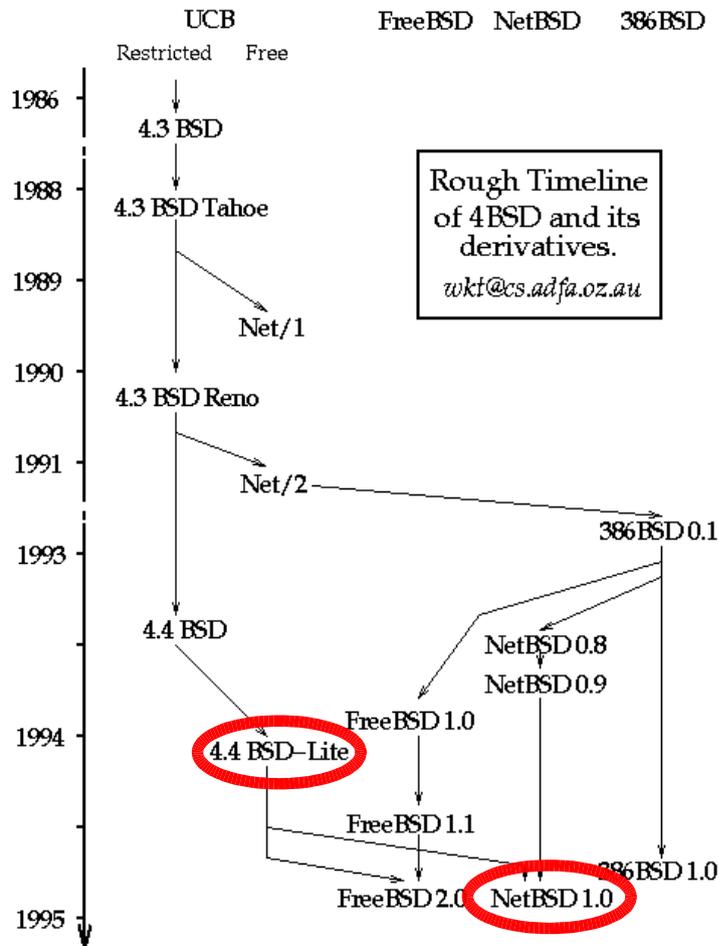
Un poco de historia (2/4)



- ◆ 386BSD deriva en:
 - NetBSD (portabilidad)
 - FreeBSD (i386)
- ◆ Primera versión:
 - NetBSD 0.8
 - 20 de abril de 1993



Un poco de historia (3/4)



- ♦ Integración de las mejoras en 4.4BSD (Lite).
- ♦ NetBSD 1.0 ve la luz 26 de octubre de 1994.



Un poco de historia (4/4)

- ♦ Hasta la fecha:
 - 23 versiones publicadas en total.
 - 10 de ellas versiones mayores.
- ♦ Última versión:
 - NetBSD 2.0.2, 14 de abril de 2005.



Objetivos (1/5)

- ♦ Diseño correcto:
 - Posiblemente el objetivo más importante.
 - Ejemplo: abstracción del acceso al bus del sistema.
 - *“It doesn't work **unless** it's right”*.
- ♦ Completitud del sistema:
 - Protocolos de red.
 - Utilidades de desarrollo.
 - Sistema de paquetes.



Objetivos (2/5)

- ♦ Estabilidad:
 - Sistema usado en producción.
- ♦ Rapidez:
 - Plataformas antiguas vs. nuevas.
 - Micro vs. macro-optimizaciones.



Objetivos (3/5)

- ♦ Libre distribución:
 - Uso de la licencia BSD.
 - Algunas herramientas añadidas son GPL.
- ♦ Transportable:
 - División MI/MD.
 - Ejemplo: fxp(4) funciona sobre alpha, i386, macppc, etc.
 - 40 arquitecturas soportadas.



Objetivos (4/5)

- ♦ Interoperable:
 - Emulación binaria: Linux, FreeBSD, Solaris, etc.
 - Emuladores: wine, qemu, doscmd, etc.
 - Sistemas de archivos: FFS, Ext2, FAT, ISO 9660, etc.
 - Protocolos de red: TCP/IP, NFS, Appletalk, etc.



Objetivos (5/5)

- ♦ Seguimiento de estándares:
 - Extremadamente cercano a POSIX.1.
 - (Nunca lo será oficialmente: es muy caro).
 - Encaminado hacia POSIX.2.
 - Estándares de facto: BSD y Linux.
 - X Window System (X11R6).



Política de versiones (1/4)

- ♦ NetBSD-current:
 - Versión de desarrollo.
 - Posiblemente inestable.
 - HEAD del CVS.
 - Numeración: N.99.M:
 - N: Número de la última versión mayor.
 - M: Número de versión del la ABI del núcleo.
 - Ejemplos: 2.99.1, 2.99.2, 3.99.5, etc.



Política de versiones (2/4)

- ♦ Versiones oficiales:
 - Mayores: 2.0, 3.0, etc.
 - Nuevas infraestructuras, controladores, etc.
 - Menores: 2.1, 2.2, etc.
 - Arreglos de múltiples tipos.
 - Nuevos controladores ya probados y estabilizados.
 - Críticas: 2.0.1, 2.0.2, etc.
 - Arreglos de seguridad y/o estabilidad importantes.



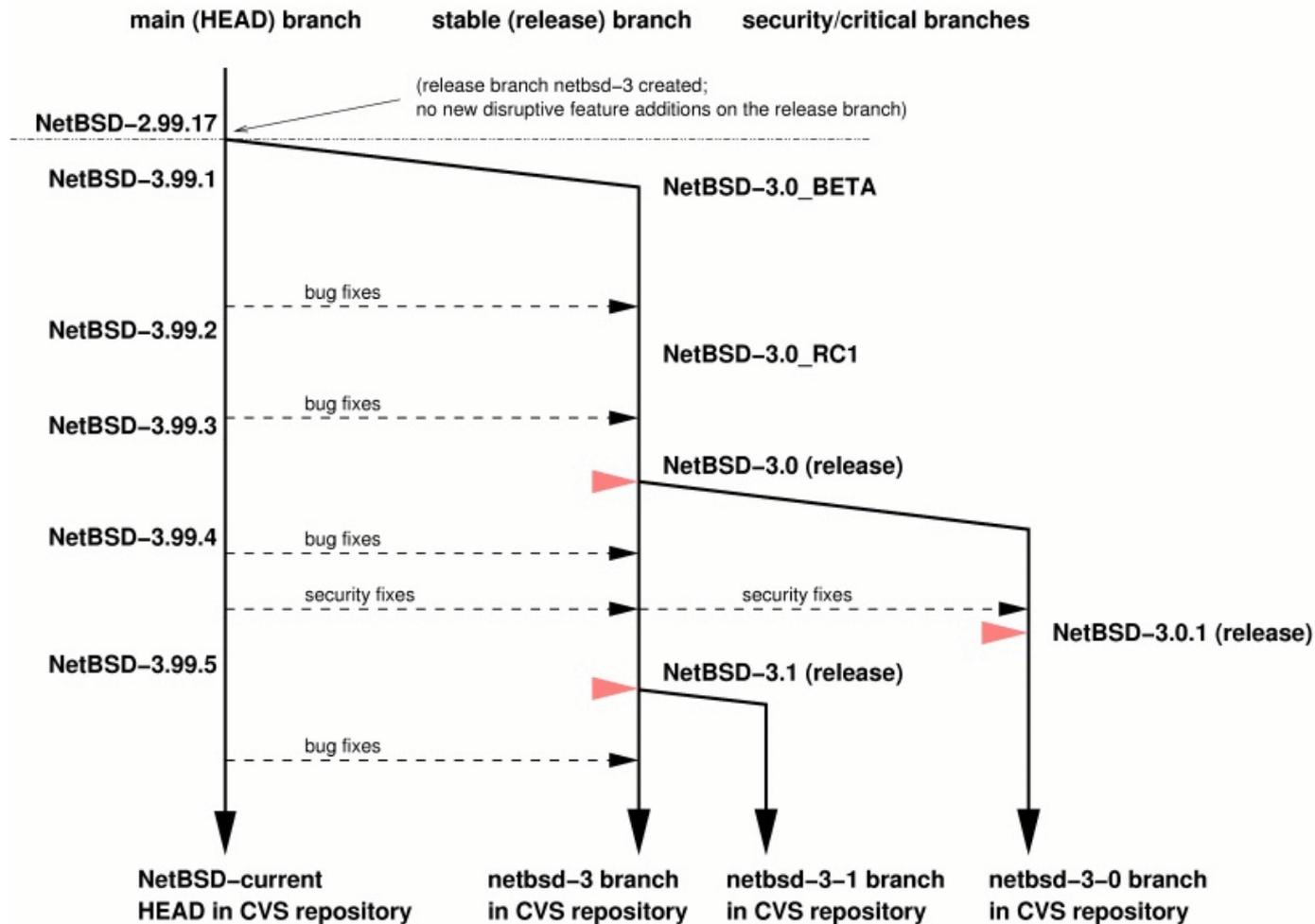
Política de versiones (3/4)

- ♦ Ramas de mantenimiento:
 - netbsd-M:
 - M: Número de versión mayor.
 - Contiene las versiones M.x (ej. M.0, M.1, etc.).
 - netbsd-M-N:
 - M: Número de versión mayor.
 - N: Número de versión menor.
 - Contiene las versiones M.N.y (ej. M.N.0, M.N.1, etc).



Política de versiones (4/4)

NetBSD release graph



Hilos POSIX

- ♦ Soporte en el núcleo (17 de enero de 2003).
- ♦ Basado en *Scheduler Activations*:
 - Modelo N:M.
- ♦ Librería para desarrollo de aplicaciones:
 - Estándar POSIX.
 - libpthread.



Multiprocesador

- ♦ Soporte en: alpha, amd64, i386, macppc, sparc64, sparc, vax.
- ♦ Limitaciones:
 - *Big lock*: sólo un proceso en el núcleo.
 - Hilos de un proceso en un sólo procesador.
- ♦ Opción MULTIPROCESSOR en el núcleo.



Sistemas de archivos (1/7)

- ♦ Fast File-System (FFS):
 - UFS con mejoras de velocidad.
 - *Soft updates*: Escritura asíncrona de meta-datos.
 - Intenta mantener la consistencia.
 - Opción BUFQ_READPRIO.
 - Opción UFS_DIRHASH.
 - Sin dietario.



Sistemas de archivos (2/7)

- ♦ Fast File-System v2 (FFSv2):
 - Basado en FFS.
 - Originario de FreeBSD.
 - Soporte de discos y archivos muy grandes (64 bits).
 - Soporte para ACLs (aún no disponible en NetBSD).



Sistemas de archivos (3/7)

- ♦ Memory File-System (MFS):
 - FFS sobre páginas de memoria virtual.
 - Idea original y demostración del sistema de capas.
 - Estructuras de datos y algoritmos poco eficientes.
 - Necesidad de un reemplazo.

Sistemas de archivos (4/7)

- ♦ Log-structured File-System (LFS):
 - Con dietario.
 - Agrupa las escrituras en disco.
 - Necesidad de “limpieza” periódica.
- ♦ De red:
 - NFSv3.
 - SMB.
 - Coda.



Sistemas de archivos (5/7)

- ♦ Compatibilidad con otros sistemas:
 - FAT12, FAT16 y FAT32.
 - NTFS (sólo lectura; escritura muy limitada).
 - Ext2.
 - ISO 9660.



Sistemas de archivos (6/7)

- ♦ Por capas:
 - Null: duplicación de un árbol.
 - Union: mezcla de dos árboles.
 - Portal: tcp, fs, rfilter, wfilter.
 - Overlay: útil como ejemplo.
 - Umap: Cambio de UIDs/GIDs dinámico.



Sistemas de archivos (7/7)

- ♦ En desarrollo:
 - UDF (lectura y escritura).
 - Dietario para FFS.
 - Adaptación de ReiserFS y XFS (sólo FreeBSD).
 - Otros (dentro del Summer of code).



Summer of code (1/9)

- ♦ Iniciativa de Google.
- ♦ 8 proyectos para NetBSD.
- ♦ Proyecto NetBSD-SoC:
 - Punto de encuentro para los 8 proyectos.
 - <http://NetBSD-SoC.sourceforge.net/>



Summer of code (2/9)

- ♦ Sistema de archivos eficiente en memoria (tmpfs):
 - FFS no está diseñado para trabajar en memoria.
 - Necesidad de un reemplazo para MFS.
 - Menor consumo de recursos posible.



Summer of code (3/9)

- ♦ Soporte de caracteres anchos en curses (wcurses):
 - Necesidad de internacionalización.
 - curses utiliza bytes.
 - Posibles problemas de velocidad.



Summer of code (4/9)

- ♦ OpenPGP con licencia BSD (bpg):
 - Implementación del estándar OpenPGP.
 - Necesario para proporcionar archivos firmados.
 - Licencia BSD.
 - Reemplazo de GnuPG:
 - Complicado de usar.
 - Licencia GPL.



Summer of code (5/9)

- ♦ Zeroconf (zeroconf):
 - Configurar una red IP:
 - Sin servidores.
 - De forma automática.
 - Implementación de Zeroconf para NetBSD.



Summer of code (6/9)

- ♦ Pruebas de regresión (regress):
 - Probar el correcto funcionamiento del sistema.
 - Evitar regresiones durante modificaciones.
 - Automatizar las pruebas existentes.



Summer of code (7/9)

- ♦ Sistemas de archivos en espacio de usuario (userfs):
 - Implementación de sistemas de archivos en espacio de usuario.
 - Similar a FUSE de Linux.



Summer of code (8/9)

- ♦ Controlador de red NDIS (ndis):
 - Necesidad de uso del hardware.
 - Especificaciones no disponibles.
 - Sólo controladores propietarios.



Summer of code (9/9)

- ♦ HFS+ (hfs):
 - Implementación de HFS+ en el núcleo.
 - Instalación de NetBSD de forma autónoma.
 - Evitar el uso de hfsutils, Linux o Mac OS (X).
 - Mejorar la interacción con Mac OS (X).



Cómo obtener NetBSD (1/3)

- ♦ Escoger una réplica cercana; consultar:
 - <http://www.NetBSD.org/mirrors/>
- ♦ FTP
 - <ftp://ftp.réplica.org/pub/NetBSD/NetBSD-2.0.2/>
 - <ftp://ftp.réplica.org/pub/NetBSD-daily/>
 - <ftp://ftp.réplica.org/pub/iso/>



Cómo obtener NetBSD (2/3)

- ♦ CVS:
 - Obtener los datos de la página de réplicas.
 - Módulos disponibles:
 - src: Código del sistema base.
 - xsrc: Código de XFree86.
 - htdocs: Código de la página web.
 - pkgsrc: Sistema de paquetes.



Cómo obtener NetBSD (3/3)

- ♦ Ejemplo de CVS:
 - CVS_RSH=/usr/bin/ssh
export CVS_RSH
CVSROOT=:ext:anoncvs@anoncvs.NetBSD.org:/cvsroot
export CVSROOT
cd /usr
cvs -q -P checkout -r netbsd-2 src
cvs -q -P checkout -r netbsd-2 xsrc



Compilación del sistema (1/4)

- ♦ Fuentes de todo el sistema en /usr/src.
- ♦ Compilación con make(1) complicada:
 - Desincronización de las utilidades.
 - Dificultad de compilar para otras plataformas.
 - Muchos comandos a recordar.



Compilación del sistema (2/4)

- ♦ `build.sh`:
 - Guión para automatizar la compilación del sistema.
 - Aparece en NetBSD 1.6.
 - Facilita la compilación cruzada:
 - Crea un *toolchain* completo para la plataforma deseada.
 - Lo utiliza para compilar todo el sistema.
 - Compilación de NetBSD desde otros sistemas.
 - Un único comando que lo engloba todo.



Compilación del sistema (3/4)

- ♦ Sintaxis:
 - `cd /usr/src && ./build.sh [opciones] <objetivos>`
- ♦ Opciones interesantes:
 - `-j <num>`: Compilación en paralelo.
 - `-M <dir>`: Usa 'dir' como directorio de trabajo.
 - `-m <nombre>`: Selecciona la plataforma de destino.
 - `-N <nivel>`: Selecciona el detalle de los mensajes.
 - `-x`: Compila X11R6 junto al sistema.



Compilación del sistema (4/4)

- ♦ Creación de un *toolchain*:
 - build.sh tools
 - build.sh -m mac68k tools
- ♦ Compilación e instalación del sistema base:
 - build.sh distribution install=/
 - build.sh release
 - build.sh -R /archive/NetBSD-current release



Compilación de un núcleo (1/4)

- ♦ Pasos a seguir con build.sh:
 - Crear un archivo de configuración.
 - Usar build.sh para compilar el kernel.
- ♦ Pasos a seguir sin build.sh:
 - Crear un archivo de configuración.
 - Usar config(8) para configurar la compilación.
 - Ejecutar make(1) en dicho directorio.



Compilación de un núcleo (2/4)

- ♦ Crear un archivo de configuración:
 - `cp /usr/src/sys/arch/i386/conf/GENERIC \ /root/CUSTOM`
 - `config -x /netbsd >/root/CUSTOM`
- ♦ Editar dicho archivo:
 - Eliminar lo que no sea necesario.
 - Cambiar parámetros estáticos.



Compilación de un núcleo (3/4)

- ♦ Con build.sh:
 - build.sh kernel=/root/CUSTOM
 - Al finalizar, anotar el directorio de compilación.
- ♦ Sin build.sh:
 - config -s /usr/src -b /usr/obj/CUSTOM \ /root/CUSTOM
 - cd /usr/obj/CUSTOM
 - make



Compilación de un núcleo (4/4)

- ♦ Instalación del núcleo:
 - Cambiar al directorio de compilación.
 - Ejecutar 'make install'.



Compilación de X11R6 (1/2)

- ♦ Fuentes en /usr/xsrc.
- ♦ Actualmente se usa XFree86 4.5.0.
- ♦ XFree86 no permiten la compilación cruzada.
- ♦ NetBSD proporciona /usr/src/x11:
 - Conjunto de Makefiles (*reachover build*).
 - Permiten la compilación cruzada.
 - Permiten el uso de directorios de compilación.



Compilación de X11R6 (2/2)

- ♦ A través de build.sh:
 - Usar la opción '-x'.
 - Añadir 'MKX11=yes' a /etc/mk.conf.
- ♦ Ejemplo:
 - `cd /usr/src && ./build.sh -x release`
- ♦ Nota: también se compila todo el sistema.



pkgsrc (1/14)

- ♦ Sistema de paquetes de NetBSD.
- ♦ Origen: ports de FreeBSD (1995 ?).
- ♦ Permite:
 - Compilación e instalación de programas de terceros.
 - Gestión de todos sus archivos.
 - Gestión automática de dependencias.
- ♦ Árbol centralizado de compilación de paquetes.



pkgsrc (2/14)

- ♦ Características:
 - Código limpio.
 - Consistencia entre paquetes.
 - Alta cohesión entre los paquetes.
 - Portabilidad a varios sistemas operativos.
- ♦ Terminología:
 - Paquete binario.
 - Paquete fuente.
 - Distfile.



pkgsrc (3/14)

- ♦ Estructura de /usr/pkgsrc:
 - Paquetes: /usr/pkgsrc/*categoría/paquete*.
 - Archivos comunes: /usr/pkgsrc/mk.
 - Código fuente de los programas: /usr/pkgsrc/distfiles.
 - Paquetes binarios: /usr/pkgsrc/packages.
 - Documentación: /usr/pkgsrc/doc.
 - *Bootstrap*: /usr/pkgsrc/bootstrap.



pkgsrc (4/14)

- ♦ Estructura de un paquete fuente:
 - DESCR: Descripción del paquete.
 - distinfo: *Checksums* del código fuente.
 - Makefile: Información de:
 - Compilación e instalación.
 - Dependencias necesarias.
 - Descarga del código.
 - Etcétera.
 - PLIST: Listado de archivos.



pkgsrc (5/14)

- ♦ Herramientas pkg_install:
 - Conjunto de programas.
 - Usados para instalar y mantener los paquetes instalados.
 - pkg_add, pkg_delete, pkg_info, etc.



pkgsrc (6/14)

- ♦ Pasos durante la instalación de un paquete fuente:
 - Descarga del código.
 - Verificación de su integridad.
 - Aplicación de parches.
 - Creación de *wrappers* y del directorio *buildlink*.
 - Configuración.
 - Compilación.
 - Instalación.
 - Verificación de la instalación.



pkgsrc (7/14)

- ♦ Instalación de un paquete fuente:
 - `cd /usr/pkgsrc/meta-pkgs/gnome`
`make && make install && make clean`
- ♦ Instalación de un paquete binario:
 - `pkg_add gnome-2.10.1.tgz`
 - `PKG_PATH=ftp://ftp.NetBSD.org/pub/...`
`export PKG_PATH`
`pkg_add gnome`



pkgsrc (8/14)

- ♦ buildlink3:
 - Problemática: interferencias entre paquetes.
 - Solución: simular un *chroot*.
 - Enlaces a cabeceras.
 - Enlaces a librerías.
 - Reescribir llamadas al compilador.

pkgsrc (9/14)

- ♦ wrappers:
 - Problemática: diferentes interfaces para utilidades similares.
 - Solución: usar *wrappers* que modifiquen las llamadas.
 - Útil para:
 - Llamadas al compilador.
 - Llamadas a utilidades del sistema.
 - Mejoras de portabilidad.



pkgsrc (10/14)

- ♦ pkgviews:
 - Cada paquete en un directorio.
 - Múltiples versiones concurrentes.
 - Creación de vistas.

pkgsrc (11/14)

- ♦ `pkg_comp`:
 - Compilación de paquetes dentro de un *chroot*.
 - Útil para:
 - Compilar paquetes para otras máquinas.
 - Compilar paquetes actualizados para la misma máquina.
 - Compilar paquetes para otras versiones de NetBSD.



pkgsrc (12/14)

- ♦ Uso en otros sistemas:
 - `cd pkgsrc/bootstrap`
`more README.`uname``
`./bootstrap`
`cp work/mk.conf.example /usr/pkg/etc/mk.conf`
`vi /usr/pkg/etc/mk.conf`
 - Usar `/usr/pkg/bin/bmake` en lugar de `make`.



pkgsrc (13/14)

- ♦ Uso no privilegiado:
 - `cd pkgsrc/bootstrap`
`./bootstrap --ignore-user-check`
`cp work/mk.conf.example ~/pkg/etc/mk.conf`
`vi ~/pkg/etc/mk.conf`
 - Usar `~/pkg/bin/bmake` en lugar de `make`.
 - O bien: `UNPRIVILEGED=yes` en `/etc/mk.conf`.



pkgsrc (14/14)

- ♦ Sistema de alternativas (pkg_alternatives):
 - Seleccionar entre utilidades similares.
 - Completamente opcional.
 - Ejemplos:
 - nvi/vim.
 - Máquinas virtuales de Java.
 - Intérpretes de Python.
 - Etcétera.



syspkg

- ♦ Sistema base distribuido mediante *tarballs*.
- ♦ Adaptación del sistema base a paquetes.
- ♦ Ventajas:
 - Distribución de actualizaciones binarias.
 - Eliminación de partes del sistema base.
- ♦ Aún en desarrollo.



Dónde obtener ayuda (1/2)

- ♦ Página web:
 - <http://www.NetBSD.org/>
 - <http://www.pkgsrc.org/>
- ♦ The NetBSD Guide:
 - <http://www.NetBSD.org/guide/en/>
- ♦ Listas de correo:
 - <http://www.NetBSD.org/MailingLists/index.html>



Dónde obtener ayuda (2/2)

- ♦ Estructura de las listas de correo:
 - netbsd-help, netbsd-users: Ayuda en general.
 - current-users: Preguntas sobre NetBSD-current.
 - tech-pkg: Preguntas sobre el sistema de paquetes.
 - Otras tech-*: propuestas técnicas.
 - port-*algo*: Preguntas sobre la plataforma *algo*.
 - Varias más.



Cómo reportar fallos (1/2)

- ♦ The NetBSD Project usa GNATS.
- ♦ *Problem Report* (PR): Informe de fallo.
- ♦ Usar `send-pr(1)`.
- ♦ Antes de enviar un PR:
 - Asegurarse que el fallo existe y es reproducible.
 - Comprobar que no haya sido reportado ya:
 - <http://www.NetBSD.org/Gnats/>



Cómo reportar fallos (2/2)

- ♦ Al rellenar el formulario del PR, incluir:
 - Cómo ocurrió el problema.
 - Versión concreta del sistema operativo.
 - Cómo reproducir el problema.
 - Por qué debe arreglarse (si el fallo no es obvio).
 - Un parche, si sabemos cómo arreglarlo.



Fin

Gracias por vuestra atención.

Turno de preguntas.

