Ein komplettes Betriebssystem

NetBSD verwendet das Paketmanagement `pkgsrc`, das nicht nur auf fast allen Unix-/Linux-Plattformen funktioniert, sondern gerade auch NetBSD immer mit aktuellen Paketen versorgt. DragonFlyBSD und MirOS haben aufgrund der Portabilität auch pkgsrc als eigenes Paketmanagement übernommen und vergrößern so nochmal die Benutzer- und Entwicklerbasis von pkgsrc.

Man kann NetBSD so minimal konfigurieren, 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 `veriexec`, 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 diese nicht für NetBSD vorhanden ist, dann kann man immer noch die integrierte Emulation von Linux, FreeBSD oder vielen anderen Systemen zurückgreifen. Diese ist dabei fast ohne Geschwindigkeitsverlust benutzbar, da nichts virtualisiert wird, sondern nur die Syscalls entsprechend umgesetzt werden.

Natürlich muss man nicht alle Pakete selber kompilieren, was auf alten Rechnern oft auch schwer fallen würde: Man kann auf zahlreiche Repositories zurückgreifen und sich Binärpakete bequem mit dem Kommando `pkg_add` innerhalb von Sekunden (oder auch Minuten, je nach Paketgröße) installieren. Alle Abhängigkeiten und Konflikte werden dabei automatisch aufgelöst bzw. dem User zur Auswahl angezeigt, und sind mit wenigen Handgriffen lösbar.

NetBSD gibt einem komplette Kontrolle über die Lizenzen der verwendeten Pakete. Während das Basissystem selber nur aus BSD-lizensiertem Code besteht, kann man für zusätzliche Software in der Konfigurationsdatei `mk.conf` kann man genau festlegen, welche Lizenzen für das System erlaubt sind und welche nicht.

Vollständige Dokumentation

NetBSD hat, wie alle anderen BSDs auch, eine sehr gute und vollständige Dokumentation.

Es gibt ein zentrales (freies) NetBSD-Buch, den NetBSD Guide (siehe Links). Er enthält eine detaillierte Anleitung zur Installation und Einrichtung von NetBSD, wie man Software dazuinstalliert, allgemein das System administriert, aber auch alle wichtigen Subsysteme wie z.B. das Software-RAID, Bluetooth oder die Festplattenverschlüsselung sind dadrin ausführlich beschrieben.

Zum anderen ist NetBSD fast vollständig mit Manpages durchdokumentiert. Zu jedem Programm und jedem Kerneltreiber gibt es eine Manpage, die die Benutzung, den Sinn und die Funktionsweise beschreibt. Damit ist NetBSD vor allem auch für Entwickler gut geeignet, die nicht lange nach Dokumentation suchen, sondern diese schnell und einfach mit einem Tool lokal finden wollen. Auch wurde gerade in einem GSoC-Projekt das neue Tool XXX zur besseren Durchsuchung von Manpages entwickelt. Auch im Vergleich mit den Linux-Manpages (für generische Unix-Systeme) sind die BSD-Manpages oft vollständiger und ausführlicher.

Doch auch für Anfänger ist NetBSD sehr gut geeignet. Das System gibt einem alle Möglichkeiten an die Hand, Probleme selber zu finden und mit der Dokumentation zu lösen.

Auch für Nicht-NetBSD-Benutzer hat die Dokumentation oft großen Nutzen. Die Grundmechanismen von Unix (z.B. einem vfs) sind gut beschrieben, und die NetBSD-Installationshinweise für einige ältere Plattformen sind auch allgemein als Hardware-Referenzen weit verbreitet.

Einstiegspunkte in die Dokumentation

Webseite: `http://netbsd.org/`
Guide: `http://netbsd.org/docs/guide/en/`
Artikel: `http://netbsd.org/docs/`
Mailinglisten: `http://netbsd.org/mailinglists/`
Manpages: `http://man.netbsd.org/`

Das NetBSD-Projekt

"Of course it runs NetBSD"

Was ist NetBSD?

NetBSD ist eines der ältesten, noch aktiven freien Unixe. Anders als Linux hat es dabei seine Wurzeln im originalen Unix von Bell, und seinen Ursprung im universitären Bereich, aus dem es sich 1993 in eine offene Entwicklung löste; und diese professionelle Ausrichtung hält sich bis heute.

"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.

Aber gerade auch für moderne Plattformen ist NetBSD gut geeignet und stellt ein modernes, aber klassisches universelles Unix, das als Desktop- und auch als Server-Betriebssystem benutzt werden kann.

NetBSD ist Kontinuität

Verglichen mit Linux, aber gerade auch im Vergleich zu anderen BSDs ist NetBSD ein sehr *konservatives* Betriebssystem. Das bedeutet nicht etwa Rückständigkeit, sondern nur, dass Entwicklungen mit Bedacht ausgewählt werden. NetBSD folgt wie wohl kein anderes freies Unix den klassischen Unix-Traditionen und -Prinzipien, und erreicht damit eine sehr einheitliche Bedienung.

Wo andere Betriebssysteme oft so viele Features wie möglich in ein Programm implementieren, achtet man bei NetBSD mehr als bei allen anderen darauf, dass diese Features auch sinnvoll und nicht redundant sind, und dass das System stabil läuft und keine ungare Software aufgenommen wird. Implementierte Features gammeln dann aber nicht einfach ungepflegt weiter im Code rum, sondern können nach einer Weile auch wieder rausfliegen, wenn sie sich nicht als sinnvoll erwiesen, sie ein Sicherheitsrisiko darstellen oder ihre Umsetzung mangelhaft oder redundant ist.

NetBSD hat den Anspruch, auf jeder Hardware-Plattform zu laufen, und auch wenn man nur ein aktuelles x86-System benutzt, profitiert man davon. Der Code ist dadurch sehr sauber und portabel geschrieben und auch immer gut dokumentiert, damit jeder in den Code gut einsteigen kann.

Trotzdem wird dadurch keine Entwicklung gebremst, dies macht es nur möglich, moderne Features auch auf alten Rechnern zu benutzen. Erst kürzlich wurde eine neue Firewall – `npf` – entwickelt, die die Nachteile von bisherigen Firewalls ausgleichen soll, oder einzigartige Userland-Virtualisierungslösungen wie `rump` 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. Wo bei Linux viele kleine Einzelteile irgendwie zusammen arbeiten, stecken bei NetBSD Entwicklungen, die aus einer Hand kommen, deren Zusammenarbeit und Einheitlichkeit garantiert ist.

Für alle Plattformen

NetBSDs Slogan ist “Of course it runs NetBSD”, und der ist Programm. Auch wenn Linux viele Plattformen unterstützt, hat man dabei keine einheitliche Plattform, jede Distribution für eine spezielle Plattform ist anders. NetBSD bietet auf über 10 Architekturen und über 55 Plattformen ein aktuelles Betriebssystem mit einheitlichem (gleichem) Userland, gleichem Paketmanagement und aktiver Entwicklung. Man erhält dabei 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, oder die Plattform existiert sogar schon. Und auch falls es eine neue Plattform sein sollte, so macht es der Code einem möglichst leicht, diesen auch für neue Hardware anzupassen.

NetBSD bietet außerdem als Virtualisierungslösung Xen als Host-, wie auch als Gastsystem an, sodass man andere Betriebssysteme virtualisieren oder selber als virtuelle Maschine auf anderen gehosteten Rechnern laufen lassen kann.

Einfache Möglichkeit zum Crosskompilieren

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

Man kann generell auf fast jedem Unix (inkl. MacOSX) und jeder Platfform NetBSD und Toolchains für jede unterstützte Plattform bauen, auch als unprivilegierter Nutzer. Das Skript `build.sh` erledigt das Bauen einer Toolchain für sparc z.B. mit einem Aufruf `build.sh -m sparc tools`, und durch Auswechseln des Ziels (also `build.sh -m sparc distribution`) kann man auch gleich das ganze Betriebssystem kompilieren, ohne weitere Schritte unternehmen zu müssen.

Eine aktive Community

NetBSD hat eine sehr aktive, relativ zentral organisierte Community. Die NetBSD Foundation stellt nicht nur die Webseiten und die Infrastruktur zur Entwicklung und Verteilung von NetBSD, sondern auch die Mailinglisten, über die die Kommunikation der NetBSD-User abläuft. Mit einer Mail an diese Liste erreicht man einen Großteil aller User und bekommt schnell Hilfe.

Weiterhin hat auch NetBSD in den großen IRC-Netzen (v.a. im IRCNet und Freenode) Channels (netbsd), in denen man Kontakt zu anderen NetBSD-Usern aufnehmen kann und fast rund um die Uhr Hilfe bekommt.

Auf den Mailinglisten wie auch im IRC trifft man häufig Entwickler an, sodass für sehr spezielle Probleme manchmal auch gleich der Verantwortliche die Probleme mitliest.

NetBSD-User sind weit 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).

Wer an NetBSD mitarbeiten will, findet auf den Mailinglisten oder im IRC schnell Kontakt zu Entwicklern, 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 selber Code schreiben, um teilzuhaben. Es gibt viele sehr wichtige Arbeiten, bei denen man auch ohne großes technisches Vorwissen helfen kann, vor allem, was die Dokumentation betrifft. Auch hier kann man einfach nachfragen, was es zu tun gibt, Arbeit lässt sich meist schnell finden.