

Ein komplettes Betriebssystem

NetBSD benutzt zur Paketverwaltung `pkgsrc`, das nicht nur auf vielen Unixen (auch Linux) läuft, sondern auch immer für neue Plattformen portiert wird. Einige Betriebssysteme, z.B. DragonFly BSD, benutzen `pkgsrc` als ihre Standardpaketverwaltung, und vergrößern damit die Entwickler- und Benutzerbasis von `pkgsrc`.

Man kann NetBSD minimal konfigurieren, sodass es nur wenige Megabyte verbraucht, aber auch so viel Software nachinstallieren, wie man will. Gerade was Firewall-Lösungen betrifft, bringt NetBSD mehr mit als alle anderen freien Unixe. Weitere Sicherheitsfeatures wie `verexec`, das nur die Ausführung von registrierten Binaries erlauben, machen NetBSD nicht nur für Server, sondern auch für embedded Umgebungen tauglich.

Ob man einen KDE-, Gnome- oder Xfce-Desktop haben will, alles findet man in aktueller Version auch für NetBSD. Fast jede Open Source Software gibt es auch für NetBSD, und wenn sie nicht vorhanden ist, kann man eine Anfrage erstellen, dass sie portiert wird oder es gleich selber machen. Außerdem kann man bei einigen Anwendungen auch auf eine Emulation von Linux oder FreeBSD zurückgreifen. Diese ist fast ohne Geschwindigkeitsverlust benutzbar, da nicht virtualisiert wird, sondern nur die Syscalls entsprechend umgesetzt werden.

Auch wenn man seine Pakete selber kompilieren kann (was auf alten Rechnern lange dauern kann), kann man auch die Repositories benutzen, um vorkompilierte Pakete mit dem Befehl `pkg_add(1)` zu installieren. `pkg_add` löst Abhängigkeiten automatisch auf und installiert und registriert das Paket im System, als wenn es mit `pkgsrc` installiert kompiliert worden wäre.

NetBSD ist Freie Software (BSD-Lizenz), aber `pkgsrc` enthält auch Pakete, die nicht BSD-lizenziert sind. `pkgsrc` gibt dem Administrator volle Kontrolle darüber, welche Lizenzen er akzeptiert und welche nicht, mit einer mitgelieferten Liste von gängigen freien Lizenzen.

Eine aktive Community

NetBSD hat eine sehr aktive, aber ziemlich zentral organisierte Community. Die NetBSD Foundation stellt nicht nur die Webseiten und die Infrastruktur zur Entwicklung und Verteilung von NetBSD bereit, sondern auch Mailinglisten zur Kommunikation von Usern und Entwicklern. Mit einer Mail an diese Liste erreicht man einen Großteil aller User und bekommt schnell Hilfe. Es gibt Listen für allgemeine und Sicherheits-Ankündigungen, für `pkgsrc`, etc.

Weiterhin hat auch NetBSD in den größeren IRC-Netzen wie Freenode, IRCnet und EFnet Channels (meist `#netbsd` genannt), in denen man Kontakt zu anderen NetBSD-Usern, häufig auch Entwicklern, rund um die Uhr aufnehmen kann. Man sollte nur evtl. ein wenig Geduld mitbringen.

NetBSD-User sind weltweit gestreut, daher gibt es regionale Mailinglisten (`regional-de@NetBSD.org`), auf denen man deutschsprachige Hilfe findet. Alternativ gibt es auch bei allen größeren Open Source-Events (Froscon, Linuxtag, Chemnitzer Linuxtage, etc.) einen NetBSD-Stand, an dem man sich Hilfe holen oder über aktuelle Entwicklungen informieren kann.

Einfache Teilhabe

Die NetBSD-Community ist sehr eng verbandelt, auch als Einsteiger hat man schnell direkten Kontakt mit Entwicklern. Eingesendete Code-Vorschläge werden schnell diskutiert und bearbeitet (ggf. eingepflegt, oder Probleme anderweitig gelöst), v.a. per Bug Report oder Vorschlag per Mailingliste.

Wer an NetBSD mitarbeiten will, findet auf den Mailinglisten oder im IRC schnell Entwickler, die Überblick haben und Vorschläge liefern, an was gearbeitet werden kann und auch selber Hilfe anbieten, falls man mit der Entwicklung nicht weiter kommt.

Natürlich muss man nicht Code schreiben, um mitzumachen. Es gibt viele sehr wichtige Arbeiten, bei denen man auch ohne Coden helfen kann, z.B. Dokumentation, Webseitenpflege und Tests.

Das NetBSD-Projekt

“OF COURSE IT RUNS NETBSD”



Was ist NetBSD?

NetBSD ist eines der ältesten, noch aktiven freien Unixe. Dabei hat es seine Wurzeln im originalen Unix von Bell, und seinen Ursprung im universitären Bereich, aus dem es sich 1993 in eine offene Entwicklung löste; aber diese professionelle Ausrichtung hält sich bis heute. Die Entwicklung findet hauptsächlich im Internet statt, daher der Name *Net*BSD.

“Of course it runs NetBSD” – NetBSD achtet sehr stark auf die Sauberkeit und Portabilität des Codes. Es läuft auf vielen Plattformen, die andere Betriebssysteme schon vor Jahren abgeschrieben haben, sei es ein kleiner HP Jornada Palmtop oder ein 40kg schwerer DEC Alpha-Server – auf allen kann man ein aktuelles NetBSD mit aktueller Software installieren.

Gerade für moderne Plattformen ist NetBSD gut geeignet und stellt ein modernes, aber klassisches universelles Unix, das als Desktop-, als Server-, aber auch als Embedded-Betriebssystem benutzt wird.

Im folgenden wird der Begriff *Linux* äquivalent zum Begriff *GNU/Linux* benutzt.

NetBSD ist Kontinuität

Verglichen mit anderen freien Unixes, hat NetBSD ein relativ *konservatives* Entwicklungsmodell, Entscheidungen werden erst nach reichlicher Diskussion getroffen, und Standards und Konventionen werden so weit wie praktisch möglich befolgt.

NetBSD legt seinen Fokus auf Qualität, Stabilität, Portabilität und Sicherheit, wobei diese Aspekte wichtiger sind als neue Features und Techniken, die evtl. noch unzureichend getestet sind, zu implementieren. Implementierte Features, die sich als nicht sinnvoll erweisen, ein Sicherheitsrisiko darstellen oder in der Umsetzung mangelhaft oder redundant sind, können dann auch wieder entfernt werden, wie bspw. *Systrace*.

Auch wenn man nur ein aktuelles x86-System benutzt, profitiert man von dem Ziel, auf möglichst vielen Plattformen und Architekturen zu laufen. Der Code ist dadurch sehr sauber und portabel geschrieben und gut dokumentiert, wie auch das restliche Betriebssystem.

NetBSDs Entwicklungsmodell macht es möglich, dass moderne Features auch auf alten Rechnern benutzt werden können, wo andere Betriebssysteme hardware-spezifischen Code schreiben (müssten). Erst kürzlich wurde die neue Firewall `npf(3)` entwickelt, die die Nachteile der bisherigen Paketfilter ausgleichen soll, die Userland-Virtualisierungslösung `rump(3)` und ein neuer `apropos(1)`-Ersatz wurden entwickelt, und *Lua* in den Kernel integriert. All diese Aspekte machen nicht nur die aktuellsten Rechner, sondern auch alte vergessene Maschinen den Branchenriesen auf aktuellen Servern gegenüber konkurrenzfähig.

Vor allem auch die Einheitlichkeit von Kernel und Userland grenzen NetBSD gegenüber Linux ab: Während bei Linux der Kernel, viele GNU-Teile, 3rd-Party-Programme und -Treiber zusammen arbeiten, ist NetBSD ein komplettes Betriebssystem mit Kernel und (fast) vollständigem Userland, die beide von einem Projekt verwaltet werden. Dies garantiert, dass die Komponenten zusammen arbeiten und sich einheitlich bedienen lassen, und auch die Dokumentation lässt sich gut strukturieren und durchsuchen.

Für alle Plattformen

NetBSDs Slogan ist "Of course it runs NetBSD", und der ist Programm. Auf über 10 Architekturen und über 55 Plattformen aus einer einzigen Codebasis bietet NetBSD ein aktuelles Betriebssystem mit gleichem Userland, gleichem Paketmanagement und aktiver Entwicklung. Man erhält dabei auf alten Rechnern nicht etwa eine abgespeckte Variante, sondern immer das volle Betriebssystem mit allen Features.

Genau aus diesem Grund wird NetBSD sehr häufig für Embedded-Lösungen verwendet: Die Portierung auf eine neue Plattform ist oft nur Anpassung von Treibern, und der Code macht es einem möglichst einfach, nur den hardware-spezifischen Code schreiben zu müssen und keine anderen Teile des Systems.

NetBSD bietet außerdem als Paravirtualisierungslösung Xen als Host-, wie auch als Gastsystem an, sodass man andere Betriebssysteme virtualisieren oder NetBSD selber als Gast auf anderen Betriebssystemen laufen lassen kann.

Einfache Möglichkeit zum Crosskompilieren

NetBSD läuft auf vielen alten Plattformen, die oft zu leistungsschwach sind, größere Mengen Code in endlicher Zeit zu kompilieren. Daher hat NetBSD ein sehr ausgereiftes und komfortables System zum Crosskompilieren des ganzen Betriebssystems oder auch nur der Toolchains selber.

Man kann auf fast jedem Unix (inkl. MacOSX) und jeder Plattform NetBSD und Toolchains für jede unterstützte Plattform bauen, auch als unprivilegierter User. Das Skript `build.sh` baut z.B. die Toolchain für SPARC mit dem Aufruf `build.sh -m sparc tools`, und durch Auswechseln des Ziels (also `build.sh -m sparc distribution`) kann man auch gleich das ganze Betriebssystem ohne weitere Schritte kompilieren. Auf genau diese Weise werden auch die offiziellen Releases von den ftp-Servern gebaut.

Vollständige Dokumentation

NetBSD hat, wie die meisten BSDs, eine sehr gute und vollständige Dokumentation.

Es gibt ein zentrales (freies) NetBSD-Buch, den *NetBSD Guide* (siehe Links). Er enthält detaillierte Anleitungen zur Installation und Einrichtung von NetBSD, zur Installation von Software, sowie allgemeine Administrationsanweisungen. Er enthält auch eine genaue Beschreibung, wie man den Kernel und das System selber kompiliert und erklärt wichtige Subsysteme wie z.B. das Netzwerk, Software-Raid, Festplattenverschlüsselung, etc.

Zum anderen ist NetBSD fast vollständig mit Manpages durchdokumentiert. Programme, Kerneltreiber, Syscalls, Systembibliotheken, Dateiformate, spezielle Dateien, Kernel-Interns - alle haben ihre eigene Webseite. Durch diese Dokumentation ist NetBSD besonders gut für Entwickler und fortgeschrittene User geeignet, die Informationen zum System schnell finden wollen, ohne im Web zu suchen, aber auch für Anfänger, die das System und das Leben auf der Kommandozeile kennen lernen wollen.

Auch für Nicht-NetBSD-Benutzer kann die Dokumentation nützlich sein. Die Grundmechanismen von Unix (z.B. einem `vfs`) und Standard-C-Bibliotheken sind ausführlich beschrieben, und die NetBSD-Installationshinweise für einige ältere Plattformen sind auch allgemein als Hardware-Referenzen weit verbreitet.

Einstiegspunkte

Webseite: <http://www.NetBSD.org/>

Guide: <http://www.NetBSD.org/docs/guide/en/>

Artikel: <http://www.NetBSD.org/docs/>

Mailinglisten: <http://www.NetBSD.org/maillinglists/>

Manpages: <http://man.NetBSD.org/>