

Kostenloses eBook

LERNEN FreeBSD

Free unaffiliated eBook created from
Stack Overflow contributors.

#freebsd

Inhaltsverzeichnis

Über ... 1

Kapitel 1: Erste Schritte mit FreeBSD ... 2

Bemerkungen ... 2

Versionen ... 2

Examples ... 4

Installation oder Setup ... 4

Kapitel 2: Aus Quelle erstellen ... 5

Einführung ... 5

Bemerkungen ... 5

Überblick über den gesamten Prozess ... 5

Holen Sie sich die Anzahl der Prozessoren ... 5

Examples ... 5

Laden Sie den neuesten Quellcode herunter ... 5

SVN ... 5

Holen Sie sich aktuell ... 6

Veröffentlichungen erhalten ... 6

Tarball (http & ftp) ... 6

http ... 6

ftp ... 6

Git ... 6

GitHub ... 6

Konfigurieren Sie den Kernel ... 6

Baue die Welt und den Kernel ... 7

Baue die Welt ... 7

Geschätzte Zeit ... 7

Bauen Sie den Kernel auf ... 7

Geschätzte Zeit ... 7

Konfigurieren Sie das Root-Dateisystem Ihrer neuen FreeBSD ... 7

Installieren Sie die Welt und den Kernel ... 8

Installiere die Welt ... 8
Installieren Sie den Kernel ... 8
Kapitel 3: FreeBSD-Jails ... 9
Examples ... 9
Gefängnis einsetzen ... 9
Einfache Jail-Bereitstellung aus Binärdateien ... 9
Einfache Jail-Bereitstellung von der Quelle ... 9
Einfache Thin Jail-Bereitstellung ... 10
Initialisierung unserer Umgebung ... 10
Quellen herunterladen ... 10
Initialisierung unseres dünnen Gefängnisses ... 10
Vernetzung und Gefängnisse ... 11
Netzwerkunterstützung entfernen ... 11
Nur IPv4-Netzwerke zulassen ... 11
Erlaubt nur IPv6-Netzwerke ... 11
Dedizierter Netzwerkstack (VNET) ... 11
Kapitel 4: Paket- und Portverwaltung ... 13
Bemerkungen ... 13
Examples ... 13
Ports-Baum abrufen ... 13
Portsnap ... 13
Aktualisierung des Ports-Baums mit Portsnap ... 13
cron-Job für tägliche Updates einplanen ... 13
SVN ... 13
Kopf ... 13
quaterly ... 13
Tarball (http oder ftp) ... 14
Git ... 14
Software suchen ... 14
Schlagwortsuche ... 14

Namenssuche 14
Fresports verwenden 14
Software erstellen und installieren 14
Einfacher Aufbau und Installation mit manueller Konfiguration 14
Einfache Erstellung und Installation mit automatischer Konfiguration 15
Softwarequellen konfigurieren 15
www/apache24 15
Verpackung 15
Manuelle Verpackung 15
poudriere 15
Poudriere installieren 15
Poudriere konfigurieren 16
Poudriere-Gefängnis einsetzen 16
Aktualisierung des Gefängnisses 16
Bereitstellen von poudriere ports tree 16
Poudriere-Ports-Baum wird aktualisiert 16
Massenaufbau 16
Kapitel 5: Richten Sie die FreeBSD-Entwicklungsumgebung ein 17
Examples 17
ctags 17
Erstellen exctags(1) mithilfe von Ports exctags(1) 17
Laden Sie eine vorgefertigte Binärdatei von Exuberant Ctags herunter und installieren Sie 17
Erstellen Sie die Tag-Datei 17
Credits 18

Über

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: freebsd

It is an unofficial and free FreeBSD ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official FreeBSD.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Kapitel 1: Erste Schritte mit FreeBSD

Bemerkungen

In diesem Abschnitt erhalten Sie einen Überblick darüber, was Freebsd ist und warum ein Entwickler es verwenden möchte.

Es sollte auch alle großen Themen in freebsd erwähnen und auf die verwandten Themen verweisen. Da die Dokumentation für freebsd neu ist, müssen Sie möglicherweise erste Versionen dieser verwandten Themen erstellen.

Versionen

Einige Versionen wurden ausgelassen, da das genaue Veröffentlichungsdatum nicht bekannt ist. In der Quelle dieses Beitrags finden Sie die Liste der ausgelassenen Versionen.

Ausführung	Veröffentlichungsdatum
1,0	1993-11-01
1.1	1994-05-01
2,0	1994-11-22
2.0.5	1995-06-10
2.1	1995-11-19
2.2.5	1997-10-22
2.2.6	1998-03-25
2.2.7	1998-07-22
2.2.8	1998-11-29
3.1	1999-02-15
3.2	1999-05-17
3.3	1999-09-17
3.4	1999-12-20
3,5	2000-06-24
4,0	2000-03-14

Ausführung	Veröffentlichungsdatum
4.1	2000-07-27
4.1.1	2000-09-27
4.2	2000-11-21
4.3	2001-04-20
4.4	2001-09-20
4,5	2002-01-29
4.6	2002-06-15
4.6.2	2002-08-15
4.7	2002-10-10
4,8	2003-04-03
4.9	2003-10-28
4,10	2004-05-27
4.11	2005-01-25
5,0	2003-01-14
5.1	2003-06-09
5.2	2004-01-09
5.2.1	2004-02-25
5.3	2004-11-06
5.4	2005-05-09
5.5	2006-05-25
6,0	2005-11-04
6.1	2006-05-08
6.2	2007-01-15
6.3	2008-01-18
6.4	2008-11-28

Ausführung	Veröffentlichungsdatum
7,0	2008-02-27
7.1	2009-01-04
7.2	2009-05-04
7.3	2010-03-23
7.4	2011-02-24
8,0	2009-11-25
8.1	2010-07-23
8.2	2011-02-24
8.3	2012-04-18
8.4	2013-06-07
9,0	2012-01-12
9.1	2012-12-30
9.2	2013-09-30
9.3	2014-07-16
10,0	2014-01-20
10.1	2014-11-14
10.2	2015-08-13
10.3	2016-03-28

Examples

Installation oder Setup

FreeBSD ist bekannt für sein gut geschriebenes Handbuch (Link) . Der Installationsvorgang wird ausführlich in **Kapitel 2. FreeBSD** installieren beschrieben.

Erste Schritte mit FreeBSD online lesen: https://riptutorial.com/de/freebsd/topic/5708/erste-schritte-mit-freebsd

Kapitel 2: Aus Quelle erstellen

Einführung

Die folgenden Beispiele sind nicht unbedingt in der richtigen Reihenfolge. Weitere Informationen zum gesamten Prozess finden Sie im Abschnitt "Anmerkungen".

Bemerkungen

Überblick über den gesamten Prozess

1. Laden Sie den neuesten Quellcode herunter.
2. Konfigurieren Sie den Kernel.
3. Baue die Welt und den Kernel.
4. Konfigurieren Sie das Root-Dateisystem Ihrer neuen FreeBSD.
5. Installieren Sie die Welt und den Kernel.

Holen Sie sich die Anzahl der Prozessoren

Um den Aufbau und die Installation des neuen Systems zu beschleunigen, können Sie einfach mehr Prozessoren verwenden, um die Rechenleistung zu erhöhen.

Um herauszufinden, wie viele Prozessoren Sie benötigen, um den Prozess zu beschleunigen:

```
sysctl hw.ncpu
```

Zum Beispiel:

```
hw.ncpu: 1
```

Setzen `$NUMBER_OF_PROCESSORS` Umgebungsvariable `$NUMBER_OF_PROCESSORS` :

```
export $NUMBER_OF_PROCESSORS=$(sysctl hw.ncpu | tr -d 'a-z.: ')
```

Examples

Laden Sie den neuesten Quellcode herunter

SVN

Das FreeBSD-Projekt verwendet SVN als Standard-SCM. Die Quelle könnte mit der `svnlite` Software heruntergeladen werden.

Holen Sie sich aktuell

```
cd /usr/src
svnlite checkout https://svn.freebsd.org/base/head .
```

Veröffentlichungen erhalten

```
cd /usr/src
svnlite checkout https://web.freebsd.org/base/release/11.0.0 .
```

Tarball (http & ftp)

Sie können die Quelle auch aus dem eingefrorenen Archiv mit dem `fetch`

http

```
cd /tmp
fetch http://ftp.freebsd.org/pub/FreeBSD/releases/amd64/11.0-RELEASE/src.txz
cd /usr/src
tar xJvf /tmp/src.txz
```

ftp

```
cd /tmp
fetch ftp://ftp.freebsd.org/pub/FreeBSD/releases/amd64/11.0-RELEASE/src.txz
cd /usr/src
tar xJvf /tmp/src.txz
```

Git

GitHub

```
git clone https://github.com/freebsd/freebsd freebsdsrc
```

Konfigurieren Sie den Kernel

1. Wechseln Sie in das Verzeichnis mit dem Quellcode:

```
cd freebsdsrc
```

2. Wechseln Sie in das Verzeichnis mit dem Konfigurationscode des Kernels:

```
# If your system is 32-bit.
cd sys/i386/conf/
# If your system is 64-bit.
cd sys/amd64/conf/
```

3. Besorgen Sie sich eine Kopie des **GENERIC-** Kernels (nennen wir ihn *MODEDKERNEL*). Es wird die Basis Ihrer Anpassungen sein.

```
cp GENERIC MODEDKERNEL
```

4. `MODEDKERNEL` Sie die `MODEDKERNEL` Datei nach Belieben.

Baue die Welt und den Kernel

Baue die Welt

Gehen Sie zum `freebsdsrc/` (dem Stammverzeichnis des FreeBSD- `freebsdsrc/` den Sie bereits heruntergeladen haben) und erstellen Sie die Welt:

```
sudo make -j${NUMBER_OF_PROCESSORS} buildworld KERNCONF=MODEDKERNEL -DNO_CLEAN
```

Geschätzte Zeit

- *Geschätzte Zeit für den Hasee Q540S, der auf einem Prozessor läuft: 8 Stunden.*
- *Geschätzte Zeit für Dell L702X auf 8 Prozessoren: 98 Minuten.*

Bauen Sie den Kernel auf

So erstellen Sie den Kernel-Lauf:

```
sudo make -j${NUMBER_OF_PROCCESORS} buildkernel KERNCONF=UFFIE -DNO_CLEAN
```

Geschätzte Zeit

- *Geschätzte Zeit für den Hasee Q540S, der auf einem Prozessor läuft: 2 Stunden.*
- *Geschätzte Zeit für Dell L702X auf 8 Prozessoren: 19 Minuten.*

Konfigurieren Sie das Root-Dateisystem Ihrer neuen FreeBSD

Lassen Sie uns das `/usr/home/beastie/MODEDKERNEL` für das Root-Dateisystem Ihrer neuen FreeBSD konfigurieren (beispielsweise `/usr/home/beastie/MODEDKERNEL`).

1. Fügen Sie die folgenden Zeilen zu `/etc/src.conf` , um sie einzurichten:

```
.if ${KERNCONF} == "MODEDKERNEL"
    DESTDIR?=/usr/home/beastie/MODEDKERNEL
    MODULES_OVERRIDE=md ufs
.endif
```

   Denken Sie daran, Leerzeichen statt Tabulatoren zu verwenden, wenn Sie den Code einrücken möchten.

2. Erstellen Sie jetzt das Root-Dateisystem:

   - Verteilungsverzeichnisse erstellen:

     ```
     sudo make distrib-dirs KERNCONF=MODEDKERNEL
     ```

     Geschätzte Zeit für den Hasee Q540S: einige Sekunden.

   - Machen Sie die Verteilung:

     ```
     sudo make distribution KERNCONF=UFFIE
     ```

     Geschätzte Zeit für Hasee Q540S: 3 Minuten.

Installieren Sie die Welt und den Kernel

Installiere die Welt

```
sudo make installworld KERNCONF=MODEDKERNEL
```

Geschätzte Zeit für Hasee Q540S: 5 Minuten.

Installieren Sie den Kernel

```
sudo make installkernel KERNCONF=MODEDKERNEL
```

Geschätzte Zeit für den Hasee Q540S: einige Sekunden.

Aus Quelle erstellen online lesen: https://riptutorial.com/de/freebsd/topic/7062/aus-quelle-erstellen

Kapitel 3: FreeBSD-Jails

Examples

Gefängnis einsetzen

Ein Gefängnis ist einfach eine `chroot` mit starker Isolation. Wenn Sie also ein Gefängnis erstellen möchten, müssen Sie einfach ein alternatives Stammverzeichnis erstellen und ein neues Gefängnis darin starten.

Einfache Jail-Bereitstellung aus Binärdateien

```
# create our alternative root path
JAILROOT="/path/to/my/jail"
mkdir -p "${JAILROOT}"
cd "${JAILROOT}"

# get distribution from freebsd repository
fetch http://ftp.freebsd.org/pub/FreeBSD/releases/amd64/11.0-RELEASE/base.txz

# extract it in our alternative root
tar xJvf base.txz

# now we can launch our jail
jail -c name=simplejail path=${JAILROOT}

# to check if jail is up and running we use jls
jls

# now we can enter in our new jail
jexec simplejail sh
```

Einfache Jail-Bereitstellung von der Quelle

```
# create our alternative root path
JAILROOT="/path/to/my/jail"
mkdir -p "${JAILROOT}"

# we need to build binaries from source...
cd /usr/src
make buildworld

# ... and install it in our alternative path
make installworld DESTDIR=${JAILROOT}

# now we can launch our jail
jail -c name=simplejail path=${JAILROOT}

# to check if jail is up and running we use jls
jls
```

```
# now we can enter in our new jail
jexec simplejail sh
```

Einfache Thin Jail-Bereitstellung

Ein dünnes Gefängnis ist einfach ein Gefängnis mit einem gemeinsam genutzten, schreibgeschützten alternativen Root, der mit nullfs geladen wird .

Initialisierung unserer Umgebung

```
# making our shared alternative root
SHARED_ROOT=/path/to/your/shared/root
mkdir -p "${SHARED_ROOT}"

# making our jail root
JAIL_ROOT=/path/to/your/jail/root
mkdir -p "${JAIL_ROOT}"
```

Quellen herunterladen

```
# to initialize our shared root, we can use
# all method described above. Here, we will use
# simple binary initialization from official
# repository
cd "${SHARED_ROOT}"

# get distribution from freebsd repository
fetch http://ftp.freebsd.org/pub/FreeBSD/releases/amd64/11.0-RELEASE/base.txz

# extract it in our alternative root
tar xJvf base.txz
```

Initialisierung unseres dünnen Gefängnisses

```
# now we need to initialize our dedicated
# jail root
cd "${JAIL_ROOT}"
mkdir base

# we make symbolic link pointing to
# files stored in read-only directory
for link in bin boot lib libexec rescue sbin
do
  ln -s ${link} /base/${link}
done

# we do same thing with directory in /usr
for link in bin include lib lib32 libdata libexec sbin share
do
  ln -s usr/${link} /base/usr/${link}
```

```
done

# now we are ready to start our jail!
jail -c name=thinjail path="${JAIL_ROOT}" \
        mount="${SHARED_ROOT} ${JAIL_ROOT} nullfs ro 0 0"

# check if our thin jail is ok...
jls

# we can now grab in it!
jexec thinjail sh
```

Vernetzung und Gefängnisse

FreeBSD-Jails können eine feinkörnige Netzwerkkonfiguration haben. Standardmäßig verwenden alle Jails dieselbe Netzwerkkonfiguration wie der Host.

Netzwerkunterstützung entfernen

```
jail -c name="nonetwork" path="/path/to/your/jail" ip4=disable ip6=disable
```

Nur IPv4-Netzwerke zulassen

```
jail -c name="onlyipv4" path="/path/to/your/jail" ip4=inherit ip6=disable
```

Erlaubt nur IPv6-Netzwerke

```
jail -c name="onlyipv6" path="/path/to/your/jail" ip4=disable ip6=inherit
```

Dedizierter Netzwerkstack (VNET)

`VNET` ist eine neuere Funktion, die es `VNET` erlaubt, einen eigenen Netzwerkstack zu haben. Bei dieser Konfiguration müssen Sie dem Host eine Routing-Funktion hinzufügen. `VIMAGE` Option ist im Host-Kernel erforderlich.

```
# starting our own jail with vnet
jail -c name="vnetjail" path="/path/to/your/jail" vnet=new

# we need a bridge...
ifconfig bridge0 create

# a pair of ethernet interface...
ifconfig epair0 create

# and interconnecting epair, jail and bridge
```

```
ifconfig epair0b vnet vnetjail
ifconfig bridge0 add epair0a
ifconfig bridge0 add ${your_external_interface}
```

FreeBSD-Jails online lesen: https://riptutorial.com/de/freebsd/topic/7070/freebsd-jails

Kapitel 4: Paket- und Portverwaltung

Bemerkungen

Tipps:

- `/usr/ports/UPDATING` Datei `/usr/ports/UPDATING` vor dem Upgrade zu überprüfen. In den von Ihnen verwendeten Programmen oder in ihrer Konfiguration kann es zu erheblichen Änderungen kommen, die Ihre aktuelle Konfiguration beeinträchtigen können.

Examples

Ports-Baum abrufen

Portsnap

```
portsnap fetch
portsnap extract
```

Aktualisierung des Ports-Baums mit Portsnap

```
portsnap update
```

cron-Job für tägliche Updates einplanen

```
0 3 * * * root /usr/sbin/portsnap cron
```

SVN

Kopf

```
cd /usr/ports
svnlite checkout https://svnweb.freebsd.org/ports/head .
```

quaterly

Das FreeBSD Ports-Team friert alle 3 Monate den Baum ein. Um diesen Portierungsbaum zu erhalten, können Sie Portzweige verwenden:

```
cd /usr/ports
svnlite checkout https://svnweb.freebsd.org/ports/branches/2016Q4 .
```

Tarball (http oder ftp)

```
cd /usr/ports
fetch http://ftp.freebsd.org/pub/FreeBSD/releases/amd64/11.0-RELEASE/ports.txz
tar xJvf ports.txz
```

Git

```
git clone https://github.com/freebsd/freebsd-ports
```

Software suchen

Schlagwortsuche

```
cd /usr/ports
make search key=apache
```

Namenssuche

```
cd /usr/ports
make search name=apache24
```

Fresports verwenden

Auf der offiziellen FreeBSD-Port-Website (http://freshports.org/) finden Sie eine nette Möglichkeit, Ports und alle dazugehörigen Informationen zu finden.

Software erstellen und installieren

Wenn Sie Ihre Software im Ports-Tree gefunden haben, können Sie sie jetzt erstellen.

Einfacher Aufbau und Installation mit manueller Konfiguration

```
cd /usr/ports/www/apache24
make
make install
```

Einfache Erstellung und Installation mit automatischer Konfiguration

```
cd /usr/ports/www/apache24
make BATCH=yes
make install
```

Softwarequellen konfigurieren

Wenn Sie eine benutzerdefinierte Konfiguration von Ports wünschen, können Sie sie vor dem `make config` konfigurieren. Alle Portkonfigurationen werden in `/var/db/ports/${CATEGORY_NAME}/options` als Makefile gespeichert.

www/apache24

```
cd /usr/ports/www/apache24
make config
make
make install
```

Diese Konfiguration wird in `/var/db/ports/www_apache24/options` .

Verpackung

Manuelle Verpackung

Sie können Ihr eigenes Paket basierend auf Ports erstellen.

```
cd /usr/ports/www/apache24
make package BATCH=yes
```

Dieser Befehl speichert Ihr Paket in `/usr/ports/packages/All` .

poudriere

`poudriere` ist derzeit der offizielle `poudriere` für FreeBSD.

Poudriere installieren

```
pkg install poudriere
```

```
# or
cd /usr/ports/ports-mgmt/poudriere
make
make install
```

Poudriere konfigurieren

`poudriere` Konfiguration wird in `/usr/local/etc/poudriere.conf` und `/usr/local/etc/poudriere.d` gespeichert

Poudriere-Gefängnis einsetzen

```
poudriere jail -c -j myjail
```

Aktualisierung des Gefängnisses

```
poudriere jail -u -j myjail
```

Bereitstellen von poudriere ports tree

```
poudriere ports -c -p myports
```

Poudriere-Ports-Baum wird aktualisiert

```
poudriere ports -u -p myports
```

Massenaufbau

```
poudriere bulk -j myjail -p myports www/apache24
```

Paket- und Portverwaltung online lesen: https://riptutorial.com/de/freebsd/topic/7069/paket--und-portverwaltung

Kapitel 5: Richten Sie die FreeBSD-Entwicklungsumgebung ein

Examples

ctags

ctags ist ein nützliches Dienstprogramm, mit dem Sie den Quellcode effizienter lesen und bewegen können. Die eingebauten **ctags (1) sind** jedoch nicht das von Ihnen erwartete Dienstprogramm Exuberant Ctags.

Sie können exuberant Ctags (**exctags (1)**) entweder über ports oder `pkg` installieren:

Erstellen `exctags(1)` mithilfe von Ports `exctags(1)`

```
cd /usr/ports/devel/ctags/ && make install clean
```

Laden Sie eine vorgefertigte Binärdatei von Exuberant Ctags herunter und installieren Sie sie:

```
pkg install ctags
```

Erstellen Sie die Tag-Datei

```
exctags -R
```

Richten Sie die FreeBSD-Entwicklungsumgebung ein online lesen:
https://riptutorial.com/de/freebsd/topic/6136/richten-sie-die-freebsd-entwicklungsumgebung-ein

Credits

S. No	Kapitel	Contributors
1	Erste Schritte mit FreeBSD	Community, Mateusz Piotrowski
2	Aus Quelle erstellen	M. Kerjouan, Mateusz Piotrowski
3	FreeBSD-Jails	M. Kerjouan
4	Paket- und Portverwaltung	M. Kerjouan, Mateusz Piotrowski, Michael Zhilin
5	Richten Sie die FreeBSD-Entwicklungsumgebung ein	Mateusz Piotrowski