

The NetBSD Project

Introducción a NetBSD

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

iParty 8
22 de abril de 2006



Contenido

- NetBSD vs. Linux.
- Un poco de historia.
- Objetivos.
- Política de versiones.
- pkgsrc.
- Cómo obtener NetBSD.
- Dónde obtener ayuda.
- Cómo reportar fallos.

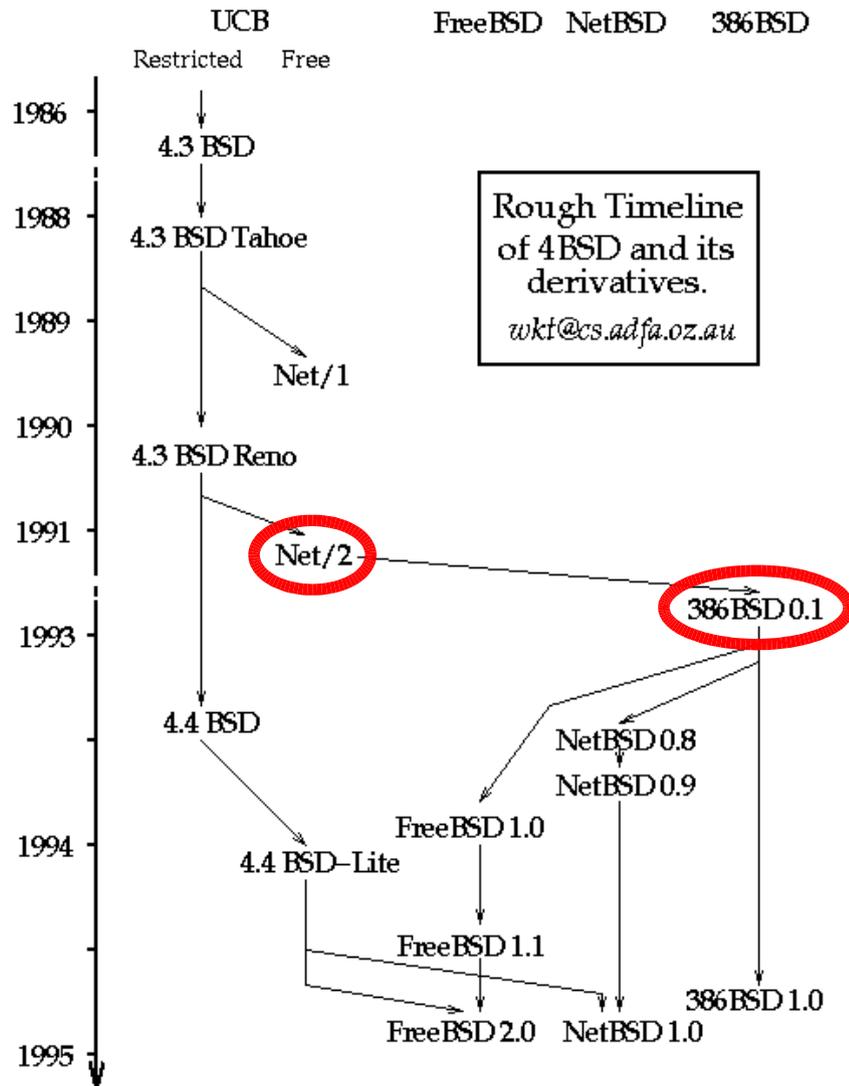


NetBSD vs. GNU/Linux

- Sistema completo.
- Basado en 4.4BSD.
- Licencia BSD.
- Núcleo = Linux.
- Aplicaciones de GNU.
- Escrito desde cero.
- Licencia GPL.



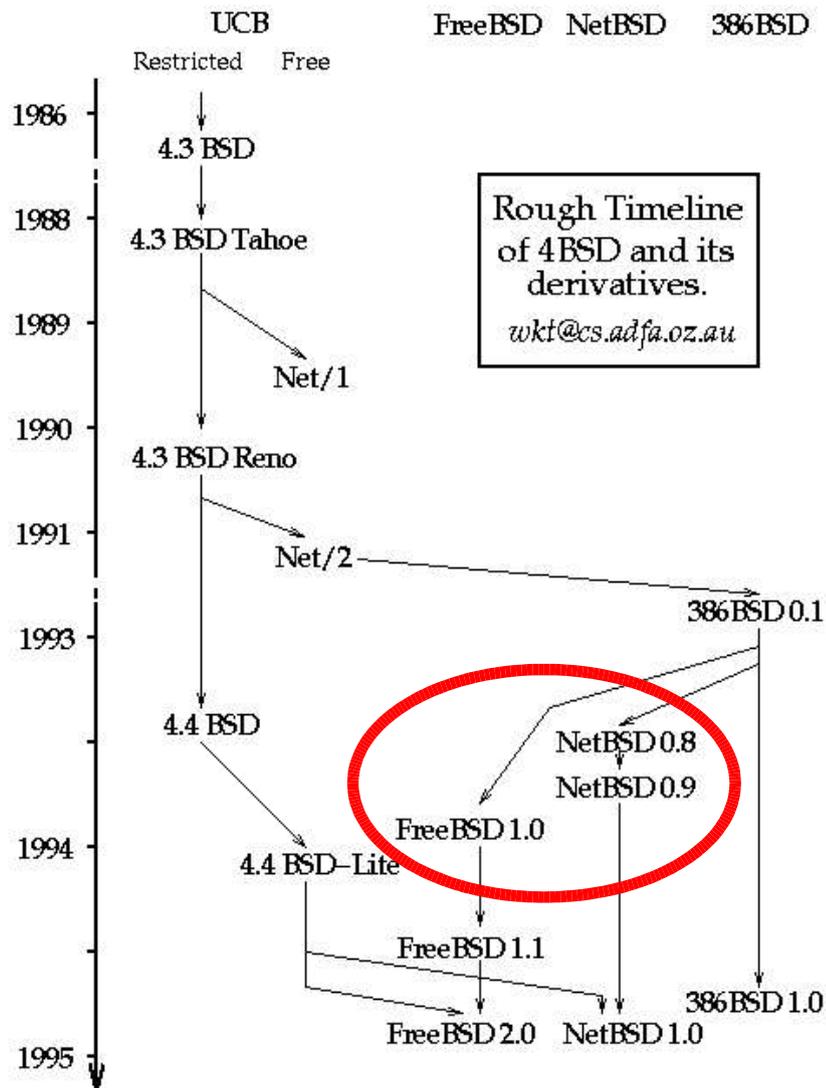
Un poco de historia (1/4)



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



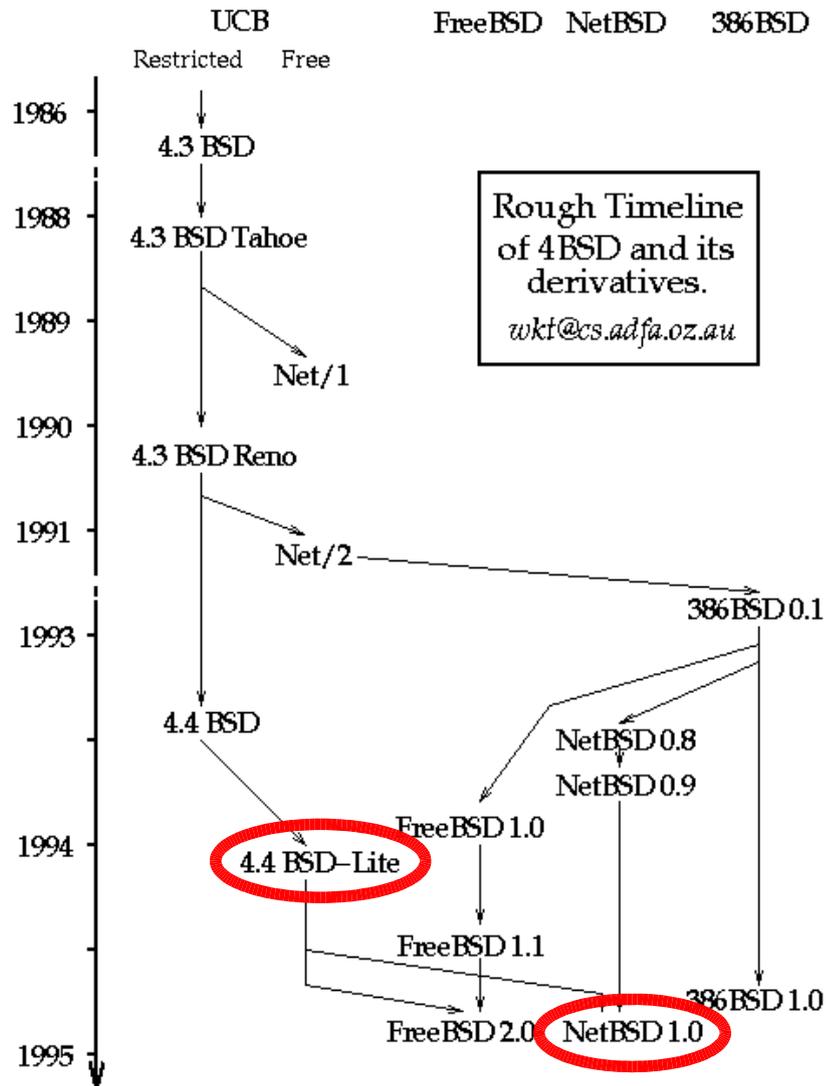
Un poco de historia (2/4)



- 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)

- Últimas versiones:
 - NetBSD 2.0.3, 31 de octubre de 2005.
 - NetBSD 2.1, 2 de noviembre de 2005.
 - NetBSD 3.0, 23 de diciembre 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.
 - Virtualización: Xen.
 - 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 (1/3)

- 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.



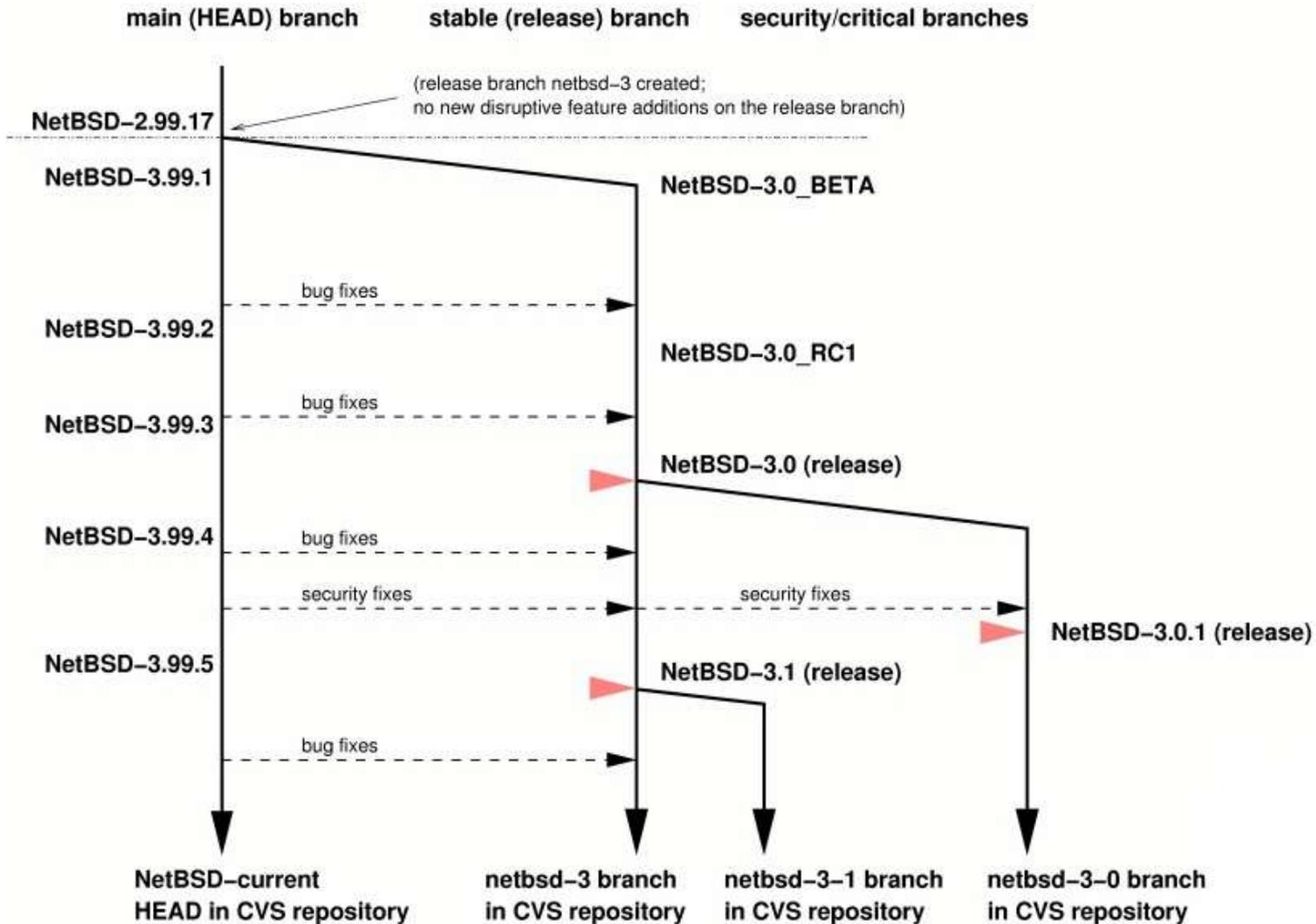
Política de versiones (2/3)

- 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 (3/3)

NetBSD release graph



pkgsrc

- Sistema de paquetes de NetBSD y DragonFly.
- Funciona en otras plataformas:
 - Linux, Windows, Mac OS X, otros BSDs...
- Instalación desde código fuente:
 - Similar a portage o los ports de FreeBSD.
 - Gestiona dependencias.
- Basado en make(1) y sh(1).
- Incluye GNOME, KDE, Apache, PostgreSQL...



Cómo obtener NetBSD (1/2)

- 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 (2/2)

- 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.



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-*algo*: Discusiones técnicas sobre *algo*.
 - 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 primera parte

¿Preguntas?



The NetBSD Project

Introducción al desarrollo de NetBSD

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

iParty 8
22 de abril de 2006



Contenido

- Cómo compilar el sistema.
- ¿Por qué ayudar?
- Summer of Code 2006.
- Objetivos.
- Áreas de trabajo y propuestas.



Compilación del sistema (1/3)

- 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 (2/3)

- `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 (3/3)

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



¿Por qué ayudar?

- Dar visibilidad al proyecto.
- Hacer que el proyecto sobreviva.
- Obtener experiencia en cualquier ámbito:
 - Portabilidad.
 - Redes.
 - Sistemas de ficheros.
 - Etc.
- Ganar reconocimiento.



Summer of Code 2006

- Iniciativa de Google.
- Oportunidad de:
 - Aprender cosas nuevas.
 - Ayudar a un proyecto que necesita voluntarios.
- Proyecto NetBSD-SoC:
 - Punto de encuentro para los proyectos escogidos.
 - <http://NetBSD-SoC.sourceforge.net/>
- ¿Qué se puede hacer?



Objetivos

- Tácticos:
 - A corto plazo.
 - Benefician a unos pocos usuarios.
 - No dan visibilidad al proyecto.
- Estratégicos:
 - Tal vez a largo plazo.
 - Pueden incrementar la popularidad de NetBSD.
 - Muy fácilmente visibles para los nuevos usuarios.



Sistemas de archivos (1/4)

- 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 (2/4)

- Fast File-System v2 (FFSv2):
 - Basado en FFS.
 - Originario de FreeBSD.
 - Soporte de discos y archivos muy grandes (64 bits).
 - Soporte para ACLs:
 - Llamadas al sistema disponibles.
 - Falta añadir soporte en FFSv2.



Sistemas de archivos (3/4)

- Otros:
 - Log-structured File-System (LFS).
 - De red: NFSv3, SMB, Coda.
 - En memoria: tmpfs, MFS.
 - Externos: FAT, NTFS, Ext2, ISO 9660.
 - Por capas: union, null, overlay, etc.



Sistemas de archivos (4/4)

- ¿Qué se necesita?
 - Sistema de archivos transaccional:
 - Rápida recuperación de fallos.
 - Posibilidades: mejorar FFS, portar algún sistema existente (ReiserFS, ZFS...) o crear uno nuevo.
 - Atributos extendidos:
 - Permiten soportar ACLs.
 - Mejoras como servidor de archivos.



Aceleración gráfica (1/2)

- NetBSD incorpora XFree86.
- Opcionalmente se puede usar X.Org (pkgsrc).
- No hay aceleración gráfica para OpenGL.
- Problemas:
 - Los juegos no pueden funcionar adecuadamente.
 - No se podrán usar los futuros escritorios (ej. Xgl).
 - Dificultad para usar aplicaciones de modelado.



Aceleración gráfica (2/2)

- ¿Qué hace falta?
 - Implementar DRI y DRM.
 - Mínimo: conseguir aceleración 3D con los controladores libres.
 - Posibilidad de portar los controladores binarios.



X.Org

- NetBSD usa XFree86 en la actualidad.
- Se quiere migrar a X.Org.
- ¿Qué falta por hacer?
 - Portar los controladores propios a XFree86.
 - Adaptar src/x11 para compilar X.Org.



Reescritura de pkg_install (1/2)

- Conjunto de aplicaciones para:
 - Instalar paquetes binarios y desinstalarlos.
 - Crear paquetes binarios desde pkgsrc.
 - Actualizar paquetes instalados.
- Problemas:
 - Mal diseño: código muy difícil de entender.
 - Dificultad para hacer ampliaciones.
 - Soporte deficiente para actualizaciones.



Reescritura de pkg_install (2/2)

- pkg_install tiene que reescribirse:
 - Debe ser compatible con la versión existente.
 - Nuevas funcionalidades:
 - Permitir actualizaciones automáticas.
 - Disponer de una biblioteca para manipular paquetes.
 - Integración con el instalador de NetBSD.
 - Interfaz gráfica.
 - No usar C:
 - Alternativas: C++, POSIX Shell...
 - A evitar: Perl, Python...



Montaje automático (1/2)

- Montar dispositivos automáticamente:
 - Al conectarlos.
 - Al acceder a un directorio concreto.
- Problemas actuales:
 - GNOME y KDE no son “amigables”.
 - Imposibilidad de autenticarse mediante llaves USB.



Montaje automático (2/2)

- ¿Cómo puede solucionarse?
 - Implementar notificaciones de conexión:
 - Añadir dispositivo virtual “hotplug” al núcleo.
 - Implementar demonio que actúe según las notificaciones.
 - Portar HAL:
 - Añadir soporte para el dispositivo desarrollado.
 - Hacer que GNOME use HAL donde sea posible.
 - Implementar devfs:
 - Permitir acceder a dispositivos usando nombres lógicos.



Gestión de alimentación

- ACPI se ha popularizado.
- Necesidad de poder suspender portátiles.
- Soporte incompleto actualmente.
- ¿Qué hay que hacer?
 - Implementar soporte para suspender en RAM.
 - Guardar y recuperar el estado del hardware.



Pruebas de regresión (1/2)

- Importancia:
 - Verificar que todo funciona como se espera.
 - Detectar errores lo antes posible.
 - Garantizar la máxima calidad de las distribuciones.
- Problemas:
 - Dificultad de ejecutar las pruebas existentes.
 - Faltan pruebas para muchas áreas del sistema.
 - Imposibilidad de automatizarlo todo.



Pruebas de regresión (2/2)

- ¿Posible solución?
 - Automatizar al máximo la ejecución de las pruebas.
 - Generar resúmenes con los resultados:
 - Pruebas ejecutadas, falladas, no ejecutadas...
 - Posibilitar la ejecución en una granja de servidores:
 - Pruebas de diferentes arquitecturas simultáneamente.
 - Evitar duplicación de código en las pruebas.



syspkg

- Sistema base distribuido mediante *tarballs*.
- Adaptación del sistema base a paquetes.
- Ventajas de tener syspkg:
 - Distribución de actualizaciones binarias.
 - Eliminación de partes del sistema base.
- ¿Qué falta por hacer?
 - Integración con el instalador (sysinst).
 - Facilitar las actualizaciones (pkg_install).
 - Declarar los *tarballs* obsoletos.



Aplicaciones multimedia

- V4L permite gestionar dispositivos de vídeo.
- NetBSD no tiene V4L:
 - Dificultad para portar aplicaciones.
 - Imposibilidad de competir con otros SO.
 - Actualmente se usa la interfaz bktr.
- ¿Cómo solucionarlo?
 - Diseñar una API para acceder a dispositivos de vídeo.
 - Añadir una capa de compatibilidad V4L sobre ella.



Soporte multiprocesador

- El soporte MP en NetBSD es muy simple:
 - *Giant lock*: Un único lock para proteger el núcleo.
 - `PTHREAD_CONCURRENCY > 1` no funciona bien.
- Las máquinas MP se están popularizando.
- NetBSD perderá competitividad si no se mejora.
- ¿Soluciones?
 - Usar distintos bloqueos para cada subsistema.
 - Arreglar el soporte de hilos en varias CPUs.



Nuevos ports

- Añadir soporte a G5 en macppc.
- Ayudar en el desarrollo de ia64.
- Portar a los nuevos Mac Intel.
- Portar a videoconsolas:
 - Game Cube.
 - Nintendo 64.
 - Etc.



Fin

Gracias por vuestra atención.

¿Preguntas?

