

NetBSD auf dem neuesten Stand



WALTER JUSTEN, SENIOR SYSTEM ENGINEER

Ein tagesaktuelles NetBSD-System, immer auf dem neuesten Stand? Kein Problem mit den Quellen aus dem NetBSD-current-Baum. Dieser Beitrag beschreibt, wie man ein Update des Systems auf NetBSD-current durchführt.

NetBSD ist wie jedes andere Unix-Derivat in mehr oder weniger häufigen Zyklen als eine »Stable«-Version verfügbar, aktuell als 1.6.1. Wer jedoch immer die neueste und aktuellste Software auf seinem System haben möchte oder gerne selbst einmal an den aktuellen Quelltexten mitentwickeln will, kann auf NetBSD-current zurückgreifen. Dort befinden sich die letzten und aktuellen Quellen, die täglich von den Entwicklern im CVS-System eingepflegt werden. Jedoch ist auch Vorsicht beim Verwenden von NetBSD-current geboten. Neueste Quellen bedeuten in den meisten Fällen auch neueste Bugs.

CVS und Download

Die aktuellsten Quellen von NetBSD findet man auf dem CVS-Server des Projekts. Um sie zu beziehen, muß als erstes die Installation von CVS auf dem eigenen System vollzogen werden. Nachdem CVS installiert ist, werden die aktuellen Quellen aus dem NetBSD-CVS gezogen und unter `/usr/src` gespeichert. Man sollte prüfen, ob genug Speicherplatz zur Verfügung steht. Ein kompletter Checkout

der Current-Quellen umfaßt derzeit gut ein GByte.

Für die Übertragung der Quelltexte ist aus Sicherheitsgründen die Secure Shell zu empfehlen. Leider funktioniert diese Möglichkeit der Übertragung aber nicht mit allen NetBSD-CVS-Servern. Eine Auswahl der SSH-fähigen Server gibt der Überblick in Tabelle 1. Um Bandbreite zu sparen, sollte der CVS-Server bevorzugt werden, der sich in geographischer Nähe befindet.

Bevor ein Checkout oder Update von CVS gestartet werden kann, müssen noch die benötigten Umgebungsvariablen gesetzt werden. Sie enthalten einmal den ausgewählten CVS-Server und die Information, daß die

Secure Shell zur Übertragung genutzt werden soll. Der Aufruf bei Verwendung der C-Shell sieht so aus:

```
% setenv CVSROOT \
anoncvs@grappa.unix-ag.uni-kl.de:/\
cvsroot
% setenv CVS_RSH ssh
```

Mit der `sh` oder `bash` so:

```
# CVSROOT=anoncvs@grappa.unix-ag.uni-kl\
.de:/cvsroot; export CVSROOT
# CVS_RSH=ssh; export CVS_RSH
```

Dabei sollten die hier mit »\« getrennten Zeilen am Stück eingegeben werden.

Möchte man den kompletten current-

| Server | Adresse |
|-----------------|--|
| USA Mastersite: | <code>anoncvs.netbsd.org:/cvsroot</code> |
| Deutschland: | <code>grappa.unix-ag.uni-kl.de:/cvsroot</code> |
| Frankreich: | <code>anoncvs.fr.netbsd.org:/pub/NetBSD-CVS</code> |
| Österreich: | <code>anoncvs.sunsite.univie.ac.at:/cvsroot</code> |
| Polen: | <code>anoncvs.pl.netbsd.org:/cvs</code> |

Tabelle 1: SSH-fähige NetBSD CVS Server mit CVS-root Directories

Baum aus dem CVS herunterladen, muß man ein Checkout durchführen:

```
# cd /usr/
# cvs -d $CVSRROOT checkout src
# cvs -d $CVSRROOT checkout xsrc
```

Um diesen lokalen Baum anschließend auf den neusten Stand zu halten, ist jeweils ein Abgleich mit dem CVS-Server und damit ein Update erforderlich.

```
# cd /usr/src
# cvs -d $CVSRROOT update -dPA
# cd /usr/xsrc
# cvs -d $CVSRROOT update -dPA
```

Erstellen und installieren

Eine gute und gern genutzte Alternative zu CVS besteht im Download von den FTP-Servern des NetBSD-Projekts. Dort stehen die Quellen im Pfad */pub/NetBSD-current/src* beziehungsweise */pub/NetBSD-current/*

xsrc ebenfalls als einzelne Dateien oder im Pfad */pub/NetBSD-current/tar_files* als gepackte Archive zur Verfügung.

Da die Dateien auf dem FTP-Server in der Regel etwas älter sind als ein Checkout direkt aus dem CVS, empfiehlt sich nach dem FTP-Download ein Update wie bereits beschrieben. Spiegel des NetBSD-FTP-Servers zeigt Tabelle 2.

Bevor die Arbeit am System und den neuen Quellen beginnt, sollte eine Sicherheitskopie der Einstellungen und Konfiguration des laufenden Systems erstellt werden. Diese Daten werden teilweise überschrieben. Auch Anwenderdaten können verlorengehen:

```
# tar cvf /etc.tar /etc
```

Unabhängig davon, für welche Möglichkeit der Beschaffung man sich entschieden hat, sollten die NetBSD-Quelltexte nun unter */usr/src* und die X-Quellen unter */usr/xsrc* auf dem lokalen System zur Verfügung ste-

hen. Außerdem muß vor dem Kompilieren noch das Directory */usr/obj* angelegt werden:

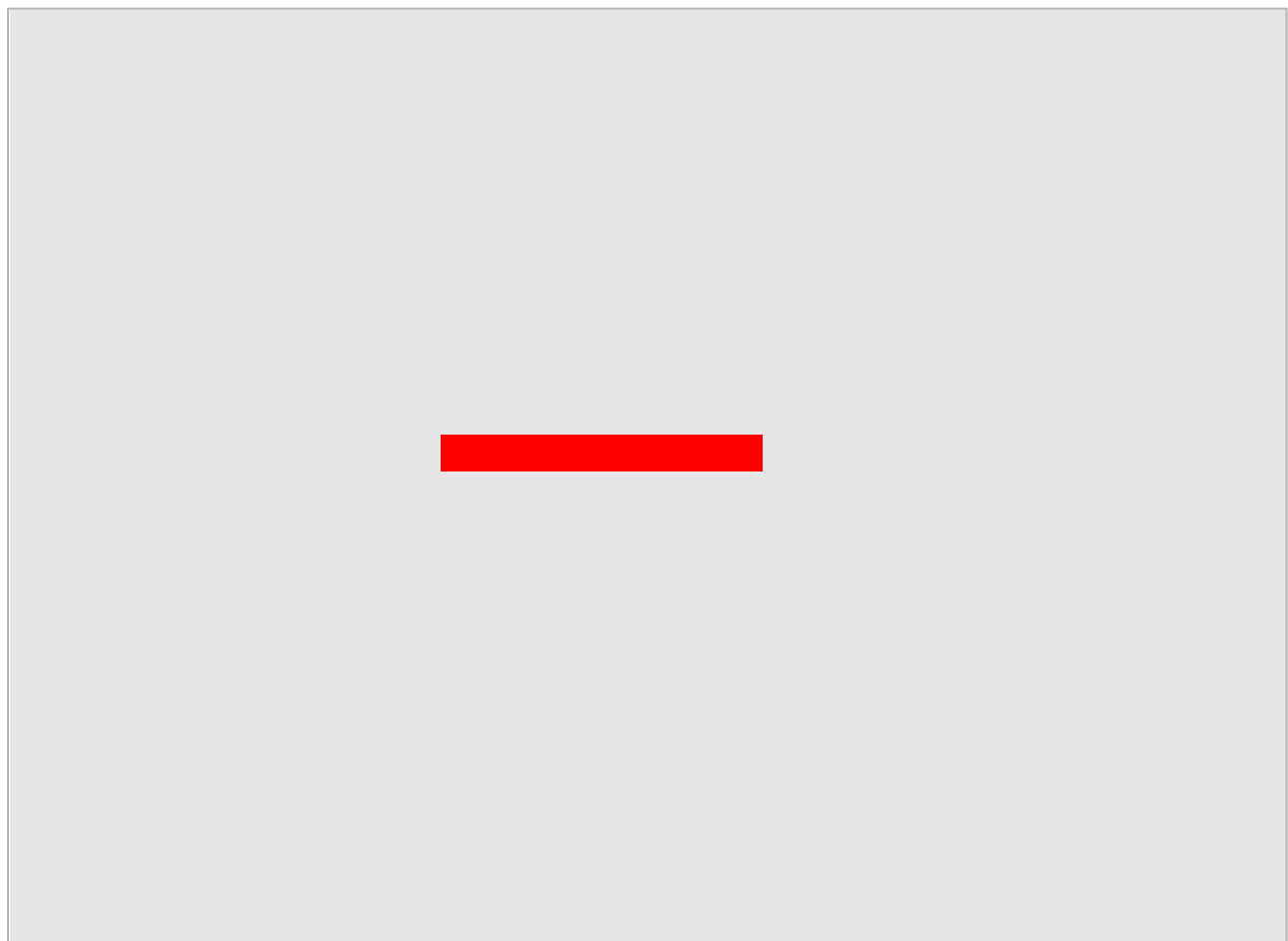
```
# cd /usr
# mkdir obj
```

Jetzt werden noch einige Tools benötigt. Dazu gehören unter anderem die aktuellen Versionen von Programmen wie *make*, *yacc* und *lex*. Sie und auch der Kernel werden im ersten Schritt kompiliert und installiert. Je nach Hardware-Plattform unterscheidet sich der Pfad zum neuen Kernel. Ist dieser Vorgang abgeschlossen und der neue Kernel installiert, wird das System mit diesem neuen Kernel gebootet:

```
# cd /usr/src
# ./build.sh tools kernel=GENERIC

# cp /netbsd /netbsd.old
# cp /usr/src/sys/arch/<PLATTFORM>/\
compile/GENERIC/netbsd /
```

War der Neustart mit dem neuen



| Land | Adresse |
|-------------|---------------------------|
| USA | <i>ftp.netbsd.org</i> |
| Deutschland | <i>ftp.de.netbsd.org</i> |
| Deutschland | <i>ftp2.de.netbsd.org</i> |
| Frankreich | <i>ftp2.fr.netbsd.org</i> |
| Österreich | <i>ftp.at.netbsd.org</i> |

Tabelle 2: ftp-Server für NetBSD

Kernel erfolgreich, sollte schnellstmöglich der Rest des Systems auf den neuesten Stand gebracht werden. Je nachdem, welcher Zeitraum zwischen dem verwendeten aktuellen Kernel und der immer noch nicht aktualisierten Umgebung liegt, kann es zu Problemen kommen. Das betrifft vor allem Programme und Tools, die direkt auf Funktionen des Kerns aufsetzen.

Nun können die Quellen der Umgebung mit dem Skript *build.sh* kompiliert werden. Meist wird man es mit folgenden Optionen aufrufen:

```
# cd /usr
# mkdir destdir
# mkdir releasedir
# cd /usr/src
# ./build.sh -U -D \
  /usr/destdir distribution
```

Die Option *distribution* steht für einen Build der kompletten Distribution, die Option *-D* gibt das Zielverzeichnis an, in das die Distribution nach dem Kompilieren kopiert werden soll. Der Name des Verzeichnisses kann frei gewählt werden, es muß jedoch vor dem Kompilieren bereits vorhanden sein.

Möchte man für weitere Installation auf anderen Systemen die Installations-Sets ebenfalls erzeugen, wird noch die Option *-R* benötigt:

```
# ./build.sh -U -u -D /home/destdir \
  -R /home/releasedir release
```

releasedir definiert das Verzeichnis, in das die Installations-Sets kopiert werden sollen. Dieser Name ist ebenfalls frei wählbar. Auch hier gilt, daß es vor dem Kompilierungsvorgang vor-

handen sein muß.

Weitere Optionen des Skripts finden Sie auch in der Datei */usr/src/BUILDING*.

Im nächsten Schritt muß die Distribution, die hier im Beispiel unter */home/destdir* erstellt wurde, in das Root-Directory installiert werden, um produktiv zu gehen. Diese Arbeit kann nur vom Systemadministrator *root* durchgeführt werden:

```
# ./build.sh -U -D /home/destdir \
  install=
```

Damit nach der Installation des Systems und einem Reboot noch alle Konfigurationen aktuell sind und auch noch alles so funktioniert wie man es erwartet, sollten nun die Konfigurationen abgeglichen werden. Für diese Aufgabe ist das Tool *etcupdate* sehr nützlich. Bei älteren NetBSD-Versionen ist es möglich, daß *etcupdate* noch fehlt. Dann muß es nachinstalliert werden. Die Durchführung des Konfigurationsabgleichs für das Beispiel könnte wie folgt aussehen:

```
# cd /
# tar xvf /etc.tar
# /usr/sbin/etcupdate -s /usr/src -v -a
```

Weitere Optionen und Möglichkei-



ten des Tools sollten in der entsprechenden Manpage nachgelesen werden. Ist die Konfiguration unter */etc* erfolgreich abgeglichen, kann mit einem Neustart das aktualisierte Betriebssystem in Betrieb genommen werden:

```
# reboot
```

Neben dem Kernel und der Systemumgebung stehen uns auch noch die neuesten Quellen der grafischen Oberfläche zur Verfügung. Ebenso wie bei der Umgebung kann hier ein Zielverzeichnis angegeben werden. Außerdem lassen sich Installations-Sets für den weiteren Gebrauch erstellen:

```
# cd /usr
# mkdir xdestdir
# cd /usr/xsrc
# export RELEASEDIR=/usr/releasedir
# export DESTDIR=/usr/xdestdir
# make release
```

In diesem Beispiel werden die Optionen für die beiden benötigten Verzeichnisse als Umgebungsvariable übergeben. Die X-Distribution wird unter *DESTDIR* erstellt, die Installations-Sets sind nach dem Kompilieren unter *RELEASEDIR* zu finden. Weitere Informationen und Optionen zum Übersetzen der X-Quellen können in der Make-Datei unter */usr/xsrc* nachgelesen werden. Ist der Vorgang erfolgreich abgeschlossen, kann die erstellte Distribution in den Standardpfad */usr/X11R6* verschoben werden.

```
# cd /usr
# mv X11R6 X11R6.old
# mv /usr/xdestdir/usr/X11R6 .
```

Alternativ besteht die Möglichkeit, die X-Quellen zu kompilieren und direkt ohne den Umweg über ein *DESTDIR* und ohne das Erstellen von Installationssets direkt nach */usr/X11R6* zu installieren:

```
# cd /usr/xsrc
# make build
```