

FreeBSD

Ports und Pakete



Im Laufe der Zeit werden sich zahlreiche Dateien im Verzeichnis `distfiles` (die komprimierten Quellcodepakete) ansammeln. Sie verwenden folgenden Befehl, um alle nicht länger gebrauchten `distfiles` zu löschen:

```
# portsclean -D
```

Falls Sie nur alle `distfiles` löschen wollen, die von keinem derzeit installierten Port benötigt werden:

```
# portsclean -DD
```

Ports entfernen

Ports können sie mit `pkg_deinstall` entfernen, welches Teil von `Portupgrade` ist. Sie können alternativ dazu auch `pkg_delete` nutzen. Wenn sie es einfacher haben möchten, können Sie den Port `sysutils/pkg_cutleaves` verwenden, dieser findet automatisch alle Pakete, welche nicht von einem anderen Paket/Port benötigt werden, und schlägt die Löschung jedes einzelnen Paketes Schritt für Schritt vor.

Grafische Paketmanager

Selbstverständlich gibt es auch grafische Oberflächenden für Ports und Pakete. Neben dem `nurses`-basierendem `sysutils/pkgfe` gibt es z.B. `sysutils/portbrowser` und die vorzüglichen `sysutils/desktopbsd-tools`, welche auf KDE aussetzen und Standard sind in `DesktopBSD` und `PC-BSD`. Mit diesen Werkzeugen können auch Anfänger sehr leicht Ports installieren und verwalten.

Weitere Informationen

Falls Sie mehr über Ports wissen wollen empfehlen wir Ihnen einen Blick in das exzellente `FreeBSD` Handbuch:

<http://www.freebsd.org/doc/de/books/handbook/>

Weitere Quellen finden Sie hier:

<http://www.freebsd.org/doc/en/books/porters-handbook/>

<http://www.lpthe.jussieu.fr/~talon/freebsdports.html>

Portupgrade

`portupgrade` wurde entwickelt, um die Aktualisierung von Ports zu vereinfachen. Es befindet sich wie fast alle Utilities für die Ports in der Kategorie "sysutils". Installation:

```
# cd /usr/ports/sysutils/portupgrade && make install clean
```

Durchsuchen Sie regelmäßig (am besten vor jeder Aktualisierung) die Liste der installierten Ports mit `pkgdb -F` und beheben Sie alle gefundenen Probleme.

Wenn Sie `portupgrade -a` eingeben, beginnt die automatische Aktualisierung aller veralteten Ports Ihres Systems. Verwenden Sie den Schalter `-i`, wenn Sie individuell entscheiden wollen, ob ein Port aktualisiert werden soll:

```
# portupgrade -ai
```

Wenn Sie nur eine einzelne Anwendung anstelle aller Anwendungen aktualisieren wollen, verwenden Sie das Kommando `portupgrade pkgname`. Geben Sie den Schalter `-R` an, wenn `portupgrade` zuvor alle Ports aktualisieren soll, die von dem gegebenen Paket abhängen.

Der Schalter `-P` verwendet zur Installation Pakete anstelle von Ports. Mit dieser Option durchsucht `portupgrade` die in der Umgebungsvariablen `PKG_PATH` aufgeführten Verzeichnisse nach Paketen. Sind lokal keine Pakete vorhanden, versucht `portupgrade` die Pakete über das Netz herunterzuladen. Gibt es die Pakete weder lokal noch auf entfernten Rechnern, werden die Ports verwendet. Sie sehen bereits, `portupgrade` ist sehr mächtig, lesen Sie bitte unbedingt die [manual page](#) durch vor Verwendung!

Portaudit

Zum Erkennen von Programm-Sicherheitslücken hilft das Programm `security/portaudit`. Mit:

```
# portaudit -Fa
```

werden alle bekannten Sicherheitslücken aller auf dem Rechner installierten Programme aufgelistet. Dies kann man natürlich auch per `cron` zu beliebigen Intervallen durchführen lassen. Sie können `portaudit` auch vor Installation eines Ports/Paketes ausführen und überprüfen, ob die Applikation verwundbar ist.

Platz schaffen

Die Ports-Sammlung (Portstree) kann sehr viel Platz verschlingen. Falls Sie nicht immer `clean` beim Installieren eines Ports ausgeführt haben, können Sie dies auf einen Rutsch nachträglich ausführen im gesamten Portstree mit `portsclean`, welches Teil von `portupgrade` ist:

```
# portsclean -C
```

Einführung

Ein Betriebssystem wie z.B. `FreeBSD` ist die Software, die die Verwendung eines Computers ermöglicht. Es verwaltet Speicher, Ein- und Ausgabegeräte usw. und steuert die Ausführung von Programmen. Ohne Programme wie z.B. Browser, Textverarbeitung, Mailprogramme usw. (Applikationen) ist ein Betriebssystem ziemlich nutzlos.

Sie werden sich jetzt fragen, wie man dieses Applikationen installiert, updatet, nach verwundbaren Applikationen sucht usw., die Antwort finden Sie in diesem Flyer.

Eine gern gestellte Frage ist, ob `FreeBSD` denn genug Applikationen hätte. Sie werden verwundert sein zu hören, dass `FreeBSD` mehr Applikationen zu bieten hat, als jedes andere freie Betriebssystem inklusive `Debian` (dieses hat zwar mehr Pakete als `FreeBSD`, die Zählweise ist aber eine andere). Wir sind uns absolut sicher, dass sie bei über **16.000** Applikationen für fast jeden Zweck etwas finden werden. Wir vergessen nicht zu erwähnen, dass nahezu alle Applikationen aus `Linux` auch unter `FreeBSD` lauffähig sind, die Ausnahmen sind sehr gering.

Ports und Packages

Um Software zu installieren, stellt `FreeBSD` zwei sich ergänzende Methoden zur Verfügung: Die Ports-Sammlung (zur Installation aus dem Quellcode) sowie Pakete (auch als `Packages` bezeichnet, zur Installation von vorkompilierten binären Softwarepaketen). Sie können beide Methoden benutzen, um Ihre Lieblingsanwendungen von lokalen Medien oder über das Netzwerk zu installieren.



Das FreeBSD-Paket einer Anwendung besteht aus einer einzigen Datei, die Sie sich herunterladen müssen. Das Paket enthält schon übersetzte Kommandos der Anwendung, sowie zusätzliche Konfigurationsdateien oder Dokumentation. Zur Handhabung der Pakete stellt FreeBSD-Kommandos wie `pkg_add`, `pkg_delete`, `pkg_version` oder `pkg_info` zur Verfügung. Mit einem einzigen Kommando, `pkg_add`, wird alles automatisch installiert.

Der FreeBSD-Port einer Anwendung ist eine Sammlung von Dateien, die das Kompilieren der Quelltexte einer Anwendung automatisieren. Die Dateien eines Ports führen für Sie alle Schritte zum Installieren einer Anwendung durch. Mit wenigen Kommandos wird der Quellcode der Anwendung automatisch heruntergeladen, ausgepackt, gepatcht, übersetzt und installiert. Ports können also auch dazu benutzt werden Pakete zu erzeugen.

Pakete und Ports beachten Abhängigkeiten (dependencies) zwischen Anwendungen. Sie wollen z.B. eine Anwendung installieren, die von einer Bibliothek abhängt, und beide sind als Paket oder Port für FreeBSD verfügbar. Wenn Sie `pkg_add` oder das Portsystem benutzen, werden Sie feststellen, dass die Bibliothek zuerst installiert wird, wenn sie nicht schon vorher installiert war.

Warum nun beide Methoden? Beide Methoden haben ihre Stärken und welche Sie einsetzen, hängt letztlich von Ihren Vorlieben ab.

Vorteile von Paketen

Das komprimierte Paket einer Anwendung ist normalerweise kleiner als das komprimierte Archiv der Quelltexte.

Pakete müssen nicht mehr kompiliert werden, ein großer Vorteil bei großen Applikationen wie KDE, GNOME etc.

Wenn Sie Pakete verwenden, brauchen Sie nicht zu verstehen, wie Sie Software unter FreeBSD kompilieren.

Vorteile von Ports

Da die Pakete auf möglichst vielen System laufen sollen, werden Optionen beim Übersetzen zurückhaltend gesetzt. Wenn Sie eine Anwendung über die Ports installieren, können Sie die Angabe der Optionen optimieren. Apache kann zum Beispiel über viele eingebaute Optionen konfiguriert werden. Wenn Sie das Portsystem benutzen, können Sie die Vorgaben für die Optionen überschreiben.

Manchmal gibt es mehrere Pakete einer Anwendung. Für **ghostscript** gibt es ein ghostscript-Paket und ein ghostscript-nox11-Paket, die sich durch die X11 Unterstützung unterscheiden. Diese grobe Unterscheidung ist mit dem Paketsystem möglich, wird aber schnell unhandlich, wenn eine Anwendung mehr als zwei Optionen aufweist.

Die Lizenzbestimmungen mancher Software verbietet ein Verbreiten in binärer Form. Diese Software muss als Quelltext ausgeliefert werden.

Einige Leute trauen binären Distributionen nicht. Wenn Sie den Quelltext besitzen, können Sie sich diesen (zumindest theoretisch) durchlesen und nach möglichen Problemen durchsuchen.

Wenn Sie eigene Anpassungen besitzen, benötigen Sie den Quelltext, um diese anzuwenden.

Freshports: Wie finde ich Applikationen?

Wenn Sie den Namen einer Anwendung nicht kennen, besuchen Sie eine Webseite wie z.B. FreshMeat (<http://www.freshmeat.net/>), um eine passende Anwendung zu finden. Schauen Sie dann unter www.freshports.org nach, wo Sie nach Kategorien (Audio, Games, Editoren usw.) suchen können, sehen welche neuen Applikationen hinzugekommen sind, welche verwundbar sind oder Probleme verursachen etc. Machen Sie sich die Mühe und stöbern Sie ein wenig durch Freshports. Vorsicht: Freshports hat hohes Suchtpotential!

Die Ports-Sammlung installieren

Bevor Sie einen Port installieren können, müssen Sie zuerst die Ports-Sammlung installieren, die aus Makefiles, Patches und Beschreibungen besteht. Die Ports-Sammlung wird für gewöhnlich unter `/usr/ports` installiert. Die Ports-Sammlung können Sie entweder bei der Installation von FreeBSD mit `sysinstall` installieren, per `cvsup` oder `csup`. Wir empfehlen Ihnen jedoch die Verwendung von `portsnap`, welches ab FreeBSD 6.1 im Basissystem vorhanden ist. Die Anwendung ist simpel:

1. Laden Sie einen komprimierten Snapshot der Ports-Sammlung in das Verzeichnis `/var/db/portsnap` herunter.

```
# portsnap fetch
```

2. Wenn Sie `portsnap` das erste Mal verwenden, müssen Sie den Snapshot nach `/usr/ports` extrahieren:

```
# portsnap extract
```

3. Wollen Sie die bereits installierte Ports-Sammlung nur aktualisieren, reicht dieser Befehl:

```
# portsnap fetch update
```

Sie können auch per cronjob in jedem beliebigen Intervall aktualisieren, falls Sie dies möchten. Sie sehen, alles sehr einfach und wie fast immer bei FreeBSD haben Sie mehrere Möglichkeiten ihr Ziel zu erreichen.

Ports installieren

Wenn Sie eine aktuelle Ports-Sammlung haben und **root** geworden sind, können Sie frohen Mutes an die Installation ihrer gewünschten Ports (Applikationen) gehen.

Die Ports gehen davon, dass Ihr System über eine funktionierende Internetverbindung verfügt, falls das nicht der Fall ist lesen Sie bitte im FreeBSD Handbuch nach. Keine Sorge, auch solche Fälle sind einfach zu handhaben. Nehmen wir an, Sie möchten ihre geliebte bash als shell haben. Wechseln Sie in das Verzeichnis des zu installierenden Ports, bauen Sie ihn mit `make`, installieren Sie ihn mit `install` und räumen Sie das Arbeitsverzeichnis `work` auf mit `clean`. Huch, Sie sind erschrocken? Keine Panik!

```
cd /usr/ports/shells/bash/ && make install clean
```

Das ist alles, was Sie tun müssen! Sie wollen statt eines Ports ein Paket verwenden? Immer mit der Ruhe:

```
pkg_add -r bash
```

So einfach geht das! Einige Ports besitzen zusätzliche Funktionen oder Sicherheitsoptionen. Zusätzliche Optionen können beispielsweise für [www/mozilla](http://www.mozilla.com) und [mail/sylpheed-claws](http://www.sylpheed-claws.org) angegeben werden. Wenn ein Port über zusätzliche Optionen verfügt, werden diese beim Bau des Ports auf der Konsole ausgegeben.

Ports aktualisieren

Vor allem Neulinge erliegen ständig der Versuchung in geradezu manischer Weise immer die neuesten Versionen einer Applikation besitzen zu wollen, wir sprechen hier von einer Updateritis oder Portsucht.

Nachdem Sie die Ports-Sammlung auf den neusten Stand gebracht haben (`portsnap fetch update`), lesen Sie bitte **unbedingt!!!** `/usr/ports/UPDATING` bevor Sie einen Port aktualisieren. In dieser Datei werden Probleme und zusätzlich durchzuführende Schritte bei der Aktualisierung einzelner Ports beschrieben. Dazu gehören solche Dinge wie geänderte Dateiformate, verschobene Konfigurationsdateien, aber auch Inkompatibilitäten zu einer Vorgängerversion. **Extrem viele Probleme beim Update sind durch das Nichtlesen von Updating bedingt!** Falls sie große Applikationen wie KDE oder Gnome updaten wollen, ist ein `pkg_delete -r` mit anschließendem `pkg_add` oftmals einfacher. Eine Liste aller installierten Ports mit einer neuen Version lassen Sie sich wie folgt anzeigen:

```
# pkg_version -v
```

Es wird Sie nicht verwundern, dass verschiedene Werkzeuge beim Aktualisieren zur Verfügung stehen. Die bekanntesten sind `portmanager`, `portmaster` und **portupgrader**, wir beschränken uns hier mangels Platz auf letzteres.