

## Die BSD-Familie und ihre Schwerpunkte

### FreeBSD - <http://www.FreeBSD.org/>

Auf normalen PCs ist FreeBSD das weitestverbreitete BSD. Dies liegt daran, daß es die breiteste Hardware-Unterstützung für diese Plattform bietet und dafür am besten optimiert ist. Es weist die meisten Features auf und hat die größte Anzahl an Nutzern und Entwicklern. Mehr als bei anderen BSDs wird Wert auf eine einfache Bedienung und Konfiguration gelegt, speziell für Einsteiger. Zudem bietet es mit über 14.000 Programmen, die sogenannten "Ports", die größte Sammlung an Applikationen.

FreeBSD betrachtet sich nicht nur als reines Serversystem, sondern es zielt bewußt auch auf Endbenutzer, inklusive Umsteiger von Windows und Linux. Unterstützt werden auch amd64, Alpha, Itanium, SPARC, PPC (Mac) usw. FreeBSD bietet sehr gute Unterstützung für Mehrprozessorsysteme (SMP). Mittlerweile gibt es Projekte, welche speziell auf Desktop-Nutzer abzielen und eine leichte Installation, Konfiguration und Wartung ermöglichen: DesktopBSD und PC-BSD. Zusätzlich steht mit FreeSBIE eine Live-CD zur Verfügung, welche auch als Installationsmedium dienen kann. FreeBSD wird in sehr vielen Unternehmen eingesetzt und erfährt breite Unterstützung durch namhafte Firmen wie Yahoo und andere.

### NetBSD – <http://www.NetBSD.org/>

NetBSD wurde 1993 mit dem Ziel gegründet, die Multiplattformfähigkeiten des damaligen Berkeley Unix (BSD) fortzuführen. Daraus resultiert auch seine herausragendste Eigenschaft, auf einer grossen Zahl von Hardware-Plattformen zu laufen - NetBSD unterstützt mit über 50 Plattformen mehr Architekturen aus einem Quellcode als jedes andere Betriebssystem, woraus auch der Wahlspruch des Projektes resultiert: Of course it runs NetBSD!

NetBSD hat einen Fokus: Korrekten Code auf Basis von abstrahierten System-Interfaces generieren, der überall sicher, stabil und performant läuft. Oft fällt die Wahl für NetBSD dabei sehr leicht, da es einfach keine Alternativen auf manchen Hardware-Plattformen gibt. Diese reichen von Embedded Devices und Handhelds über moderne Desktop-PCs bis hin zu 64bit Opteron-Systemen mit mehreren CPU-Cores, wobei in letzter Zeit auch verstärkt Virtuelle Plattformen wie Xen unterstützt werden, um Hosting-Projekte zu ermöglichen. Anwendungen sind für NetBSD über die Paketsammlung pkgsrc verfügbar, zur Zeit mehrere Tausend aus allen möglichen Einsatzgebieten von Datenbank- und Webserver über Desktop-Systeme bis hin zu Firewalling-Anwendungen. Live-CDs sind u.a. in Form von NeWBIE und Xenopix verfügbar.

### OpenBSD – <http://www.OpenBSD.org/>

OpenBSD entstand aus NetBSD und hat sich „maximale Sicherheit“ zum Primärziel erkoren. Dieser Aspekt wird zwar bei den anderen BSDs nicht vernachlässigt, doch bei OpenBSD ist dies Priorität. Sicherheit bedeutet bei OpenBSD vor allem auch Codequalität und den vollkommenen Verzicht auf sämtliche nicht als wichtig erachteten Dienste und Features in der Grundinstallation. Das Team von Theo de Raadt führt rigorose Quelltext-Auditing durch, um Schwachstellen aufzuspüren. OpenBSD ist dominiert vom Bekenntnis zur Kryptographie, IPsec und IPv6 sind in der Grundinstallation enthalten, Kerberos, Hashfunktion, ProPolice, W^X, AFS sind weitere Stichworte.

Weitere Projekte im Zusammenhang sind OpenSSH, OpenSSL, OpenBGPD, OpenNTPD und OpenCVS. OpenBSDs Innovationen wie pf, pfsync, CARP sind mittlerweile von den anderen BSDs übernommen worden. Genauso wichtig ist aber das Festhalten an Standards und dem Grundgedanken von freier Software. Dies beinhaltet die Ablehnung von Softwarepatenten genauso wie den Verzicht auf proprietäre Techniken, Protokolle etc. Es finden sich 3.300 Applikationen verfügbar als Binärpakete oder aus der Ports Collection. OpenBSD legt weniger Wert auf Features und Programmvielfalt als andere BSDs.

### DragonFly BSD – <http://www.DragonFlyBSD.org/>

DragonFly BSD ging aus FreeBSD hervor und hat sich zum Ziel gesetzt SSI (Single System Image clustering) zu verwirklichen. Anstatt sich also um viele einzelne Systeme kümmern zu müssen, wird man diese zukünftig einfach vernetzen können, und DragonFly wird sich wie ein einziges System verhalten.

Die Codequalität hat bei DragonFly BSD Projekt einen sehr hohen Stellenwert. Auch Sicherheit und Stabilität kommen unter DragonFly nicht zu kurz. Das System unterstützt sowohl den Paketfilter pf von OpenBSD als auch ipfw (FreeBSD). Zusätzlich wurden SACK, eine Erweiterung des TCP-Protokolls, die für bessere Performance bei Paketverlusten sorgt, und SCTP, ein SYN-Flood-resistentes Transportprotokoll implementiert. Als Schutz vor Buffer Overflows wurde die GNU Compiler Collection um dem "Stack-Smashing Protector" erweitert. DragonFly BSD unterstützt zur Zeit nur x86basierte Hardware, sowie amd64/em64t im 32bit-Modus. Wie FreeBSD hat auch DragonFly eine sehr gute SMP Unterstützung vorzuweisen. An einer echten amd64-Portierung wird derzeit gearbeitet. Die LiveCD dient gleichzeitig auch als Installationsmedium. Mit pkgsrc stehen analog zu NetBSD sehr viele Anwendungen zur Verfügung.

# \*BSD

## Die \*BSD-Familie

### Websites der Projekte:

<http://www.FreeBSD.org/>

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

<http://www.OpenBSD.org/>

<http://www.DragonFlyBSD.org/>

Andere BSD-Projekte:

MirOS BSD #8: <http://MirBSD.de/>

FreeSBIE 1.1: <http://www.FreeSBIE.org/>

DesktopBSD 1.0: <http://DesktopBSD.net/>

PC-BSD 1.0: <http://www.PCBSD.org/>

Auswahl deutschsprachiger BSD-websites:

<http://www.BSDForen.de/>

<http://www.allBSD.de/>

Aktuelle BSD-Versionen, [Stand 04/2006](#)

FreeBSD 6.0-RELEASE

NetBSD: 3.0-RELEASE

OpenBSD: 3.8-RELEASE

DragonFly BSD: 1.4.4-RELEASE



Eine gern gestellte Frage ist der Vergleich zwischen den Eigenheiten der 3 grossen BSDs (andere BSDs bleiben aus Platzgründen unberücksichtigt, dies ist aber keine Wertung in irgendeiner Form).

## Gemeinsamkeiten

Traditionell unterteilt man die Gruppe der UNIX-Betriebssysteme in zwei Lager: System V („SysV“) und BSD. Die heutigen BSD-Systeme sind direkte Nachfahren von „4.4BSD Lite2“, der letzten offiziellen BSD-Version der Universität Berkeley, Urahn aller BSD-Systeme. Hier ein paar wichtige Punkte:

In BSD-Systemen stecken über 30 Jahre harte Arbeit, es sind stabile, reife Unices.

Im Laufe der Zeit entstanden mehrere BSDs, jedes mit unterschiedlicher Zielsetzung, die aber im Kern immer noch sehr ähnlich sind. Sie ergänzen sich durch Austausch von Code, Dokumentation etc.

Keines der BSDs ist eine „Linux Distribution“ - Linux und FreeBSD verwenden zwar teilweise die gleichen Werkzeuge (gcc z.B.), aber Kernel, Lizenz und Entwicklungsmodell unterscheiden sich gravierend. Ein Distributionschaos gibt es bei BSD nicht.

Allen BSDs gemeinsam ist die Liebe zu UNIX, Standards wie POSIX, ANSI, X/Open etc. und zu stabilem, sicherem, effizientem und leistungsfähigem Code.

## BSD-Lizenz

Alle BSDs verwenden die gleichnamige Lizenz, welche dem Nutzer im Vergleich zur GPL wesentlich größere Freiheiten bietet; der Nutzer kann alles mit dem Code tun, solange er die Quelle nennt, weitere Beschränkungen gibt es nicht. Dies entspricht dem liberalen und freien Geist der BSD-Gemeinschaft.

## Aus einem Guss

Jedes BSD definiert nicht nur den Kernel, sondern auch Bibliotheken, Utilities sowie Build-Tools als „Betriebssystem“, das als in sich geschlossene Einheit gepflegt wird. Upgrades beziehen sich immer auf das gesamte Betriebssystem und nicht nur auf den Kernel, wie bei Linux. Betriebssystem und Kernel lassen sich mit wenigen Kommandos aus den Quellen komplett neu bauen. Es herrscht, durch eine geeignete Filesystem-Hierarchie, eine strikte Trennung zwischen Betriebssystem und Applikationen (Firefox, Thunderbird, OpenOffice usw.). Betriebssystem

und Anwendungsprogramme lassen sich nur getrennt aktualisieren, was in der Praxis die Langlebigkeit der Installationen enorm vergrössert.

## Professionelle Entwicklung

Alle BSDs pflegen den Source in öffentlich einsehbaren CVS-Systemen. Alle Änderungen lassen sich nachvollziehen und ggf. auch zurücknehmen. Nur ausgewählte Personen (committer genannt), welche über die Erfahrung und das Können verfügen, dürfen Änderungen vornehmen. Aber jedermann kann Einblick nehmen, Verbesserungsvorschläge machen und sich so in den Entwicklungsprozess einbringen. Die ständige Überprüfung durch viele Personen sorgt für erhöhte Sicherheit und bessere Codequalität. Diese wird als wesentlich wichtiger erachtet als z.B. die Masse an Features.

## Straffe Organisation

Die BSD-Projekte sind kein „loser Haufen“, sondern bilden eine professionelle Organisation wie in kommerziellen Software-Projekten. Es existieren bei NetBSD und FreeBSD Gremien, genannt Core, welche Planungsaufgaben wahrnehmen. Um das Core-Team herum gibt es eine große Anzahl weiterer Entwickler - committer - die für Dokumentation, Applikationen etc. zuständig sind. Zusätzlich gibt es Sicherheitsteams (sehr wichtig!) und andere Teams mit Spezialaufgaben.

## Software im Überfluss

Alle BSDs können über 99% der im Quellcode verfügbaren Software (z.B. von <http://freshmeat.net>) kompilieren und ausführen, meist gibt es aber auch schon fertige Pakete. Durch ein ABI (Application Binary Interface) lassen sich Linux-Programme auch unter BSD nutzen, selbst wenn es nur Binär-Pakete für Linux bzw. ausschließlich Linux-Versionen (Acrobat-Reader, Oracle etc.) gibt. Dies geschieht ohne sichtbare Geschwindigkeitseinbußen. Die Qualität bzw. Vollständigkeit des Linux ABI ist jedoch nicht unter allen BSDs gleich gut entwickelt – FreeBSD und DragonFly BSD haben hier die meisten Fähigkeiten.

## Stabiles Dateisystem

Das UFS-Dateisystem (bzw. FFS) ist seit Jahrzehnten bewährt und sehr leistungsfähig, durch Softupdates wird die Integrität der Meta-Daten auch bei einem Crash sichergestellt. UFS2 (FreeBSD, NetBSD) bietet zusätzlich noch weitere Möglichkeiten, um den Betrieb schneller aufnehmen zu können. Ein Journaling File System ist in

Entwicklung und steht in FreeBSD bald zur Verfügung. Daneben können Alternativen wie NFS, XFS, ext2, FAT32, NTFS, ReiserFS 3.6 etc. genutzt werden, diese allerdings teilweise eingeschränkt und/oder nur-lesend.

## Paketsysteme

Viele frei verfügbare Software ist nur im Source verfügbar, oder kann nicht ohne juristische Probleme als Package (fertig kompiliertes Paket, das BSD-Gegenstück zum RPM) weiterverbreitet werden. Daher gibt es das Ports-Framework in FreeBSD/OpenBSD. Es besteht aus einer Hierarchie von Makefiles und ggf. Patches die beschreiben, wie sich ein bestimmtes Source-Paket entpacken, kompilieren und installieren lässt. Das Ports-System kann eigenständig benötigte Quellcode- oder Binärdateien aus dem Internet oder von CD laden und berücksichtigt dabei auch Abhängigkeiten. Das Ports-System lässt sich sowohl auf Kommandozeilenebene als auch per graphischer Oberfläche leicht bedienen; es ist einer der wesentlichen Stärken von BSD. NetBSD und neuerdings DragonFly BSD setzen auf pkgsrc als Alternative zu den klassischen Ports; pkgsrc lässt sich auf sehr vielen Plattformen und sogar unterschiedlichen Betriebssystemen einsetzen; pkgsrc ist übrigens auch auf FreeBSD und OpenBSD verfügbar. MirBSD nutzt das auch auf OpenBSD und MacOS X verwendbare MirPorts Framework, ein Derivat des OpenBSD Ports Trees.

## Exzellente Dokumentation

Man kann mit Berechtigung behaupten, daß alle BSDs die am besten dokumentierten Betriebssysteme überhaupt sind. Handbücher, manual pages, Webseiten etc. sind klar strukturiert, verständlich geschrieben und ständig aktualisiert. Die gebotenen Hilfestellungen sind exzellent und für jedermann frei nutzbar.

