

# Die Zukunft: NetBSD

DR. ALEXANDRE WENNMACHER, NETBSD FOUNDATION

*Das NetBSD-Release 2.1 ist für Mitte September 2005, Release 3.0 für Oktober 2005 geplant. Dieser Beitrag gibt einen Überblick über die wichtigsten Neuerungen in NetBSD 3.0 und einen Ausblick darauf, was in NetBSD 4.0 zu erwarten ist.*

Nach dem Release von NetBSD 2.0 und vor dem Abzweigen von NetBSD 3.0 ging das NetBSD-Projekt zu einem moderneren Versionsnummern-Schema über. NetBSD-Versionen werden nun als  $M.m[p]$  dargestellt, wobei  $M$  die Major-Nummer,  $m$  die Minor-Nummer und  $p$  die Patch-Nummer ist. Neue Features sowie geänderte Kernel- und libc-APIs werden nur in Major Releases (wie zum Beispiel NetBSD 3.0) aufgenommen. Minor-Releases (wie NetBSD 2.1) dienen dem Anbringen von Korrekturen an Major-Releases, die Minor-Nummer zählt diese durch. Die mit Hilfe der Patch-Nummer indizierten Patch-Releases dienen lediglich dazu, kritische Patches anzubringen. Unter kritischen Patches versteht man Sicherheitspatches sowie Patches, die die Stabilität von Systemen gewährleisten oder wiederherstellen.

Neben den Releases von NetBSD gibt es natürlich auch noch die gemeinhin »NetBSD-current« genannte Entwicklungsversion. Um sie mit einer Versionsnummer zu charakterisieren, wurde die Minor-Nummer »99« reserviert. Für NetBSD-current wird die Patch-Nummer zum Durchzählen der Interface-Änderungen benutzt. Beispielsweise bezeichnet die Versionsnummer 3.99.4 die Version nach der vierten Interface-Änderung von NetBSD-current nach dem Branching von NetBSD-3.0.

Bild 1 zeigt schematisch die Gestaltung der Releases in den Jahren 2004 und 2005. Nach dem Release von NetBSD 1.6.2 wurde am 28. März 2004 NetBSD 2 abgezweigt. Nach diesem Datum wurde NetBSD-

current weiterentwickelt, Verbesserungen des NetBSD-2-Zweigs erfolgten hauptsächlich durch »Pullups«, also durch Transferieren von Codeänderungen von NetBSD-current nach NetBSD 2. Am 9. Dezember 2004 wurde dann NetBSD 2.0 veröffentlicht, nachdem die Release-Kandidaten hinreichend stabilisiert waren. Da das NetBSD-Projekt nur zwei Release-Zweige gleichzeitig pflegt, wurde das Lebensende (End-Of-Life, EOL) der NetBSD-1.5-Version kurz nach der Veröffentlichung von NetBSD 2.0 angekündigt.



Als Idealziel strebt das NetBSD-Projekt zwei Major-Releases pro Jahr an. Zu Beginn des Release-Zyklus werden die Ziele des Releases festgelegt, zum Beispiel, welche neuen Subsysteme aufgenommen werden sollen und welche privaten Entwicklungszweige in den NetBSD-current-Entwicklungszweig hineingemischt werden sollen. Dabei wird natürlich auch der Stand der Arbeiten und die Erreichbarkeit des Ziels mitbewertet. Das Abzweigen (Branchen) selbst wird zu einem Zeitpunkt durchgeführt, in dem der Entwicklungszweig sehr stabil ist. Danach übernimmt das Release-Engineering-Team die Pflege der Release-Branch. Wünscht ein Entwickler eine Änderung der Release-Branch, informiert er das Release-Engineering-Team davon über ein Ticket-Tracking-System. Das Release-Engineering-Team erledigt diese Anforderungen dann mit Hilfe der oben erwähnten Pullups.

## Neues im Kernel von NetBSD 3

Ein Blick auf Bild 1 zeigt, daß das gesteckte Ziel von zwei Releases pro Jahr nicht erreicht wird. Es ist derzeit realistischer, von einem Release pro Jahr auszugehen. Die erwähnten Releases 2.1 und 3.0 sind mit offenen Symbolen in Abbildung 1 eingezeichnet. Für den alten 1.6-er Zweig wird es wohl noch ein finales Release mit der Version 1.6.3 geben, bevor das Lebensende dieses Zweiges angekündigt wird. Für das 1.6.3-Release und das EOL von 1.6 steht noch kein Datum fest, man kann aber davon ausgehen, daß beides

relativ bald nach dem Release von 3.0 erfolgen wird.

Es folgt ein Überblick über die wichtigsten Neuerungen in NetBSD 3.0. Begonnen wird mit dem Kernel. Wired Memory, also vom Kernel benutzter Hauptspeicher, der nicht ausgelagert werden darf, wurde bisher im virtuellen Memory-System UVM so behandelt, als ob es ausgelagert werden könnte. Das verschwendete nicht nur Ressourcen, sondern verkomplizierte auch den Boot-Vorgang. Durch Änderungen an den *wom\_km\_\**-Routinen wurde das behoben. *uvm(9)* alloziert nun Map-Einträge im Kernel dynamisch anstelle von statisch, so daß die berühmt-berüchtigte *MAX\_KMAPENT*-Grenze nun obsolet ist.

Eine strukturelle Verbesserung gab es auch beim Audio-Support. Die Einführung einer Struktur *audio\_format* erlaubt es, die von den verschiedenen Soundkarten angebotenen Fähigkeiten zu abstrahieren. Damit ließ sich viel Code aus den jeweiligen Device-Treibern in das allgemeine Audio-Framework transferieren. Die im Kernel befindlichen Audio-Support-Funktionen zur Raten- und Format-Konversion hatten bisher unterschiedliche Interfaces. Durch eine Vereinheitlichung an dieser Stelle sind nun Filter-Pipelines im Kernel möglich. Außerdem wird High Definition Audio nun durch einen generischen PCI-Device-Treiber (*azalia*) unterstützt.

Zur Vorbereitung der Migration auf 64-Bit-Inodes wurde der *statvfs(2)* gemäß POSIX/XOpen eingeführt. *statvfs(2)* benutzt 64-Bit große Argumente, so daß Dateisysteme größer als zwei TByte von diesem Systemaufruf unterstützt werden. Für einige »Geräte«-Treiber kann das Konzept der klonbaren Treiber verwendet werden. Klonbare Geräte liefern als Resultat eines *open()*-Systemaufrufs eine neue Instanz des Geräts zurück. In 3.0 wurden nun vie-

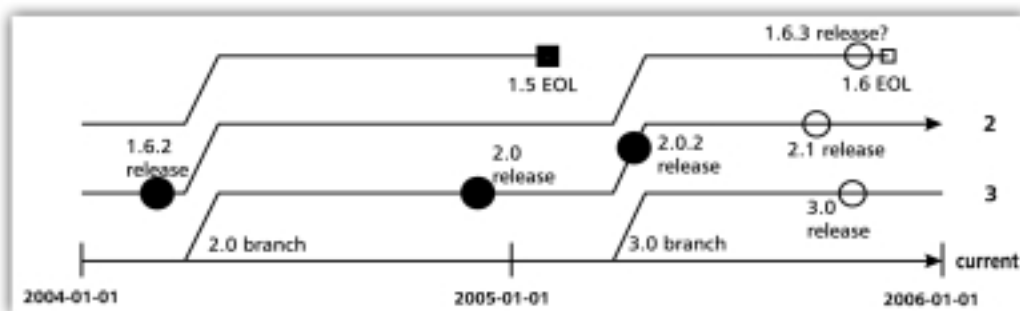


Bild 1: Release Engineering

le Gerätetreiber auf klonbare Geräte umgestellt, so zum Beispiel */dev/bpf*, */dev/ppp* und */dev/loop*.

Das maschinenunabhängige Konsolensystem *wscons* wurde in vielen Belangen verbessert. So sind nun Vorder- und Hintergrundfarbe (separat für Kernel- und Userland) dynamisch einstellbar, außerdem wird nun das bei den anderen BSD-Betriebssystem auch schon implementierte Scrollen unterstützt.

## Neue Gerätetreiber

Der Veriexec-Code, mit Hilfe dessen man erreichen kann, daß der NetBSD-Kernel nur Programme mit registrierten Prüfsummen ausführt, wurde weiter verbessert. Die Prüfsummen werden intern nun nicht mehr in einer doppelt verketteten Liste, sondern als Hashtabelle geführt, was das Auffinden beschleunigt. Wegen der bekannt gewordenen Schwäche von MD5-Prüfsummen werden nun verschiedene weitere Prüfsummen-Methoden unterstützt, unter ihnen RMD160, SHA1, SHA256, SHA384 und SHA512. Das Benutzer-Interface *veriexecctl(8)* wurde leicht modifiziert.

Dem *autoconf(9)*-System des Kernels, das unter anderem beim Booten benutzt wird, um Geräte zu detektieren und anzubinden (nicht zu verwechseln mit GNU *autoconf*), wurden Erweiterungen hinzugefügt, die auf eine bus-unabhängige Weise das Abkoppeln von Geräten und das Neuscannen des Busses erlauben. Diese Erweiterungen dienen vor allem zur Verbesserung der Unterstützung dynamisch ladbarer Gerätetreiber. NetBSD ist mit Hilfe seiner Emula-

tionsschicht in der Lage, Anwendungen vieler anderer Betriebssysteme auszuführen. In der Version 3.0 hinzugefügt wurde die *compat\_darwin* genannte Unterstützung für MacOS X.3 in der *macppc*-Architektur. In NetBSD 3.0 hinzugefügt wurde die Unterstützung für die Selective Acknowledgement für TCP, bekannt unter dem Namen TCP/SACK. Falls mehrere TCP-Pakete in einem Datenfenster verlorengehen, kann das bei herkömmlichen TCP zu erheblichen Leistungsproblemen führen. Normales TCP bestätigt empfangene Datenpakete nämlich kumulativ; es werden immer nur die im Datenfenster am weitesten links stehenden Segmente bestätigt. Das führt dazu, daß der Sender mindestens eine Round-Trip-Zeit warten muß, bis er feststellen kann, daß ein Paket verlorengegangen, oder er muß korrekt empfangene Pakete erneut versenden. Gehen mehrere Pakete verloren, geraten Sender und Empfänger aus dem Takt, was zu stark reduziertem Durchsatz führt. Mit Hilfe von TCP/SACK kann der Empfänger dem Sender alle empfangenen Daten quittieren, so daß der Sender lediglich die verlorengegangenen Pakete erneut schicken muß.

## Neuerungen im Netzwerk

Der TCP/IP-Stack unterstützt nun außerdem TSO (TCP Segmentation Offload). Die Idee bei TSO ist: Der TCP-Layer im Kernel gibt vor, daß das Netzwerkkarte eine riesige MTU (Maximum Transmission Unit) hat und übergibt deswegen sehr große Pakete über den Treiber an die Netzwerkkarte. Diese fragmentiert

das Paket, wobei es den ursprünglichen TCP-Header als Vorlage für die Fragmente nimmt. Der Vorteil bei dieser Methode ist, daß das Fragmentieren von der Hardware auf der Netzwerkkarte durchgeführt und die CPU dadurch entlastet wird. In ähnlicher Weise wird bereits seit längerem die Prüfsummenberechnung von Netzwerkpaketen an solche Netzwerkkarten, die das unterstützen, delegiert.

Ebenfalls hinzugefügt wurde die Unterstützung für das Protokoll EtherIP, mit dem man Ethernet-Pakete über IP routen kann. Das kann ganz praktisch sein, wenn man zum Beispiel zwei LANs, auf denen XNS benutzt wird, über ein IP-WAN verbinden möchte.

## Sicherheit

Weitere Neuerungen im Bereich des Netzwerks sind außerdem die Unterstützung für das PIM-SM-Routing-Protokoll. Das wurde mit Code implementiert, der vom Projekt XORP abgeleitet ist. Die Unterstützung für IPsec NAT Traversal (NAT-T, siehe RFC 3947 und 3948) ist neu.

Zusätzlich wurde ein virtuelles, klonbares Ethernet-Gerät mit dem Namen *tap(4)* dem System hinzugefügt. Dieses Gerät stellt sich dem Kernel genau wie ein reales Ethernet-Gerät dar, es kann aber auch vom Userland aus darauf zugegriffen werden, zum Beispiel, um vom System gesendete Ethernet-Frames zu lesen. Nun sind Pluggable Authentication Modules (kurz PAM) in der Form von OpenPAM in NetBSD 3.0 integriert. PAM stellt modularisierte Bibliotheksfunktionen für Authentifizierungsaufgaben zur Verfügung, die nicht nur von den Standard-NetBSD-Kommandos, sondern auch von anderen Diensten und Anwendungen benutzt werden können. Mit Hilfe einer oder mehrerer Konfigurationsdateien kann das Verhalten von Anwendungen, die das PAM-API benutzen, vom Systemadministrator festgelegt werden. Auf diese Weise kann den NetBSD-Authentifikationskommandos wie *passwd(1)* auch ein neuer

Authentifikationsmechanismus hinzugefügt werden.

Die mitgelieferte X11-Distribution wurde auf XFree86 4.5.0 aktualisiert, das zahlreiche Verbesserungen und Unterstützung für neue Hardware bietet. Für die Zukunft erwägt das NetBSD-Projekt einen Wechsel auf die X11-Distribution des X.org-Projekts, in das zur Zeit mehr technische Neuerungen einfließen.

Zusätzlich zum bekannten Paketfilter *ipf* wurde der von OpenBSD stammende Filter *pf* in das Basissystem integriert. NetBSD verfügt damit über zwei leistungsfähige Paketfilter, der Administrator kann sich den ihm am meisten zusagenden aussuchen. Der i386-spezifische Gerätetreiber *ichlpcib(4)* unterstützt nun die Enhanced-SpeedStep-Technologie, mit der der Prozessortakt nach Bedarf über eine relativ weite Spanne geregelt werden kann. Dies erlaubt es, zum Beispiel in Laptops den Prozessor im Leerlauf mit niedriger Taktfrequenz und damit mit niedrigem Stromverbrauch zu betreiben, um die Lebensdauer der Akkus zu erhöhen.

Abschließend wagen wir noch einen Blick in die Zukunft. Es sind folgende Neuerungen geplant, aber es ist noch zu früh um zu sagen, welche der Neuerungen es nach NetBSD 4.0 schaffen werden. Zuerst ein Ausblick auf den Kernel:

- Die Unterstützung von Dateisystemen größer als 2TByte soll vollendet werden. Dazu müssen die ganz neuen »Wegdes«, eine Partitions-Abstraktion, 64-Bit-Inodefähig gemacht werden, ebenso die Dateisystem-Tools.
- Die aktuelle Unterstützung von Firewire läßt zu wünschen übrig. Daher ist geplant, den Firewire (IEEE 1394)-Code von FreeBSD zu importieren (er befindet sich nun bereits in NetBSD-current).
- Das »Hotpluggen« von Geräten soll auf eine generische Weise unterstützt werden (die Betonung liegt auf »generisch«, zum Beispiel wird für USB bereits seit längerem Hotplugging unterstützt).
- Auch die Audio-Verbesserungen

sollen fortgesetzt werden. Hier ist insbesondere daran gedacht, das Mischen von Audioströmen direkt im Kernel vorzunehmen. Das ist wünschenswert, da Userland-Lösungen unter dem Problem großer Verzögerungen leiden.

- Der NetBSD-Scheduler ist »uralt« und soll durch eine modernere Variante ersetzt werden. Um eventuell unterschiedliche Scheduler auf Ein- und Mehrprozessormaschinen einsetzen zu können, wird an Scheduler-Module gedacht.
- Der Parallel-Port soll nicht nur als Gerät (*lpt*), sondern auch als Bus betrieben werden können, was zum Beispiel die Nutzung preiswerter Backup-Geräte wie ZIP-Drives erlauben wird, die über den Parallel-Port angeschlossen werden.

## Pläne für NetBSD 4.0

Im Bereich Netzwerk ist wohl als wichtige noch ausstehende Aktion die NFSv4-Integration zu nennen. Bei der Administration und im Userland ist mehr zu tun:

- Auf mittlere Sicht sollen neue Versionen des GNU-C-Compilers und des GNU-Debuggers importiert werden. Ob für 4.0 bereits ein gcc-Compiler der Version 4 eingesetzt wird, ist noch unklar. Gdb-Version 6.3 (oder später) wird für das Debuggen von Anwendungen benötigt, die Multithreading benutzen.
- Bibliotheken wie die C-Bibliothek sollen die noch fehlenden Routinen ergänzt werden, die re-entrant sind (zum Beispiel *get\*ent\_r* und *getXbyY\_r*).
- Wie bereits erwähnt wird einen Wechsel auf die X11-Distribution des X.org-Projekts erwogen.
- Die bereits seit langem geplanten System Packages, mit denen man das Betriebssystem modular (wie von *pkgsrc* bekannt) installieren kann, sollen nun wirklich bald implementiert werden.
- Schlußendlich gibt es den Plan, die in NetBSD integrierten Regressions-Tests weiter zu verbessern. ◆