

# chruetertee.ch Thermoskanne

Beat Gätzi

Version 1.0

Vor einem Jahr wurde *chruetertee.ch* ins Leben gerufen. In diesem Dokument finden sich nun die gesammelten Artikel des ersten Jahres *chruetertee.ch* wieder. Teils ein bisschen überarbeitet, aktualisiert und angepasst. Dies ist sozusagen: **Der kleine Chruetertee für unterwegs!**

Korrekturen, Fragen oder Ergänzungen können gerne an [beat@chruetertee.ch](mailto:beat@chruetertee.ch) gesendet werden.

Alle Artikel stehen unter der Creative Commons 2.5 Lizenz, welche in Anhang B genauer beschrieben ist.



Abbildung 1: BSD Daemon Copyright 1988 by Marshall Kirk McKusick. All Rights Reserved.  
<http://www.mckusick.com/beastie/mainpage/copyright.html>

## Inhaltsverzeichnis

<b>1. FreeBSD</b>	<b>1</b>
1.1. SSL Zertifikat für Apache auf einem FreeBSD selber erstellen und signieren . . . . .	1
1.2. iso Image mit FreeBSD brennen . . . . .	1
1.3. Dateiflags in einer FreeBSD Jail nutzen . . . . .	1
1.4. Installierte Ports mit pkg_cutleaves ausmisten . . . . .	2
1.5. CPU throttling mit FreeBSD 6.0 . . . . .	2
1.6. Herausfinden zu welchem Port eine Datei gehört . . . . .	3
1.7. WEP-verschlüsseltes WLAN mit DHCP nutzen . . . . .	3
1.8. Platz schaffen mit portsclean . . . . .	3
1.9. Ports aktualisieren mit dem gnome_upgrade Skript . . . . .	4
1.10. Überblick über alle installierten Ports . . . . .	5
1.11. Neuer Port: jailaudit . . . . .	5
1.12. FreeBSD Ports richtig deinstallieren . . . . .	5
1.13. Beastie Menu unter FreeBSD 6.0 . . . . .	5
1.14. Systemstatistiken auf der Konsole mit sysstat . . . . .	6
1.15. Flashplugin für Firefox auf FreeBSD . . . . .	6
1.16. Manpages . . . . .	7
1.17. su: not running setuid . . . . .	7
1.18. Port auf einen älteren Stand zurücksetzen mit portdowngrade . . . . .	8
1.19. Live-CD's mit QEMU emulieren . . . . .	9
1.20. Portbaum aktualisieren mit Portsnap . . . . .	10
1.21. Flux CMS Port für FreeBSD . . . . .	11
1.22. Dateien mit allen Dateiattributen kopieren . . . . .	11
1.23. Lokale Mails über einen Mail-Relay versenden . . . . .	12
1.24. Installierte Ports auf Sicherheitsprobleme prüfen . . . . .	12
1.25. Portbaum einfach durchsuchen mit portsearch . . . . .	13
1.26. iPod mit FreeBSD benutzen . . . . .	14
1.27. Einen schnellen CVSupServer finden mit fastest_cvsup . . . . .	15
1.28. Neue Make-Targets säubern das System nach einem Update . . . . .	16
1.29. Audio CD's von der Konsole aus bedienen . . . . .	17
1.30. Konsolenauflösung auf FreeBSD ändern . . . . .	18
1.31. Hostnamen einer Jail ändern . . . . .	18
1.32. Quellcode mit csup anstatt CVSup aktualisieren . . . . .	19
1.33. make targets für den FreeBSD Portbaum . . . . .	19
1.34. Portspezifische Makevariablen setzen . . . . .	20
1.35. Mehrere WEP-verschlüsselte WLAN's mit DHCP nutzen . . . . .	20
1.36. Portabhängigkeiten graphisch darstellen . . . . .	21
1.37. Flashplugin 7 für Firefox auf FreeBSD 6.1 . . . . .	22
1.38. Mit Xinerama zwei Bildschirme an einem Rechner verwenden . . . . .	22
1.39. Eingeloggte Benutzer beobachten . . . . .	23
1.40. Verzeichnis eines Rechners mittels SSH mounten . . . . .	24
1.41. Benutzern temporär das Anmelden verbieten . . . . .	25
1.42. FreeBSD Service Jails mit unionfs erstellen . . . . .	25
<b>2. OpenBSD</b>	<b>28</b>
2.1. OpenBSD von 3.7 auf 3.8 aktualisieren . . . . .	28
2.2. OpenOffice.org 2.0 auf OpenBSD . . . . .	29
2.3. OpenBSD Dateiset entfernen . . . . .	29
2.4. Aktualität des OpenBSD-Portbaumes überprüfen . . . . .	29
2.5. OpenBSD Packages aktualisieren . . . . .	30

## Inhaltsverzeichnis

<b>3. BSD Allgemein</b>	<b>30</b>
3.1. Dateien unter Free- und OpenBSD mit Flags schützen . . . . .	30
3.2. SMB Freigaben einfach mounten mit Sharify-Light . . . . .	31
3.3. bsdtalk . . . . .	32
3.4. Zeit synchronisieren mit ntpd . . . . .	32
3.5. Kalender für die Kommandozeile . . . . .	32
3.6. Befehle und Manpages finden mit apropos . . . . .	33
3.7. Terminal sperren . . . . .	33
3.8. Meldungen an eingeloggte Benutzer senden . . . . .	34
3.9. Auf der Kommandozeile ans Beenden der Arbeit erinnert werden . . . . .	35
3.10. Terminalausgabe aufzeichnen . . . . .	35
3.11. Komprimierte Textdateien lesen . . . . .	36
3.12. Herausfinden welche Benutzer angemeldet sind . . . . .	36
3.13. Grösse eines Verzeichnisinhaltes bestimmen . . . . .	37
<b>4. Programme</b>	<b>37</b>
4.1. Schöner surfen mit Firefox . . . . .	37
4.2. mySQL Datenbank sauber herunterfahren und auf Fehler prüfen . . . . .	38
4.3. Jobkontrolle in der Shell mit bg, fg und jobs . . . . .	38
4.4. E-Mails mit Attachment von der Konsole aus senden . . . . .	38
4.5. DOS Zeileunumbrüche CR/LF in Unix LF umwandeln . . . . .	39
4.6. Konsolen verwalten mit GNU Screen . . . . .	39
4.7. WebDAV Client für die Kommandozeile . . . . .	40
4.8. Leere Verzeichnisse und alte Symlinks löschen . . . . .	41
4.9. Rechnen auf der Kommandozeile . . . . .	41
4.10. Verschiedene Komprimierungsmöglichkeiten . . . . .	42
4.11. MP3 und Ogg-Dateien aus Musik CD's erstellen . . . . .	43
4.12. Anzeigen welcher Host wieviel Bandbreite benutzt . . . . .	44
<b>5. vi(1)</b>	<b>44</b>
5.1. vi Kurzreferenz . . . . .	44
5.2. vi Makros benutzen . . . . .	46
5.3. Abkürzungen im vi benutzen . . . . .	47
5.4. vi Fenster teilen . . . . .	48
<b>6. tcsh(1)</b>	<b>48</b>
6.1. Umlaute in der (t)csh verwenden . . . . .	48
6.2. Piepston der tcsh deaktivieren . . . . .	48
6.3. DEL-Taste in der tcsh benutzen . . . . .	48
<b>7. X.Org</b>	<b>48</b>
7.1. Auflösung unter X.Org während des Betriebes ändern . . . . .	48
7.2. X.Org: could not open default font "fixed" . . . . .	49
7.3. Umlaute nach X.Org Update von 6.8.2 auf 6.9.0 . . . . .	49
7.4. Hosts erlauben den X Server zu benutzen . . . . .	49
<b>A. Danksagung</b>	<b>51</b>
<b>B. Lizenz</b>	<b>51</b>
<b>Stichwortverzeichnis</b>	<b>52</b>

## 1. FreeBSD

### 1.1. SSL Zertifikat für Apache auf einem FreeBSD selber erstellen und signieren

[ Befehl der Woche ] [ FreeBSD ] by Beat @ 01.10.2005 12:49 CET

Mein erster "Befehl der Woche"Eintrag erstellt ein SSL Zertifikat für den Apache2 Webserver und signiert dieses auch selber:

```
# mkdir /usr/local/etc/apache2/ssl.key
# mkdir /usr/local/etc/apache2/ssl.csr
# mkdir /usr/local/etc/apache2/ssl.crt
# openssl genrsa -out /usr/local/etc/apache2/ssl.key/server.key 1024
# openssl req -new -key /usr/local/etc/apache2/ssl.key/server.key \
-out /usr/local/etc/apache2/ssl.csr/server.csr
# openssl x509 -req -days 3650 -in /usr/local/etc/apache2/ssl.csr/server.csr \
-signkey /usr/local/etc/apache2/ssl.key/server.key \
-out /usr/local/etc/apache2/ssl.crt/server.crt
```

Wichtig, bei CommonName (CN) den Servername angeben!

Jetzt werden in der /etc/rc.conf noch die richtigen Einträge hinzugefügt:

```
apache2_enable="YES"
apache2ssl_enable="YES"
```

Und nach einem Neustart des Apaches mit /usr/local/etc/rc.d/apache2.sh restart startet der Apache mit dem selbst erstellten Zertifikat.

### 1.2. iso Image mit FreeBSD brennen

[ Befehl der Woche ] [ FreeBSD ] by Beat @ 07.10.2005 06:08 CET

Um ein iso Image mit FreeBSD zu brennen, braucht man einen CD-Brenner und eine Konsole. Einfach folgende Zeile in die Konsole eingeben:

```
# burncd -f /dev/<Brenner> data <Pfad zu ISO Datei> fixate
```

Um zum Beispiel ein Image auf meinem IBM ThinkPad T42 zu brennen, gebe ich folgenden Befehl ein:

```
# burncd -f /dev/acd0 data /home/beat/6.0-BETA5-i386-disc1.iso fixate
```

burncd befindet sich im Basissystem und muss nicht extra installiert werden. burncd kann noch viel mehr als iso Images brennen, also unbedingt mal einen Blick in die Manpage burned(8) werfen.

### 1.3. Dateiflags in einer FreeBSD Jail nutzen

[ FreeBSD ] by Beat @ 07.10.2005 07:26 CET

Um Dateiflags, wie zum Beispiel schg oder sappnd, in einer Jail nutzen zu können, muss zuerst im Hostsystem die Variable security.jail.chflags\_allowed auf 1 gesetzt werden.

Beispiel:

```
root@host# sysctl security.jail.chflags_allowed
security.jail.chflags_allowed: 0

root@jail# ls -aol
-rw----- 1 beat users - 618 Oct 5 15:20 testdatei
root@jail# chflags schg testdatei
chflags: testdatei: Operation not permitted

root@host# sysctl security.jail.chflags_allowed=1
security.jail.chflags_allowed: 0 -> 1

root@jail# chflags schg testdatei
root@jail# ls -aol
```

## 1. FreeBSD

```
-rw----- 1 beat users schg 618 Oct 5 15:20 testdatei
root@jail# rm -f testdatei
rm: testdatei: Operation not permitted
```

### 1.4. Installierte Ports mit pkg\_cutleaves ausmisten

[ Befehl der Woche ] [ FreeBSD ] by Beat @ 29.10.2005 02:40 CET

Mit der Zeit sammeln sich installierte Ports an, die man vergessen hat oder nicht mehr braucht und von denen kein anderer Port abhängig ist. Zum Glück gibt es da pkg\_cutleaves.

Einfach den Port sysutils/pkg\_cutleaves installieren:

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

Danach einfach pkg\_cutleaves als root starten:

```
# pkg_cutleaves
```

pkg\_cutleaves führt nun alle Ports auf, von denen kein anderer Port abhängig ist und nun kann jeweils entschieden werden, ob man den Port löschen oder behalten will.

Mit folgenden Tasten wird pkg\_cutleaves gesteuert: [Enter] Lässt den Port installiert d Markiert den Port zur Löschung f Hebt alle Löschkennzeichnungen auf a Bricht pkg\_cutleaves ab, ohne Ports zu deinstallieren

Die zur Löschung markierten Ports werden deinstalliert, sobald man die Liste aller Ports, von denen kein weiterer abhängig ist, durchgearbeitet hat.

Die vollständige Liste kriegt man mit der -l Option:

```
# pkg_cutleaves -l
```

Ports, die in /usr/local/etc/pkg\_leaves.exclude aufgeführt sind, werden von pkg\_cutleaves nicht berücksichtigt, wenn pkg\_cutleaves mit der Option -x aufgerufen wird.:

```
# pkg_cutleaves -x
```

### 1.5. CPU throttling mit FreeBSD 6.0

[ FreeBSD ] by Beat @ 02.11.2005 18:59 CET

Um das CPU throttling mit FreeBSD 6.0 nutzen zu können, einfach das cpufreq Kernelmodul laden:

```
# kldload cpufreq
```

Um das Modul direkt beim Hochfahren zu laden, einfach folgenden Eintrag in /boot/loader.conf machen:

```
cpufreq_load="YES"
```

In /etc/rc.conf müssen nun die benötigten Variablen gesetzt werden:

```
powerd_enable="YES"
powerd_flags="-a max -b min"
```

Danach den powerd Daemon als root starten:

```
# /etc/rc.d/powerd start
```

Ist der powerd\_enable="YES" Eintrag in /etc/rc.conf gemacht, so wird der powerd Daemon beim Hochfahren automatisch gestartet. Mit der -a max Option läuft die CPU auf der maximalen Frequenz, solange der Laptop am Strom hängt. Mit der Option -b min, läuft die CPU auf der minimalsten Frequenz beim Batteriebetrieb. Statt min kann auch die Option adaptive verwendet werden, um die Taktfrequenz dynamisch der Last anzupassen.

Die aktuelle Frequenz kann mit folgendem Befehl ausgelesen werden:

```
# sysctl dev.cpu.0.freq
```

Die minimale Frequenz kann man mit folgender sysctl-Variable festgelegt werden:

```
# sysctl debug.cpufreq.lowest=<Minimale Frequenz>
```

Mehr Informationen zu powerd in der Manpage powerd(8)

## 1. FreeBSD

### 1.6. Herausfinden zu welchem Port eine Datei gehört

[ Befehl der Woche ] [ FreeBSD ] by Beat @ 05.11.2005 11:27 CET

Manchmal ist es ganz nützlich herauszufinden, zu welchem Port eine Datei gehört. Dies kann ganz einfach mit `pkg_which` gemacht werden.

```
# pkg_which <Datei>
```

Falls ich also wissen will, aus welchem Port `pkg_which` ist:

```
# pkg_which pkg_which
portupgrade-20041226_7 portupgrade-20041226_8
```

Wenn man nach einer Datei sucht, die nicht aus einem Port stammt, ruft `pkg_which which(1)` auf und durchsucht `PATH` nach der Datei.

`pkg_which` befindet sich, wie oben herausgefunden, im `sysutils/portupgrade` Port. Zum Installieren einfach folgenden Befehl benutzen:

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

Mehr Informationen zu `pkg_which` in der Manpage `pkgdb(1)`

### 1.7. WEP-verschlüsseltes WLAN mit DHCP nutzen

[ FreeBSD ] by Beat @ 06.11.2005 15:23 CET

Um in einem WEP-verschlüsselten WLAN beim Starten eine IP Adresse über DHCP zu erhalten, einfach folgende Zeile in `/etc/rc.conf` einfügen:

```
ifconfig_<WLAN Karte>="DHCP"
```

Danach die Datei `/etc/start_if.<WLAN Karte>` mit folgendem Inhalt erstellen:

```
ifconfig <WLAN Karte> wepkey <WEPkey> wepmode on ssid <SSID>
```

Wichtig: Den WEPkey in Hex angeben und auch das 0x am Anfang des Schlüssels nicht vergessen. Die Bezeichnung der WLAN Karte ergibt sich aus dem verwendeten Treiber der Karte. Eine Übersicht über alle verfügbaren Netzwerkkarten und deren Bezeichnung kriegt man mit `ifconfig -a`.

Auf FreeBSD 6.0 hat es einige Änderungen bezüglich wlan und ath, gegenüber FreeBSD 5.4 gegeben. Hier mehr dazu: <http://tigion.bsd-crew.de/blog/?postid=141> Danke an Mich für die Informationen.

Folgende Kerneloptionen werden auf FreeBSD 6.0 benötigt:

```
device wlan
device wlan_wep
```

`/etc/start_if.<WLAN Karte>` sieht auf FreeBSD 6.0 so aus:

```
ifconfig <WLAN Karte> wepkey 1:<WEPkey> wepmode on weptxkey 1 ssid <SSID>
```

### 1.8. Platz schaffen mit portsclean

[ Befehl der Woche ] [ FreeBSD ] by Beat @ 12.11.2005 06:17 CET

Mit der Zeit wird der Platz auf der Festplatte immer kleiner, da kann man zum Beispiel mit `pkg_cutleaves` die Ports aufräumen oder mit `portsclean` den Portbaum säubern. Zuerst kann der Portbaum auf den neusten Stand gebracht werden, damit die veralteten Quellcodedateien gelöscht werden können. Dies kann man zum Beispiel mit `cvsup` machen. Danach einfach folgenden Befehl als root ausführen:

```
# portsclean -CDP
```

-C Löscht alle Arbeitsverzeichnisse aus dem Portbaum

-D Löscht alle veralteten Quellcodedateien aus dem `/usr/ports/distfiles` Verzeichnis

-P Löscht alle veralteten Pakete aus dem `/usr/ports/packages` Verzeichnis

Mehr Informationen zu `portsclean` in der Manpage `portsclean(1)`.

## 1.9. Ports aktualisieren mit dem gnome\_upgrade Skript

[ FreeBSD ] by Beat @ 13.11.2005 16:49 CET

Am 05.11.05 wurden laut /usr/ports/UPDATING gtk/glib und GNOME aktualisiert.

```
20051105:
  AFFECTS: all users of gnome/glib/gtk
  AUTHOR: gnome@FreeBSD.org

  GNOME has been updated to 2.12 and gtk/glib to 2.8.
  DO NOT USE portupgrade(1) to update your ports. Instead, use
  the gnome_upgrade.sh script from
  http://www.marcuscom.com/downloads/gnome_upgrade212.sh
```

Zeit, die Ports auf meinem Laptop auf den neusten Stand zu bringen. Also gleich mal das gnome\_upgrade212.sh heruntergeladen:

```
# fetch http://www.marcuscom.com/downloads/gnome_upgrade212.sh
```

Als erstes habe ich Openoffice mit pkg\_deinstall gelöscht, da ich nicht wollte, dass das Upgrade-Skript Openoffice aktualisiert.

Mit der Option -h alle möglichen Optionen angeschaut:

```
#!/gnome_upgrade212.sh -h
usage: ./gnome_upgrade212.sh [-f] [-k] [-h] [-P] [-p] [-v] [-restart <list
  filename>]

  -f      : Do not prompt for any confirmations (think: force)
  -k      : Keep going even if an error is encountered
  -h      : Print this usage summary and exit
  -p      : Use packages for upgrades where possible
  -P      : Only use packages for upgrades
  -v      : Print the version of ./gnome_upgrade212.sh and exit
  -restart : Restart a failed upgrade using the <list filename> to get the list
            of ports that still require an upgrade
```

...und danach das Skript ohne eine Option gestartet. Nach der Bestätigung mit Enter, dass ich das Upgrade durchführen möchte, begann das Skript mit der Aktualisierung der Ports.

Mit folgendem Befehl kann der Fortschritt des Upgrades mitgelesen werden:

```
# tail -f /var/tmp/gnome_upgrade_log.<6 zufällige Zeichen>
```

Nachdem das Skript schon 2 Stunden und 55 Minuten am Aktualisieren war, kam das:

```
** Listing the failed packages (*:skipped / !:failed)
  ! multimedia/mplayer-skins (mplayer-skins-1.1.1_1)      (checksum
    mismatch)
  ! misc/help2man (help2man-1.35.1)      (unknown build error)
  ! textproc/intltool (intltool-0.33)    (unknown build error)
  ! misc/gnome-icon-theme (gnome-icon-theme-2.10.1_1)    (configure error)
  * multimedia/mplayer (mplayer-gtk-esound-0.99.7_4)
---> Packages processed: 63 done, 0 ignored, 1 skipped and 4 failed
INFO: GNOME upgrade FAILED at Sat Nov 12 01:14:07 CET 2005
```

Für mplayer konnten nicht die richtigen Dateien heruntergeladen werden, also gleich mplayer deinstalliert:

```
# pkg_deinstall -R -d -v mplayer-gtk-esound
```

Die anderen drei Ports, die sich nicht aktualisieren liessen, versuchte ich mit portupgrade -f <port-name> manuell zu aktualisieren. Wo dies nicht funktionierte, half ein make deinstall && make reinstall. Danach das gnome\_upgrade212.sh wieder gestartet:

```
# ./gnome_upgrade212.sh -restart /var/tmp/gnome_upgrade.lst.<6 zufällige Zeichen>
```

Und nach weiteren 4 Stunden und 13 Minuten aktualisieren und total 108 aktualisierten Ports, die lang ersehnte Meldung:

```
INFO: GNOME upgrade finished successfully at Sat Nov 12 06:17:45 CET 2005
```

Danach Openoffice mit pkg\_add wieder installiert und ebenso mplayer ohne die Skins, die nicht korrekt heruntergeladen werden konnten.

## 1. FreeBSD

### 1.10. Überblick über alle installierten Ports

[ Befehl der Woche ] [ FreeBSD ] by Beat @ 18.11.2005 15:21 CET

Um einen Überblick über alle installierten Ports zu erhalten, einfach folgenden Befehl benutzen:

```
# portversion
```

Zusätzlich wird bei jedem Port angezeigt, ob sich im lokalen Portbaum eine neuere Version befindet oder ob die installierte Version aktuell ist.

Ein bisschen mehr Informationen bekommt man mit dem verbose Modus von portversion:

```
# portversion -v
```

portversion befindet sich im sysutils/portupgrade Port. Mehr Informationen findet man in der Manpage portversion(1)

### 1.11. Neuer Port: jailaudit

[ FreeBSD ] by Beat @ 25.11.2005 13:52 CET

Gerade auf freshports diesen neuen Port entdeckt: jailaudit

```
This port contains a script to generate portaudit reports for jails running on a FreeBSD system.
```

```
Normally portaudit just creates reports for the Host-system or the jail it is installed in. With a large number of jails running on a system, installing and updating portaudit in every jail is time-consuming and error-prone. Jailaudit uses the portaudit installed in the Host-system to create a report for every jail.
```

```
The reports are appended to the daily security run and can be sent to a specific mail address, which allows the Host-system administrator to dispatch portaudit reports to the owners of jails, keeping them informed about potential security advisories of their installed ports.
```

jailaudit kann nur auf einem FreeBSD 5.1 oder neuer verwendet werden, da jls benötigt wird.

### 1.12. FreeBSD Ports richtig deinstallieren

[ Befehl der Woche ] [ FreeBSD ] by Beat @ 17.12.2005 13:17 CET

Wenn man einen Port auf FreeBSD deinstallieren möchte, ist es empfehlenswert, diesen mittels pkg\_deinstall oder pkg\_delete zu entfernen, da so dieser nur entfernt wird, falls keine anderen Programme mehr von ihm abhängig sind. Ein make deinstall im Portverzeichnis achtet nicht auf Abhängigkeiten, was zu Problemen führen kann.

pkg\_delete befindet sich bereits in Basissystem. Möchte man einen Port deinstallieren und gleichzeitig auch alle Ports, von denen nur der zu deinstallierende Port abhängig ist, lohnt es sich sysutils/portupgrade zu installieren. Darin befindet sich pkg\_deinstall. Damit können Ports auch aufwärts rekursiv deinstalliert werden. Um also einen Port samt allen nur von ihm gebrauchten Abhängigkeiten zu deinstallieren verwende ich folgenden Befehl:

```
# pkg_deinstall -R -d -v <Portname>
```

Die Option -d entfernt alle leeren Verzeichnisse des Portes und mit -v kriegt man ein bisschen mehr Informationen.

### 1.13. Beastie Menu unter FreeBSD 6.0

[ FreeBSD ] by Beat @ 20.12.2005 17:36 CET In FreeBSD 6.0 wurde der BSD Daemon im Startbildschirm durch einen FreeBSD Schriftzug ersetzt. Dieser kann durch einen der folgenden Einträge in /boot/loader.conf wieder ersetzt werden:



## 1. FreeBSD

```
loader_logo="fbstdbw" # Standard; schwarz/weiss Darstellung des Wortes FreeBSD
loader_logo="beastiebw" # Der FreeBSD 5 Standard; der schwarz/weiss BSD Daemon
loader_logo="beastie" # Mein Standard; farbiger BSD Daemon
loader_logo="none" # Keine Grafik
```

Quelle: <http://www.bsdforums.org/forums/showthread.php?t=36124>

### 1.14. Systemstatistiken auf der Konsole mit sysstat

[ Befehl der Woche ] [ FreeBSD ] by Beat @ 24.12.2005 11:46 CET Im FreeBSD Basissystem gibt es ein Programm, mit dem man sich ganz einfach Systeminformationen und Statistiken auf der Konsole ausgeben lassen kann.

```
# sysstat -<Option>
```

Folgende Optionen sind möglich: pigs: Zeigt den Prozess, der den grössten Anteil der CPU konsumiert icmp: Zeigt Statistiken über empfangene und gesendete ICMP Meldungen ip: Zeigt Statistiken über empfangene und gesendete IP Pakete tcp: Zeigt Statistiken über empfangene und gesendete TCP Pakete iostat: Zeigt Informationen zur CPU Auslastung und IO Aktivitäten der Festplatten swap: Zeigt Informationen über die Benutzung des Swap Bereiches vmstat: Zeigt Informationen zur virtuellen Speicherverwaltung netstat: Zeigt Informationen zu den aktuellen Netzwerkverbindungen ifstat: Zeigt Informationen zu den Netzwerkschnittstellen

Innerhalb von sysstat kann auch mit :<Option> die Ansicht gewechselt werden. Mit :q verlässt man sysstat. Mehr Informationen dazu gibt es wie immer in der Manpage zu sysstat.

### 1.15. Flashplugin für Firefox auf FreeBSD

[ FreeBSD ] by Beat @ 30.12.2005 11:34 CET

Um das Flashplugin für den Firefox benutzen zu können, muss natürlich zuerst der Firefox installiert werden:

```
# cd /usr/ports/www/firefox && make install clean
```

Danach den Linuxpluginwrapper installieren:

```
# cd /usr/ports/www/linuxpluginwrapper && make install clean
```

Zu guter Letzt kann das Flashplugin installiert werden.

```
# cd /usr/ports/www/linux-flashplugin6 && make install clean
```

Danach je nach verwendetem FreeBSD die libmap.conf kopieren. Auf FreeBSD 4.x:

```
# cp /usr/local/share/examples/linuxpluginwrapper/libmap.conf-FreeBSD4.x
/etc/libmap.conf
```

Für FreeBSD 5.0 bis 5.3-BETA6 nimmt man libmap.conf-FreeBSD5-current:

```
# cp /usr/local/share/examples/linuxpluginwrapper/libmap.conf-FreeBSD5-current
/etc/libmap.conf
```

Für FreeBSD 5.3-BETA7 und neuer nimmt man FreeBSD5-stable:

```
# cp /usr/local/share/examples/linuxpluginwrapper/libmap.conf-FreeBSD5-stable
/etc/libmap.conf
```

Auf einem FreeBSD 6 System nimmt man natürlich libmap.conf-FreeBSD6:

```
# cp /usr/local/share/examples/linuxpluginwrapper/libmap.conf-FreeBSD6
/etc/libmap.conf
```

Danach noch einen Symlink erstellen:

```
# ln -s /usr/X11R6/lib/linux-flashplugin6/libflashplayer.so
/usr/X11R6/lib/browser_plugins/libflashplayer.so
```

Und nach einem Neustart von Firefox sollte das Plugin funktionieren. Ein about:plugins in der Adresszeile des Firefox listet alle aktivierten Plugins auf.

## 1.16. Manpages

[ Befehl der Woche ] [ FreeBSD ] by Beat @ 31.12.2005 10:36 CET

Da ich oft auf die Manpages verweise, möchte ich den letzten Befehl der Woche dieses Jahres den Manpages widmen. Die Manpages sind in 9 verschiedene Sections gegliedert. Section 1: General Commands Section 2: System Calls Section 3: Library Functions Section 4: Kernel Interfaces Section 5: File Formats Section 6: Games Section 7: Miscellaneous Information Section 8: System Manager's Manual Section 9: Kernel Developer's Manual

Jede Section besitzt eine Einleitung. Diese kann mit folgendem Befehl aufgerufen werden:

```
# man <Sectionnummer> intro
```

Eine Manpage ruft man grundsätzlich wie folgt auf, wobei die Sectionnummer optional ist:

```
# man <Sectionnummer> <Name der Manpage>
```

Wählt man eine Manpage, die in verschiedenen Sections vorkommt, jedoch ohne eine Sectionnummer anzugeben, so wird nur die erste gefundene Manpage angezeigt. Gibt man zum Beispiel nur `man intro` ein, kriegt man nur die Intro-Manpage der ersten Section. Mit der Option `-a` werden alle gefundenen Seiten nacheinander dargestellt:

```
# man -a intro
```

Wird auf eine Manpage verwiesen, so verwendet man üblicherweise folgende Schreibweise: `Befehl(Section)`. `fortune(6)` bedeutet dementsprechen, dass die Manpage von `fortune` sich in Section 6 befindet.

Die Manpages sind auch im Internet verfügbar: <http://www.freebsd.org/cgi/man.cgi>

Folgende Manpages sind immer mal einen Blick wert:

- `rc.conf(5)` Übersicht und Beschreibung der `rc.conf` Systemvariablen
- `make.conf(5)` Übersicht und Beschreibung der `make.conf` Variablen
- `login.conf(5)` Übersicht und Beschreibung der `login.conf` Variablen
- `hier(7)` Aufbau der FreeBSD Verzeichnisstruktur
- `tuning(7)` Tipps zur Leistungssteigerung eines FreeBSD Systemes
- `ports(7)` Hilfe zu den Ports und Beschreibung aller `make targets` für den Portbaum
- `security(7)` Tipps zum Absichern eines FreeBSD Systemes
- `firewall(7)` Wie man eine einfache Firewall erstellt
- `build(7)` Beschreibung der `make targets` um ein System zu aktualisieren/erstellen
- `style(9)` Richtlinien für Kernelcode

Mehr Informationen zu `man` natürlich in der Manpage: `man(1)`

## 1.17. su: not running setuid

[ FreeBSD ] by Beat @ 01.01.2006 15:04 CET

Hat man eine neue Jail erstellt und möchte einem `wheel`-Benutzer mit `su` zu `root`-Rechten verhelfen, aber dies mit folgender Meldung verweigert wird:

```
# su
su: not running setuid
```

so hat man die Jail auf einer Partition erstellt, die mit der `nosuid` Option gemountet wurde.

## 1.18. Port auf einen älteren Stand zurücksetzen mit portdowngrade

[ FreeBSD ] by Beat @ 01.01.2006 16:50 CET

Nachdem ich Sunbird auf 0.3a1 aktualisierte, waren alle Termine verschwunden und Sunbird weigerte sich standhaft die Termine zu importieren. Also musste die alte Sunbird Version wieder her.

Deshalb gleich sysutils/portdowngrade installiert:

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

Sunbird deinstalliert:

```
# pkg_deinstall -f linux-sunbird
```

Und dann deskutils/linux-sunbird wieder auf den alten Stand gebracht:

```
# setenv CVSR00T :pserver:anoncvcs@anoncvcs2.de.FreeBSD.org:/home/ncvs
# cvs login
Logging in to :pserver:anoncvcs@anoncvcs2.de.freebsd.org:2401/home/ncvs
CVS password: anoncvcs
# portdowngrade linux-sunbird

portdowngrade 0.6 by Heiner Eichmann
Please note, that nothing is changed in the ports tree
unless it is explicitly permitted in step 6!

Seeking port linux-sunbird ... found: deskutils/linux-sunbird

Step 1: Checking out port from CVS repository
CVS root directory (from CVSR00T environment variable):
      :pserver:anoncvcs@anoncvcs2.de.FreeBSD.org:/home/ncvs

Step 2: Reading the port history from the CVS repository

Step 3: Analyzing the port history from the CVS repository

Step 4: Load port version numbers and present results
Keys:  space  : next page                      d : details
p : previous page
enter  : leave presentation and downgrade if wanted
=====
number      date      portversion  comment
1   2005/12/07 13:13:31  0.3a1_1    - Update to 0.3a1
2   2005/06/17 22:59:27  0.2_1      Mega-patch to cleanup the ports infrastructure
      regarding our linux bits:
3   2005/03/01 22:24:25  0.2        new port: misc/linux-sunbird

Total lines: 3. Command: [Enter]
Enter version number to change port to (0: exit): 2

Step 5: Checking out choosen date of the port from the CVS repository

Step 6: Modifying the port
Port: deskutils/linux-sunbird
at : 2005/06/17 22:59:27
Type 'yes' to bring the port to the state of the date above
or 'no' to exit without changing anything. Note, that this only changes
the port, not the installed software! yes or no:yes

The port has been set to the selected version. Install it if you wish.
If you have portupgrade installed, you should run

portsdb -Uu

now, to see the changes in the ports database. In any case

portupgrade -f linux-sunbird
```

## 1. FreeBSD

will install the changed port. Note: if you run cvsup, the port is changed back to the chosen label!

Danach linux-sunbird wieder installiert und alle Termine waren wieder da.

```
# cd /usr/ports/deskutils/linux-sunbird && make install clean
```

### 1.19. Live-CD's mit QEMU emulieren

[ Befehl der Woche ] [ FreeBSD ] by Beat @ 07.01.2006 11:05 CET

Falls man eine Live-CD testen möchte, braucht man nicht das ISO-Image runterladen, eine CD brennen, den Rechner neu starten - denn dafür gibt es QEMU.

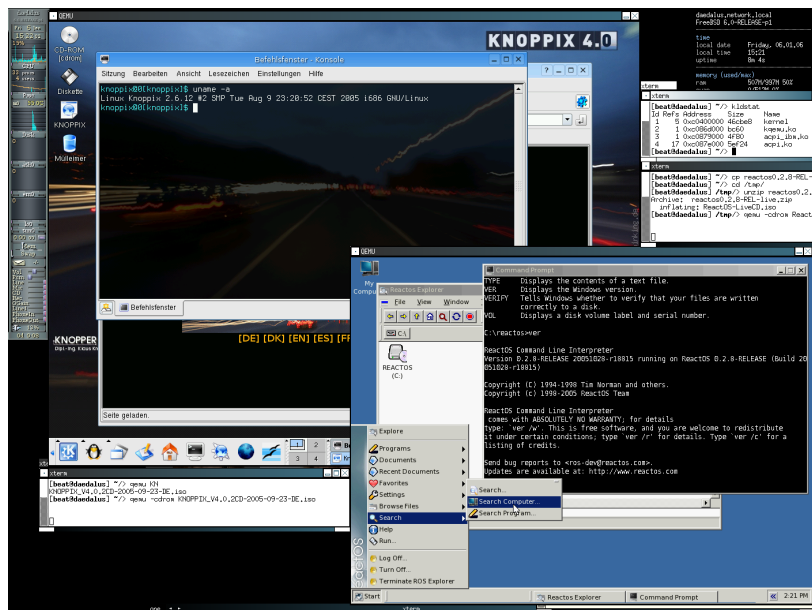


Abbildung 2: FreeBSD 6.0, auf dem ReactOS 0.2.8 und Knoppix 4.0.2 mit Hilfe von QEMU emuliert werden.

QEMU kann ganz einfach aus den Ports installiert werden:

```
# cd /usr/ports/emulators/qemu && make install clean
```

Um die Geschwindigkeit von QEMU merkbar zu beschleunigen, kann man das QEMU Kernel Modul installieren. Zur Installation werden die Kernel Sourcen benötigt.

```
# cd /usr/ports/emulators/kqemu-kmod && make install clean
```

Danach kann das Kernel Modul geladen werden:

```
# kldload kqemu.ko
```

Wird QEMU öfter gebraucht, so kann man sich das Kernel Modul auch beim Starten laden, indem man folgende Zeile in der /boot/loader.conf hinzufügt:

```
kqemu_load="YES"
```

Wichtig: Wird ein neuer Kernel installiert, so muss man das QEMU Kernel Modul aus dem alten Kernel kopieren, falls man eine gleiche Kernelversion benutzt:

```
# cp /boot/kernel.old/kqemu.ko /boot/kernel/
```

Benutzt man einen anderen Kernel, so muss der kqemu-kmod Port neu gebaut werden. Danach die gewünschte Live-CD herunterladen und diese mit folgendem Befehl starten:

## 1. FreeBSD

```
# qemu -cdrom <Pfad zum ISO Image>
```

Möchte man dem Live-CD System 512MB Arbeitsspeicher zur Verfügung stellen, das Tastatur-layout auf Schweizerdeutsch einstellen und erlauben, dass die Live-CD auch Geräusche von sich geben kann, so verwendet man folgenden Befehl:

```
# qemu -m 512 -k de-ch -enable-audio -cdrom <Pfad zum ISO Image>
```

Mit QEMU sind noch ganz viele interessante Sachen möglich, so kann man zum Beispiel ein Windows in einem FreeBSD installieren: [http://www.bsdbox.de/?page\\_id=18](http://www.bsdbox.de/?page_id=18) und dieses ins eigene Netzwerk integrieren: [http://www.bsdbox.de/?page\\_id=25](http://www.bsdbox.de/?page_id=25)

Weitere Informationen zu QEMU unter <http://fabrice.bellard.free.fr/qemu/>

### 1.20. Portbaum aktualisieren mit Portsnap

[ Befehl der Woche ] [ FreeBSD ] by Beat @ 21.01.2006 10:01 CET

Alternativ zu CVSup kann der Portbaum auch mit portsnap auf den neusten Stand gebracht werden. Seit FreeBSD 6.0-BETA3 befindet sich portsnap im nun FreeBSD Basissystem. Welche Vorteile portsnap gegenüber CVSup haben soll, kann man hier nachlesen:

<http://www.daemonology.net/portsnap/>

Benutzt man portsnap das erste mal, so muss man den Snapshot zuerst herunterladen:

```
# portsnap fetch
Looking up portsnap.FreeBSD.org mirrors... using portsnap1.FreeBSD.org.
Fetching public key... done.
Fetching snapshot tag... done.
Fetching snapshot metadata... done.
Fetching snapshot generated at Mi 11 Jan 2006 01:32:11 CET:
0d1d343821c1a727b12b377de9697b248fee48a81ea749100% of 38 MB 73 kBps 00m00s
Extracting snapshot... done.
Verifying snapshot integrity... done.
Fetching snapshot tag... done.
Fetching snapshot metadata... done.
Updating from Mi 11 Jan 2006 01:32:11 CET to Mi 11 Jan 2006 19:57:38 CET.
Fetching 4 metadata patches... done.
Applying metadata patches... done.
Fetching 0 metadata files... done.
Fetching 94 patches....10....20....30....40....50....60....70....80....90.. done.
Applying patches... done.
Fetching 10 new ports or files... done.
```

Danach kann er mit folgendem Befehl entpackt werden:

```
# portsnap extract
```

Nachdem man dies einmal gemacht hat, kann man den Portbaum nun ganz einfach auf einen neueren Stand bringen:

```
# portsnap fetch update
```

Besitzt man eine FreeBSD Version in der portsnap noch nicht im Basissystem ist, also eine vor 6.0-BETA3 oder 7.0-CURRENT, so kann man portsanp alternativ aus den Ports installieren:

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

Hat man den Portbaum noch nicht installiert, so kann man auch das Paket von portsnap installieren:

```
# pkg_add -r portsnap
```

Mehr Informationen hier: [portsnap\(8\)](#)

## 1. FreeBSD

### 1.21. Flux CMS Port für FreeBSD

[ FreeBSD ] by Beat @ 24.01.2006 12:17 CET Ich habe einen FreeBSD Port für das Flux CMS geschrieben. Hier mal eine kleine Anleitung wie man den Port installieren kann.

```
# cd /usr/ports/www/fluxcms && make install clean
```

Möchte man das Flux CMS lieber als Packet installieren, so macht man dies folgendermassen:

```
# pkg_add -r fluxcms-1.4.0.tbz
```

Ist noch kein Apache installiert, so muss zuerst die httpd.conf den eigenen Wünschen angepasst, PHP5 aktiviert und der Apache in der /etc/rc.conf eingetragen werden.

```
# echo 'apache_enable="YES"' >> /etc/rc.conf
```

Danach den Apache starten.

```
# /usr/local/etc/rc.d/apache.sh start
```

Ist noch kein MySQL Server installiert, so muss dieses noch gemacht werden:

```
# cd /usr/ports/databases/mysql41-server && make install clean
# echo 'mysql_enable="YES"' >> /etc/rc.conf
```

Auch danach muss der MySQL Server gestartet werden.

```
# /usr/local/etc/rc.d/mysql-server.sh start
```

Ausserdem sollte für das Flux CMS ein VirtualHost gemäss den Anweisungen des Ports erstellt werden. Sind die Anweisungen des Ports nicht mehr sichtbar, so können diese ganz einfach angezeigt werden:

```
# cd /usr/ports/www/fluxcms && make post-install
```

Danach den Apache neu starten und den Webinstaller vom Flux CMS ausführen.

### 1.22. Dateien mit allen Dateiattributen kopieren

[ Befehl der Woche ] [ FreeBSD ] by Beat @ 28.01.2006 02:18 CET

Möchte man Dateien kopieren, die Dateiattribute der neu erstellten Dateien jedoch identisch mit den Originalen belassen, so kann man dies einfach mit cp -p machen. Dies kopiert auch alle Attribute wie die Zeitstempel, Dateiflags, Dateiberechtigungen, User ID und Group ID.

Beispiel:

Datei eines Benutzers wird von root nur mit cp kopiert:

```
# ls -ol
-rwxrwxrwx 1 beat wheel schg 9 27 Jan 16:32 testdatei*
# cp testdatei testdatei2
# ls -ol
-rwxrwxrwx 1 beat wheel schg 9 27 Jan 16:32 testdatei*
-rwxr-xr-x 1 root wheel - 9 27 Jan 16:35 testdatei2*
```

Die selbe Datei wird von root mit cp -p kopiert:

```
# cp -p testdatei testdatei3
# ls -ol
-rwxrwxrwx 1 beat wheel schg 9 27 Jan 16:32 testdatei*
-rwxr-xr-x 1 root wheel - 9 27 Jan 16:35 testdatei2*
-rwxrwxrwx 1 beat wheel schg 9 27 Jan 16:32 testdatei3*
```

Soll zusätzlich zum Kopieren noch überprüft werden, ob die Datei richtig kopiert wurde, so empfiehlt es sich cpdup zu verwenden. cpdup(1) findet man in den Ports unter sysutils/cpdup. Kopiert man eine Datei mit cpdup -m, so generiert cpdup einen MD5 Hash der Datei, kopiert die Datei und vergleicht den MD5 Hash der neu erstellten Datei mit dem Original. Stimmen diese nicht überein, so wird die Datei erneut kopiert.

## 1.23. Lokale Mails über einen Mail-Relay versenden

[ FreeBSD ] by Beat @ 02.02.2006 21:11 CET

Möchte man zum Beispiel aus einem Heimnetzwerk von der Konsole aus ein Mail verschicken und den Mail-Relay seines Providers verwenden, damit so etwas nicht passiert:

```
stat=Deferred: 450 <daedalus.network.local>: Helo command rejected: Host not found
```

Dann kann man sich entweder mit sendmail rumärgern oder ssmtp verwenden:

```
# cd /usr/ports/mail/ssmtp
# make install replace clean
```

Falls sendmail noch läuft, sendmail beenden:

```
# /etc/rc.d/sendmail stop
```

Danach sendmail in der rc.conf deaktivieren:

```
sendmail_enable="NONE"
```

Jetzt noch die /usr/local/etc/ssmtp/ssmtp.conf anpassen:

```
root=<Meine mailadresse>
mailhub=<SMTP meines Providers>
rewriteDomain=<Meine Domain>
hostname=_HOSTNAME_
```

Und schon funktioniert das Senden von Konsolenmails.

Gefunden bei: [http://www.freebsd.org/doc/de\\_DE.IS08859-1/books/handbook/outgoing-only.html](http://www.freebsd.org/doc/de_DE.IS08859-1/books/handbook/outgoing-only.html)

## 1.24. Installierte Ports auf Sicherheitsprobleme prüfen

[ FreeBSD ] [ Befehl der Woche ] by Beat @ 04.02.2006 08:01 CET

Möchte man überprüfen, ob die installierten Ports Sicherheitsprobleme aufweisen, so empfiehlt es sich security/portaudit zu installieren:

```
# cd /usr/ports/security/portaudit && make install clean
```

Danach kann portaudit in der Konsole ausgeführt werden:

```
# portaudit -Fda
auditfile.tbz                                100% of   32 kB   43 kBps
New database installed.
Database created: Fr  3 Feb 2006 16:10:03 CET
0 problem(s) in your installed packages found.
```

Die Option -F lädt die aktuellste Datenbank herunter, in der alle aktuellen Sicherheitsprobleme der Ports verzeichnet sind, -d zeigt die Zeit an, wann die Datenbank erstellt wurde und -a prüft alle installierten Ports auf Sicherheitsprobleme.

Im folgenden Beispiel sollte ein Update gewisser Ports in Betracht gezogen werden:

```
# portaudit -Fda
auditfile.tbz                                100% of   32 kB   36 kBps
New database installed.
Database created: Sat Feb  4 00:10:03 CET 2006
Affected package: clamav-0.86.2_1
Type of problem: clamav -- possible heap overflow in the UPX code.
Reference:
  http://www.FreeBSD.org/ports/portaudit/612a34ec-81dc-11da-a043-0002a5c3d308.html

Affected package: p5-Mail-SpamAssassin-3.0.4
Type of problem: p5-Mail-SpamAssassin -- long message header denial of service.
Reference:
  http://www.FreeBSD.org/ports/portaudit/7f3fdef7-51d2-11da-8e93-0010dc4afb40.html

Affected package: squid-2.5.10_5
```

## 1. FreeBSD

```
Type of problem: squid -- FTP server response handling denial of service.
Reference:
    http://www.FreeBSD.org/ports/portaudit/1c3142a3-4ab2-11da-932d-00055d790c25.html

Affected package: ruby-1.8.2_4
Type of problem: ruby -- vulnerability in the safe level settings.
Reference:
    http://www.FreeBSD.org/ports/portaudit/1daea60a-4719-11da-b5c6-0004614cc33d.html

Affected package: clamav-0.86.2_1
Type of problem: clamav -- arbitrary code execution and DoS vulnerabilities.
Reference:
    http://www.FreeBSD.org/ports/portaudit/271498a9-2cd4-11da-a263-0001020eed82.html

Affected package: unzip-5.51
Type of problem: unzip -- permission race vulnerability.
Reference:
    http://www.FreeBSD.org/ports/portaudit/9750cf22-216d-11da-bc01-000e0c2e438a.html

Affected package: pcre-5.0
Type of problem: pcre -- regular expression buffer overflow.
Reference:
    http://www.FreeBSD.org/ports/portaudit/b971d2a6-1670-11da-978e-0001020eed82.html

Affected package: squid-2.5.10_5
Type of problem: squid -- possible denial of service condition regarding NTLM
authentication.
Reference:
    http://www.FreeBSD.org/ports/portaudit/44e7764c-2614-11da-9e1e-c296ac722cb3.html

8 problem(s) in your installed packages found.

You are advised to update or deinstall the affected package(s) immediately.
```

Ausserdem installiert portaudit folgende Datei /usr/local/etc/periodic/security/410.portaudit, welche täglich während den daily security Skripten ausgeführt wird und die installierten Ports automatisch auf Sicherheitslöcher prüft.

Desweiteren wird nun bevor man einen Port installiert automatisch überprüft, ob der installierte Port Sicherheitslöcher aufweist.

Eine Weboberfläche der Datenbank findet man hier: <http://www.vuxml.org/freebsd/index.html>

### 1.25. Portbaum einfach durchsuchen mit portsearch

[ Befehl der Woche ] [ FreeBSD ] by Beat @ 18.02.2006 10:15 CET

Möchte man den Portbaum durchsuchen, so kann man einfach portsearch zur Hilfe nehmen.

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

Nach was man den Portbaum mit portsearch alles durchsuchen kann, zeigt ein Aufruf von Portsearch ohne Parameter:

```
# portsearch
Usage:

update/create database:
portsearch -u [-H portshome] [-vvv]

search for ports (based on extended regular expressions, case sensitive):
-n name      by name (name= can be used)
-k key       by name, comment or dependencies (key= can be used)
-p path      by path on the filesystem
-i info      by info (comment)
-m maint     by maintainer
-c cat       by category
-F fdep      by fetch dependencies
```



## 1. FreeBSD

```
-E edep      by extract dependencies
-P pdep      by patch dependencies
-B bdep      by build dependencies
-R rdep      by run dependencies
-D dep       by build or run dependencies
-w www       by www site
-f file      that install file
-I           ignore case
-o fields    output fields, default: name,path,info,maint,bdep,rdep,www
-V           print version information
```

Bevor man portsearch das erste Mal benutzt, muss eine Datenbank für portsearch erstellt werden. Dies kann eine Weile dauern:

```
# portsearch -u
```

Ist die Datenbank erstellt, so kann man portsearch benutzen. Möchte man nun z.B. herausfinden in welchem Port sich portversion(1) befindet, so kann portsearch mit der -f Option genutzt werden:

```
# portsearch -f portversion
Port:   portupgrade-2.0.1_1,1
Path:   /usr/ports/sysutils/portupgrade
Info:   FreeBSD ports/packages administration and management tool suite
Maint:  koma2@lovepeers.org
B-deps: ruby-1.8.4_4,1
R-deps: perl-5.8.7_2 ruby-1.8.4_4,1 ruby18-bdb1-0.2.2
WWW:
Files:  man/man1/portversion.1.gz, sbin/portversion

1 ports, 2 files
```

So kann der Portbaum effektiv durchsucht und genutzt werden.

### 1.26. iPod mit FreeBSD benutzen

[ FreeBSD ] by Beat @ 28.02.2006 14:54 CET

Um meinen 20GB iPod der 3ten Generation mittels USB unter FreeBSD zu benutzen, müssen zuerst die Device-Berechtigungen angepasst werden. Dazu muss folgender Eintrag in die /etc/devfs.rules gemacht werden, wobei beim group Eintrag, die Gruppe der Benutzer eingetragen werden muss. Bei mir ist dies users:

```
[ipod=5]
add path 'da*s*' mode 0660 group users
add path 'xpt0'  mode 0660 group users
add path 'pass0' mode 0660 group users
```

Damit die Device-Berechtigungen während des Betriebes angepasst werden, muss devfs in der /etc/rc.conf aktiviert werden:

```
devfs_enable="YES"
devfs_rulesets="/etc/devfs.rules"
devfs_system_ruleset="ipod"
```

Damit Benutzer den iPod mounten können, muss die usermount Variable in der /etc/sysctl.conf gesetzt werden:

```
vfs.usermount=1
```

Danach das Verzeichnis erstellen, wo der iPod später ins System gemountet werden soll, und die Berechtigungen setzen, wobei beat:users durch den Benutzernamen und die Gruppe des lokalen Benutzers zu ersetzen ist:

```
# mkdir /mnt/ipod
# chown beat:users /mnt/ipod
```

Um das Mounten zu vereinfachen, kann noch ein Eintrag für den iPod in der /etc/fstab gemacht werden:

## 1. FreeBSD

```
/dev/da0s2          /mnt/ipod          msdos    rw,noauto      0      0
```

Jetzt kann der Rechner neu gestartet oder die Änderungen von Hand aktiviert werden:

```
# sysctl vfs.usermount=1
# /etc/rc.d/devfs start
```

Nun kann der iPod einfach als Benutzer gemountet werden:

```
# mount /mnt/ipod
```

Um den iPod zu verwalten, benutze ich gnupod. gnupod befindet sich unter audio/gnupod in den FreeBSD Ports. gnupod ist eine Sammlung von Perl-Skripten, um den iPod auf der Kommandozeile zu bedienen.

Danach muss der iPod initialisiert werden. Die vorhandenen Lieder und die Datenbank werden dabei übernommen und gehen nicht verloren. Dies muss nur vor dem ersten Gebrauch von gnupod gemacht werden:

```
# gnupod_INIT.pl -m /mnt/ipod
```

Um Lieder auf den iPod zu laden, kann man das gnupod\_addsong.pl Skript verwenden. Dabei können auch Wildcards verwendet werden, um mehrere Lieder auf einmal auf den iPod zu laden.

```
# gnupod_addsong.pl -m /mnt/ipod/ <Pfad zu den Liedern>
```

Sind alle Lieder auf den iPod geladen, muss noch die Datenbank des iPods aktualisiert werden.

```
# mktunes.pl -m /mnt/ipod/
```

Danach kann der iPod geunmounted (gibts dieses Wort?) werden:

```
# umount /mnt/ipod/
```

Auf dem iPod Display steht aber weiterhin: Bitte nicht trennen

Um den iPod aus dem System zu entfernen, muss zuerst mit camcontrol devlist herausgefunden werden, wo der iPod im System eingebunden ist.

```
# camcontrol devlist
<Apple iPod 1.50>                at scbus0 target 0 lun 0 (da0,pass0)
```

Damit camcontrol eject funktioniert, muss im Kernel das pass device vorhanden sein. Dies ist im GENERIC-Kernel schon der Fall. Hat man einen selbst erstellten Kernel, sollte folgender Eintrag in der Kernel-Konfigurationsdatei vorhanden sein, aus der man den Kernel erzeugt hat:

```
device          pass          # Passthrough device (direct SCSI access)
```

Danach kann der iPod aus dem System entfernt werden. Die Device ID muss nicht auf jedem System 0:0:0 sein. Um die Device ID herauszufinden, musste vorher ein camcontrol devlist gemacht werden. In diesem Beispiel wird scbus0 target 0 lun 0 zu 0:0:0

```
# camcontrol eject 0:0:0
```

Auf dem iPod steht nun iPod kann getrennt werden und das USB-Kabel kann nun abgezogen werden.

### 1.27. Einen schnellen CVSupServer finden mit fastest\_cvsup

[ Befehl der Woche ] [ FreeBSD ] by Beat @ 04.03.2006 11:56 CET

Um das Aktualisieren, von z.B. des Portbaums, mit CVSup oder csup zu beschleunigen, kann man mit fastest\_cvsup(7) zuerst den schnellsten CVSup-Server herausfinden. fastest\_cvsup findet man im FreeBSD Portbaum unter sysutils/fastest\_cvsup.

Um zum Beispiel den schnellsten der schweizer- und deutschen CVSup-Server zu finden, benutzt man fastest\_cvsup wie folgt:

## 1. FreeBSD

```
# fastest_cvsup -c ch,de
>> Querying servers in countries: de ch
--> Connecting to cvsup.de.freebsd.org [212.19.57.134]...
    - server replied: OK 17 0 SNAP_16_1h CVSup server ready
    - time taken: 25.36 ms
--> Connecting to cvsup.ch.freebsd.org [217.8.192.70]...
    - server replied: OK 17 0 SNAP_16_1h CVSup server ready
    - time taken: 24.36 ms
--> Connecting to cvsup2.de.freebsd.org [213.238.33.226]...
    - server replied: OK 17 0 SNAP_16_1h CVSup server ready
    - time taken: 26.15 ms
--> Connecting to cvsup3.de.freebsd.org [131.159.72.30]...
    - server replied: OK 17 0 SNAP_16_1h CVSup server ready
    - time taken: 36.55 ms
--> Connecting to cvsup4.de.freebsd.org [212.19.57.134]...
    - server replied: OK 17 0 SNAP_16_1h CVSup server ready
    - time taken: 25.15 ms
--> Connecting to cvsup5.de.freebsd.org [195.30.6.180]...
    - server replied: OK 17 0 SNAP_16_1h CVSup server ready
    - time taken: 32.89 ms
--> Connecting to cvsup6.de.freebsd.org [62.206.253.13]...
    - server replied: OK 17 0 SNAP_16_1h CVSup server ready
    - time taken: 41.53 ms
--> Connecting to cvsup7.de.freebsd.org [194.162.162.209]...
    - server replied: OK 17 0 SNAP_16_1g CVSup server ready
    - time taken: 27.28 ms
--> Connecting to cvsup8.de.freebsd.org [212.118.165.142]...
    - server replied: OK 17 0 SNAP_16_1h CVSup server ready
    - time taken: 30.45 ms
>> Speed Daemons:
    - 1st: cvsup.ch.freebsd.org      24.36 ms
    - 2st: cvsup4.de.freebsd.org    25.15 ms
    - 3st: cvsup.de.freebsd.org     25.36 ms
```

Damit beim Ausführen von CVSup jedesmal der schnellste CVSup-Server gesucht wird, kann man zum Beispiel folgendes Hilfsskript verwenden:

```
#!/bin/sh
if SERVER='fastest_cvsup -Q -c ch,de'; then
    cvsup -g -L 2 -h $SERVER <Pfad zum supfile>
fi
```

Für csup verwendet man dies hier:

```
#!/bin/sh
if SERVER='fastest_cvsup -Q -c ch,de'; then
    csup -L 2 -h $SERVER <Pfad zum supfile>
fi
```

Mehr Informationen zu fastest\_cvsup findet man auf <http://fastest-cvsup.sourceforge.net/>. Alle Optionen von fastest\_cvsup kriegt man mit fastest\_cvsup -h.

### 1.28. Neue Make-Targets säubern das System nach einem Update

[ FreeBSD ] by Beat @ 17.03.2006 14:03 CET

Auf RELENG\_6, aus dem voraussichtlich in den nächsten Tagen auch der Zweig für 6.1 abgespaltet wird, befinden sich seit dem 7. Januar drei neue Make-Targets in src/Makefile die aus CURRENT eingeflossen sind:

```
# check-old      Gibt eine Liste mit veralteten Dateien und Verzeichnissen des
                  Basissystemes zurück
# delete-old     Löscht veraltete Dateien und Verzeichnisse interaktiv
# delete-old-libs Löscht veraltete Bibliotheken interaktiv
```

Nach /usr/src/Makefile sieht ein Update des Kernels und des Basissystemes auf CURRENT bzw. RELENG\_6 und für das kommende 6.1-RELEASE nun wie folgt aus:

## 1. FreeBSD

```
# cd /usr/src
# make buildworld
# make buildkernel KERNCONF=<KERNEL KONFIGURATIONSDATEI>
# make installkernel KERNCONF=<KERNEL KONFIGURATIONSDATEI>
# reboot      <- Im Startmenü den single user Modus wählen bzw boot -s eingeben
# mergemaster -p
# make installworld
# make delete-old
# mergemaster
# reboot
# make delete-old-libs
```

Wichtig ist natürlich, vor jedem Update /usr/src/UPDATING gut durchzulesen!

Ein make delete-old-libs sollte jedoch nur gemacht werden, falls kein Port oder selber installierte Software eine der alten Bibliotheken benötigt, da sonst diese Programme nicht mehr ordnungsgemäss funktionieren.

### 1.29. Audio CD's von der Konsole aus bedienen

[ Befehl der Woche ] [ FreeBSD ] by Beat @ 18.03.2006 15:56 CET

Audio CD's lassen sich bequem von der Konsole aus mit cdcontrol(1) bedienen. cdcontrol befindet sich schon im FreeBSD Basissystem, muss also nicht über die Ports installiert werden.

Mit der -f Option kann das CD-Laufwerk angegeben werden. Befindet sich das Laufwerk unter /dev/cdrom, /dev/cd0 oder /dev/acd0 so kann die -f Option weggelassen werden:

```
# cdcontrol [-f /dev/<laufwerk>] <Kommando>
```

Um das CD-Laufwerk zu öffnen, kann folgender Befehl gebraucht werden:

```
# cdcontrol eject
```

Zeigt die Anzahl Lieder und weitere Informationen der CD an:

```
# cdcontrol info
```

Startet das Abspielen der CD. Optional kann auch noch angegeben werden, welches Lied abgespielt werden soll:

```
# cdcontrol play [<Liednummer>]
```

Spielt das nächste Lied ab:

```
# cdcontrol next
```

Spielt das vorherige Lied ab:

```
# cdcontrol prev
```

Pause:

```
# cdcontrol pause
```

Nach einer Pause weiterspielen:

```
# cdcontrol resume
```

Zeigt das aktuell abgespielte Lied an und welche Lautstärke eingestellt ist:

```
# cdcontrol status
```

Beendet das Abspielen:

```
# cdcontrol stop
```

Weitere Optionen von cdcontrol findet man in der Manpage cdcontrol(1).

Benutzt man cdcontrol als normaler Benutzer und folgende Fehlermeldung tritt auf, so muss die Berechtigung des CD-Laufwerkes angepasst werden:

```
# cdcontrol play
cdcontrol: /dev/acd0: Permission denied
```

## 1. FreeBSD

Dies kann zum Beispiel durch einen Eintrag in der `/etc/devfs.rules` gemacht werden. Folgender Eintrag den bestehenden Regeln anfügen, wobei das Laufwerk (hier `acd0`) und die Benutzergruppe (hier `users`) individuell angepasst werden muss:

```
add path 'acd0' mode 0660 group users
```

### 1.30. Konsolenauflösung auf FreeBSD ändern

[ FreeBSD ] by Beat @ 20.03.2006 10:54 CET

Unter FreeBSD lässt sich die Auflösung der Konsole mit `vidcontrol(1)` anpassen. Seit FreeBSD 6.0 lassen sich nun auch Auflösungen höher als 800x600 einstellen. Dazu müssen folgende Optionen im Kernel vorhanden sein:

```
options VESA
options VGA_WIDTH90
options SC_PIXEL_MODE
```

Danach lassen sich die unterstützten Modi anzeigen:

```
# vidcontrol -i mode
```

Möchte man nun eine Auflösung von 1280 x 1024 auf der Konsole haben, muss der passende Modus gefunden werden:

```
# vidcontrol -i mode | grep 1280x1024
263 (0x107) 0x0000000f G 1280x1024x8 1 8x16 0xa0000 64k 64k 0xe0000000 65472k
281 (0x119) 0x0000000f G 1280x1024x15 1 8x16 0xa0000 64k 64k 0xe0000000 65472k
282 (0x11a) 0x0000000f G 1280x1024x16 1 8x16 0xa0000 64k 64k 0xe0000000 65472k
283 (0x11b) 0x0000000f G 1280x1024x24 1 8x16 0xa0000 64k 64k 0xe0000000 65472k
292 (0x124) 0x0000000f G 1280x1024x32 1 8x16 0xa0000 64k 64k 0xe0000000 65472k
```

Danach kann der gewünschte Modus eingestellt werden:

```
# vidcontrol MODE_<Modusnummer>
```

Um zum Beispiel auf einem IBM T40p oder T42 eine Auflösung von 1280 x 1024 x 24 einzustellen, wäre dies:

```
# vidcontrol MODE_283
```

Die Auflösung kann auch beim Starten automatisch eingestellt werden, indem man folgende Zeile in der `/etc/rc.conf` einträgt:

```
allscreens_flags="MODE_283"
```

Danke an Alain, der das Ganze unter FreeBSD 6.0 auf einem T40p ausprobiert hat.

Gefunden im BSDForen.de - Wiki, ursprünglich geschrieben von Sandro und veröffentlicht unter folgenden Lizenzbestimmungen.

### 1.31. Hostnamen einer Jail ändern

[ FreeBSD ] by Beat @ 26.03.2006 14:13 CEST

Muss der Hostname einer Jail geändert werden, so schlägt dies normalerweise mit folgender Fehlermeldung fehl:

```
jail# hostname <hostname>.<domain>.<tld>
hostname: sethostname: Operation not permitted
```

Dies liegt daran, dass im Hostsystem die `security.jail.set_hostname_allowed` Systemvariable standardmässig auf 0 ist:

```
host# sysctl -a | grep set_hostname
security.jail.set_hostname_allowed: 0
```

Möchte man nun den Hostnamen einer Jail ändern, muss zuerst im Hostsystem die Systemvariable geändert werden:

## 1. FreeBSD

```
host# sysctl security.jail.set_hostname_allowed=1
security.jail.set_hostname_allowed: 0 -> 1
```

Danach den Hostnamen in der Jail mit hostname(1) ändern:

```
jail# hostname <hostname>.<domain>.<tld>
```

Danach die Systemvariable im Hostsystem wieder zurücksetzen:

```
host# sysctl security.jail.set_hostname_allowed=0
security.jail.set_hostname_allowed: 1 -> 0
```

### 1.32. Quellcode mit csup anstatt CVSup aktualisieren

[ Befehl der Woche ] [ FreeBSD ] by Beat @ 01.04.2006 06:49 CEST

Seit kurzem gibt es eine unter der BSD-Lizenz stehende Alternative zu CVSup mit der man zum Beispiel den Portbaum oder den System-Quellcode aktualisieren kann: csup

Da csup in C geschrieben ist und deshalb keine zusätzlichen Abhängigkeiten in den Ports hat, gleich CVSup und das zum Bauen von CVSup benötigte ezm3 deinstallieren und csup installieren:

```
# pkg_deinstall -d -v cvsup-without-gui-16.1h_2
# pkg_deinstall -d -v ezm3-1.2
# cd /usr/ports/net/csup/ && make install clean
```

Die Sup-Dateien von CVSup können auch mit csup verwendet werden. Das Aktualisieren ist ähnlich wie bei CVSup:

```
# csup -L 2 <Pfad zur Sup-Datei>
```

Mehr Informationen und alle Optionen von csup findet man in der Manpage csup(1).

### 1.33. make targets für den FreeBSD Portbaum

[ Befehl der Woche ] [ FreeBSD ] by Beat @ 20.05.2006 11:38 CEST

Für den FreeBSD Portbaum bestehen verschiedene make targets. Diese können im Verzeichnis eines Ports angewendet werden, zum Beispiel /usr/ports/www/fluxcms, oder wo sich der Portbaum auch immer befindet.

Die meist gebrauchten targets sind vermutlich make install und make deinstall, wobei bei einem make deinstall Vorsicht geboten ist, da es nicht auf Abhängigkeiten achtet. Es gibt aber noch viele andere targets, die zum Teil nicht so oft gebraucht werden, jedoch manchmal sehr nützlich sind. Möchte man nur die Quelldateien des Ports herunterladen, kann man dies wie folgt tun:

```
# make fetch
```

Will man die Quelldateien des Ports und die seiner Abhängigkeiten herunterladen, nimmt man folgendes target:

```
# make fetch-recursive
```

Zum Herausfinden, welche Dateien heruntergeladen werden würden, sie jedoch noch nicht herunterladen will, kann man eine der folgenden targets nehmen, je nachdem ob man die Dateien der Abhängigkeiten auch anzeigen lassen will:

```
# make fetch-list
# make fetch-recursive-list
```

Die Abhängigkeiten eines Ports lassen sich mit folgendem target herausfinden:

```
# make all-depends-list
```

Will man nur die Abhängigkeiten sehen, die der Port zum Laufen braucht, das heisst ohne die Abhängigkeiten die es braucht um den Port zu bauen, so kann man dies wie folgt tun:

```
# make run-depends-list
```

## 1. FreeBSD

Einige Ports lassen sich mit einem `dialog(1)` konfigurieren. Der Dialog lässt sich durch folgende targets für nur den einen Port oder auch für alle Abhängigkeiten aufrufen:

```
# make config
# make config-recursive
```

Um die aktuelle Konfiguration anzusehen, kann man folgende targets benutzen:

```
# make showconfig
# make showconfig-recursive
```

Um die Konfigurationen zu löschen, stehen diese targets zur Verfügung:

```
# make rmconfig
# make rmconfig-recursive
```

Ein Portverzeichnis lässt sich wie folgt von temporären Daten säubern, die zum Bauen eines Ports benötigt werden:

```
# make clean
```

Will man auch noch die Quelldateien löschen, kann man das wie folgt tun:

```
# make distclean
```

Zum Säubern des Portverzeichnisses sollte man sich auch `portsclean(1)` anschauen.

Sucht man nach einem Port, kann man entweder `portsearch` oder `make search` in `/usr/ports` benutzen.

```
# cd /usr/ports/ && make search
The search target requires a keyword parameter or name parameter,
e.g.: "make search key=somekeyword"
or    "make search name=somekeyword"
```

Für mehr Informationen zum Portbaum empfiehlt sich das Kapitel 4 des FreeBSD Handbuchs. Alle make targets sind in `ports(7)` beschrieben.

### 1.34. Portspezifische Makevariablen setzen

[ FreeBSD ] [ Befehl der Woche ] by Beat @ 10.06.2006 10:10 CEST

Mit `portconf` ist es möglich, jedem Port gewisse Makevariablen zuzuweisen. `portconf` kann wie folgt installiert werden:

```
# cd /usr/ports/sysutils/portconf && make install
```

Dabei wird die `/etc/make.conf` um einige Zeilen erweitert, welche allerdings nicht editiert werden sollten. Danach können in `/usr/local/etc/ports.conf` die gewünschten Variablen gesetzt werden. Zum Beispiel:

```
editors/openoffice*: WITH_CCACHE|LOCALIZED_LANG=de
java/eclipse: WITH_MOZILLA=firefox
www/apache2*: WITH_SSL_MODULES=yes | WITH_DAV_MODULES=yes
www/firefox-i18n: WITHOUT_SWITCHER | FIREFOX_I18N=de en
x11/fakeport: CONFIGURE_ARGS=--with-modules="aaa bbb ccc"
```

Mehrere Optionen können mit Pipe getrennt werden. Die Konfigurationen, die mit `portconf` definiert werden, werden von `portmaster`, `portupgrade`, `portmanager` und `make install` berücksichtigt. Da `portconf` erst seit dem 8. Juni 2006 in den Ports ist, muss evtl. vor dem Installieren der Portbaum auf den aktuellen Stand gebracht werden.

### 1.35. Mehrere WEP-verschlüsselte WLAN's mit DHCP nutzen

[ FreeBSD ] by Beat @ 11.06.2006 14:05 CEST

Arbeitet man mit einem Laptop nur in einem WEP-verschlüsselten WLAN, so kann man dieses einfach mit einer `start_if` Datei konfigurieren. Arbeitet man hingegen in mehreren WLAN's, so ist diese Variante nicht geeignet. Dies kann jedoch mit `wpa_supplicant(8)` gelöst werden. Zuerst muss die Datei `/etc/wpa_supplicant.conf` angelegt werden. Darin werden die verschiedenen WLAN's konfiguriert:

## 1. FreeBSD

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=wheel

network={
    ssid="<SSID Netz 1>"
    scan_ssid=1
    key_mgmt=NONE
    wep_tx_keyidx=0
    wep_key0=<WEP Schlüssel Netz 1>
}

network={
    ssid="<SSID Netz 2>"
    scan_ssid=1
    key_mgmt=NONE
    wep_tx_keyidx=0
    wep_key0=<WEP Schlüssel Netz 2>
}
```

Hier können nun beliebig viele WLAN Netze eingetragen werden und das jeweils verfügbare Netz wird dann genommen. Mit wpa\_supplicant lassen sich auch WPA Netze konfigurieren. Alle Konfigurationsmöglichkeiten der wpa\_supplicant.conf sind in der Manpage wpa\_supplicant.conf(5) ausführlich beschrieben.

Danach muss die WLAN Karte nur noch in der /etc/rc.conf eingetragen werden:

```
ifconfig_WLAN_Karte="DHCP wpa"
```

Um wpa\_supplicant nutzen zu können, muss FreeBSD 6.0 oder neuer installiert sein.

### 1.36. Portabhängigkeiten graphisch darstellen

[ Befehl der Woche ] [ FreeBSD ] by Beat @ 17.06.2006 12:06 CEST

Die Abhängigkeiten eines Ports lassen sich mit einem make all-depends-list im Portverzeichnis anzeigen:

```
# cd /usr/ports/devel/porttools && make all-depends-list
/usr/ports/devel/portlint
/usr/ports/textproc/cdiff
/usr/ports/security/sudo
/usr/ports/devel/newfile
/usr/ports/lang/perl5.8
/usr/ports/lang/ruby18
/usr/ports/security/openssl
```

Möchte man jetzt wissen, wie diese Ports voneinander abhängen, so kann man dies mit pkg\_tree(7) anzeigen lassen.

```
# pkg_tree [-v] <Portname>
```

Benutzt man pkg\_tree ohne Optionen, werden nur die Abhängigkeiten eines Ports angezeigt. Zum Beispiel:

```
# pkg_tree porttools
porttools-0.77
|__ sudo-1.6.8.12_1
|__ ruby-1.8.4_4,1
|__ perl-5.8.8
|__ portlint-2.9.0
|__ openssl-0.9.8a
|__ newfile-1.0.14_1
|__ cdiff-1.4_1
```

Benutzt man die -v Option, werden auch die Abhängigkeiten der Abhängigkeiten dargestellt:

```
# pkg_tree -v porttools
porttools-0.77
|__ sudo-1.6.8.12_1
```



## 1. FreeBSD

```
| \_ ruby-1.8.4_4,1
| \_ perl-5.8.8
| \_ portlint-2.9.0
|   \_ perl-5.8.8
| \_ openssl-0.9.8a
| \_ newfile-1.0.14_1
|   | \_ openssl-0.9.8a
|   |   \_ ruby-1.8.4_4,1
|   \_ cdiff-1.4_1
|     \_ perl-5.8.8
```

pkg\_tree kann einfach über den Portbaum installiert werden:

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

### 1.37. Flashplugin 7 für Firefox auf FreeBSD 6.1

[ FreeBSD ] by Beat @ 21.06.2006 10:57 CEST

Da flashplugin6 wieder aus den Ports entfernt wurde, muss man das flashplugin7 installieren, wenn man nicht auf Flash verzichten möchte.

Folgendermassen hat die Installation auf FreeBSD 6.1 funktioniert:

```
# cd /usr/ports/www/firefox && make install clean
# cd /usr/ports/www/linuxpluginwrapper && make install clean
# cd /usr/ports/www/linux-flashplugin7 && make install clean
# cd /usr/src
# fetch http://people.FreeBSD.org/~nork/rtld_dlsym_hack.diff
# patch < rtld_dlsym_hack.diff
# cd libexec/rtld-elf
# make clean && make obj && make depend && make && make install
# cp /usr/local/share/examples/linuxpluginwrapper/libmap.conf-FreeBSD6
  /etc/libmap.conf
# ln -s /usr/X11R6/lib/linux-flashplugin7/libflashplayer.so
  /usr/X11R6/lib/browser_plugins/libflashplayer.so
```

Quellen:

<http://lists.freebsd.org/pipermail/freebsd-questions/2006-May/122903.html>

<http://lists.freebsd.org/pipermail/freebsd-questions/2006-May/122941.html>

### 1.38. Mit Xinerama zwei Bildschirme an einem Rechner verwenden

[ FreeBSD ] by Beat @ 11.07.2006 16:47 CEST

Mit Hilfe von Xinerama kann man mit einem Windowmanager mehrere Graphikkarten und Bildschirme benutzen. Fluxbox besitzt eine Xinerama Unterstützung. Auf FreeBSD kann man diese beim Installieren wie folgt aktivieren:

```
# cd /usr/ports/x11-wm/fluxbox-devel && make -DWITH_XINERAMA install
```

Danach als root eine xorg.conf erstellen lassen:

```
# X -configure
```

Wurde alles richtig erkannt, muss die erstellte xorg.conf nun noch angepasst werden. In der Section SServerLayout ist folgende Option einzutragen:

```
Option "Xinerama" "true"
```

Das TastaturLayout muss auch noch richtig gesetzt und die gewünschten Auflösungen konfiguriert werden.

Danach kann Fluxbox schon mit zum Beispiel zwei Monitoren verwendet werden.

Folgende Komponenten wurden verwendet:

- Graphikkarten: ATI Radeon 9000 und Matrox Mystique
- Betriebssystem: FreeBSD 6.1-STABLE

## 1. FreeBSD

- X-Window System: X.Org 6.9.0
- Windowmanager: fluxbox 1.0-RC2
- X.Org Konfiguration: xorg.conf
- Hintergrundbild: <http://www.bsdnexus.com/wallpapers.htm>

Ergebnis:



Abbildung 3: fluxbox mit Xinerama auf zwei Monitoren

### 1.39. Eingeloggte Benutzer beobachten

[ Befehl der Woche ] [ FreeBSD ] by Beat @ 15.07.2006 02:41 CEST

Für Schulungen oder wenn man einem eingeloggten Benutzer helfen möchte, ist es manchmal von Vorteil, wenn man die Terminalausgabe eines Benutzers sieht. Dafür gibt es `watch(8)`. Ist das device `snp` nicht im Kernel vorhanden, so wird automatisch beim ersten Aufruf von `watch` das `snp.ko`-Kernelmodul geladen. Natürlich kann `watch` nur von `root` verwendet werden.

Zuerst muss man herausfinden, mit welchem Terminal (`tty`) der zu überwachende Benutzer arbeitet. Dies lässt sich mit `w(1)` herausfinden:

```
# w
 5:22pm up 4:50, 4 users, load averages: 0,08 0,13 0,11
USER      TTY      FROM          LOGIN@   IDLE WHAT
guest     p1        :0.0          12:45pm   1 more
beat      p2        :0.0          1:18pm   1 vi Makefile
beat      p3        :0.0          4:46pm  36 csh
beat      p4        :0.0          5:13pm   8 w
```

Möchte man jetzt das Terminal (`ttyp1`) des Benutzers `guest` beobachten, so kann man das als `root` wie folgt tun:

## 1. FreeBSD

```
# watch ttyp1
```

Nun sieht man alle Ein- und Ausgaben des Benutzers guest auf ttyp1. watch beendet man mit Ctrl + G. Mit Ctrl + W löscht man die Ausgabe auf dem Bildschirm und mit Ctrl + X kann man auf ein anderes Terminal wechseln.

Möchte man zusätzlich auch Eingaben auf dem Terminal des Benutzers machen können, so kann man watch mit der -W Option aufrufen:

```
# watch -W ttyp1
```

### 1.40. Verzeichnis eines Rechners mittels SSH mounten

[ FreeBSD ] [ Befehl der Woche ] by Beat @ 29.07.2006 07:43 CEST

Mit Hilfe von FUSEFS-SSHFS lässt sich ein Verzeichnis eines entfernten Rechners auf einem lokalen Rechner mounten, wenn man einen SSH-Zugang zu diesem Rechner besitzt.

FUSEFS-SSHFS lässt sich einfach über die Ports installieren:

```
# cd /usr/ports/sysutils/fusefs-sshfs && make install
```

Da das FUSE-Kernelmodul gebraucht wird, muss der Kernelmodul-Suchpfad noch angepasst werden. Einfach folgende Zeile in der /etc/sysctl.conf anfügen:

```
kern.module_path="/boot/kernel;/boot/modules;/usr/local/modules"
```

Jetzt muss noch folgender Eintrag in der /etc/rc.conf gemacht werden:

```
fusefs_enable="YES"
```

Möchte man das SSHFS ohne einen Neustart gleich verwenden, so muss FUSE von Hand gestartet werden:

```
# sysctl kern.module_path="/boot/kernel;/boot/modules;/usr/local/modules"
# /usr/local/etc/rc.d/fusefs start
```

Danach kann ein Verzeichnis gemountet werden:

```
# sshfs -o idmap=user <Benutzername>@<Rechner>:<Verzeichnis> <Mountpunkt>
```

Alle Optionen von sshfs erhält man mit der -h Option:

```
# sshfs -h
```

**Performancetest:** Als Server diente ein FreeBSD 5.3-RELEASE, als Client ein FreeBSD 6.1-RELEASE. Der Client war über ein 11MB WEP-Verschlüsseltes WLAN mit einem Access-Point verbunden, welcher über ein 100MB Ethernet mit dem Server verbunden war. Als Testdatei wurde eine ReactOS 3.0-RC1 Live-CD verwendet.

```
# du -ch ReactOS-LiveCD.iso
63M    ReactOS-LiveCD.iso
63M    total
```

Getestet wurden scp, SMB, SSHFS, und NFS. Die Übertragungszeit wurde mit /usr/bin/time -h gemessen. Jede Übertragung wurde 3 mal durchgeführt.

#### scp

```
# /usr/bin/time -h scp ReactOS-LiveCD.iso 192.168.1.10:/home/beat
1m43,31s real          1,75s user          0,57s sys
1m42,76s real          1,73s user          0,59s sys
1m42,99s real          1,76s user          0,56s sys
```

#### SMB mit Samba 3

```
# mount_smbfs -I 192.168.1.10 //beat@zeus/beat /mnt/zeus
# /usr/bin/time -h cp ReactOS-LiveCD.iso /mnt/zeus/
1m45,54s real          0,00s user          0,13s sys
1m45,09s real          0,00s user          0,13s sys
1m46,15s real          0,00s user          0,12s sys
```

## 1. FreeBSD

### SSHFS

```
# sshfs -o idmap=user beat@192.168.1.10:/home/beat /mnt/zeus/  
# /usr/bin/time -h cp ReactOS-LiveCD.iso /mnt/zeus/  
1m42,10s real          0,00s user          0,32s sys  
1m41,99s real          0,00s user          0,32s sys  
1m42,60s real          0,00s user          0,32s sys
```

### NFS

```
# mount_nfs 192.168.1.10:/usr /mnt/zeus/  
# /usr/bin/time -h cp ReactOS-LiveCD.iso /mnt/zeus/home/beat/  
1m27,00s real          0,00s user          0,16s sys  
1m27,59s real          0,01s user          0,15s sys  
1m27,74s real          0,00s user          0,16s sys
```

Wie man hier sieht, ist das SSHFS in der Übertragungsdauer mit scp vergleichbar, SMB auch nur ein paar Sekunden langsamer. Deutlich schnellere Übertragungen erreichte man hier mit NFS.

### 1.41. Benutzern temporär das Anmelden verbieten

[ Befehl der Woche ] [ FreeBSD ] by Beat @ 05.08.2006 07:21 CEST

Möchte man, zum Beispiel wegen Wartungsarbeiten, allen Benutzern das Anmelden verbieten, so kann man einfach die Datei `/var/run/nologin` erstellen und schon können sich alle Benutzer bis auf root nicht mehr anmelden. Enthält die Datei auch noch einen Text, so wird dieser angezeigt, wenn ein Benutzer versucht sich anzumelden:

```
# echo Wegen Wartungsarbeiten ist das Anmelden bis 22 Uhr nicht gestattet >  
    /var/run/nologin  
  
# ssh localhost  
Wegen Wartungsarbeiten ist das Anmelden bis 22 Uhr nicht gestattet
```

Alle bereits angemeldeten Benutzer können jedoch weiterhin auf dem System arbeiten. Möchte man diese Benutzer vom System trennen, so kann man mit `shutdown -k` alle angemeldeten Benutzer bis auf root ausloggen.

Um das Anmelden wieder zu gestatten, muss einfach `/var/run/nologin` wieder entfernt werden:

```
# rm /var/run/nologin
```

Achtung: Hat man auf dem System das direkte Anmelden als root abgeschaltet, so ist `nologin(5)` mit Vorsicht zu geniessen, da man sich selber aus einem System aussperren kann!

### 1.42. FreeBSD Service Jails mit unionfs erstellen

[ FreeBSD ] by Beat @ 10.08.2006 12:51 CEST

Mit Hilfe von Service Jails lassen sich gemeinsame Elemente einer Jail mit mehreren Jails teilen. Damit reduziert sich der Administrationsaufwand, da man zum Beispiel Sicherheitspatches des Basissystemes nur noch in der gemeinsamen Basis einspielen muss und nicht mehr in jeder Jail. Inspiriert durch folgende Seite (<http://simon.nitro.dk/service-jails.html>) habe ich nun Service Jails mit Hilfe von unionfs erstellt.

Mit unionfs lässt sich ein Dateisystem über ein bestehendes mounten. Wird eine Datei aus dem unterliegenden System nicht durch eine gleichnamige Datei aus dem oberliegenden Dateisystem überdeckt, so ist diese auch im oberliegenden System sichtbar. Änderungen fließen allerdings nur in das obere System ein.

Da die aktuelle unionfs-Implementation von FreeBSD recht fehlerhaft sein soll, verwenden wir gleich die neue unionfs-Implementation. Damit diese verwendet werden kann, müssen die aktuellen FreeBSD-Quelldateien der verwendeten FreeBSD Version in `/usr/src` vorhanden sein. Danach kann der Patch heruntergeladen, kompiliert und die neue Implementetation installiert werden:

## 1. FreeBSD

```
# cd /usr/src
# fetch http://people.freebsd.org/~daichi/unionfs/unionfs6-p16.diff
# patch -p3 < unionfs6-p16.diff
# cp /usr/src/sys/fs/unionfs/union.h /usr/include/fs/unionfs/
# cd /usr/src/sbin/mount_unionfs/
# make obj depend && make && make install
# make clean
# cd /usr/src
# make buildkernel KERNCONF=<KERNEL-KONFIGURATION>
# make installkernel KERNCONF=<KERNEL-KONFIGURATION>
# shutdown -r now
```

Dieses Vorgehen lädt den Patch 16 für FreeBSD 6 herunter. Man sollte sich vorher allerdings vergewissern, dass man das aktuelle Patchset für die zu verwendende FreeBSD-Version herunterlädt. Danach kann die gemeinsame Basis der Jails erstellt werden. Diese wird in einer 1GB grossen Datei erstellt, welche wiederum als Memory Disk gemountet wird. Der Portbaum wird aus dem Hostsystem in die Jailbasis verschoben und mit einem Symlink dem Hostsystem wieder zur Verfügung gestellt.

```
# mkdir /usr/jails/
# truncate -s 1G /usr/jails/base
# mdconfig -a -t vnode -f /usr/jails/base -u 0
# bsdlabel -w md0 auto
# newfs md0a
# mkdir -p /mnt/jails
# mkdir -p /mnt/jails/base
# mount /dev/md0a /mnt/jails/base/
# cd /usr/src/ && make installworld DESTDIR=/mnt/jails/base
# cd /usr/src/etc && make distribution DESTDIR=/mnt/jails/base

# mv /usr/ports /mnt/jails/base/usr/ports
# ln -s /mnt/jails/base/usr/ports /usr/ports

# /sbin/mount_unionfs -c transparent -b /usr/ports /mnt/jails/base/usr/ports
# touch /mnt/jails/base/etc/fstab
```

Nun kann die gemeinsame Basis nach eigenen Wünschen konfiguriert werden. Danach kann eine Jail folgendermassen erzeugt werden:

```
# truncate -s <Grösse> /usr/jails/<Name>
# mdconfig -a -t vnode -f /usr/jails/<Name> -u <Nr>
# bsdlabel -w md<Nr> auto
# newfs md<Nr>a
# mkdir -p /mnt/jails/<Name>
# mount /dev/md<Nr>a /mnt/jails/<Name>

# mount_unionfs -c transparent -b /mnt/jails/base /mnt/jails/<Name>

# mount_devfs devfs /mnt/jails/<Name>/dev
# cd /mnt/jails/<Name>
# ln -sf /dev/null kernel
```

<Grösse> ist durch die Grösse der Datei zu ersetzen, in der man die Jail erzeugen will. Für zum Beispiel 2 GB gibt man hier 2G an. <Name> ist natürlich durch den Jailnamen zu ersetzen. Bei der ersten Jail ersetzt man <Nr> durch 1, bei der zweiten durch 2 usw.

Nun wird die Jail gestartet und kann konfiguriert werden:

```
# ifconfig <Interface> alias <IP-Adresse>/32
# jail /mnt/jails/<Name> <Name> <IP> /bin/sh
```

So können nun mehrere Jails erzeugt werden, die das gleiche Basissystem verwenden. Da diese Variante für die Jails Dateien verwendet, die als RAM-Disk gemountet werden, müssen diese auch bei einem Neustart zuerst geladen werden, damit man nachher die Jails ganz normal über die rc.conf starten kann. Dazu wird unter /etc/rc.d/jails eine Datei mit folgendem Inhalt erstellt:

## 1. FreeBSD

```
#!/bin/sh
#
# -----
# "THE BEER-WARE LICENSE" (Revision 42):
# <beat@chruetertee.ch> wrote this file. As long as you retain this notice you
# can do whatever you want with this stuff. If we meet some day, and you think
# this stuff is worth it, you can buy me a beer in return Beat Gätzi
# -----
#
# PROVIDE: jails
# REQUIRE: root mountcritlocal
# BEFORE: jail
. /etc/rc.subr

name="jails"
rcvar='set_rcvar'
start_cmd="jails_start"
stop_cmd="jails_stop"

jails_start()
{
    echo 'Setting up jails environment'
    /sbin/mdconfig -a -t vnode -f /usr/jails/base -u 0
    /sbin/mount /dev/md0a /mnt/jails/base/
    j=1
    for i in ${jails_list}
    do
        /sbin/mdconfig -a -t vnode -f /usr/jails/${i} -u ${j}
        /sbin/mount /dev/md${j}a /mnt/jails/${i}
        /sbin/mount_unionfs -c transparent -b /mnt/jails/base
            /mnt/jails/${i}
        j='expr $j + 1'
    done
}

jails_stop()
{
    j=1
    for i in ${jails_list}
    do
        /sbin/umount /mnt/jails/${i}
        /sbin/umount /mnt/jails/${i}
        /sbin/mdconfig -d -u ${j}
        j='expr $j + 1'
    done
    /sbin/umount /mnt/jails/base/usr/ports/
    /sbin/umount /mnt/jails/base/
    /sbin/mdconfig -d -u 0
}

load_rc_config $name
run_rc_command "$1"
```

Nun müssen die verschiedenen Jails noch in der `jails_list` Variable in der `/etc/rc.conf` eingetragen werden. Wichtig ist, dass die Namen mit den Namen im Pfad der Jails übereinstimmen. Beispiel: Zwei Jails sind unter `/mnt/jails/test1.chruetertee.ch` und `/mnt/jails/test2.chruetertee.ch` installiert, so muss das Startskript aktiviert und die `jails_list` Variable folgendermassen gesetzt werden:

```
jails_enable="YES"
jails_list="test1.chruetertee.ch test2.chruetertee.ch"
```

Die Jails können nun ganz normal in der `rc.conf` konfiguriert werden und stehen bei einem Neustart automatisch zur Verfügung.

```
jail_enable="YES"
jail_list="test1 test2"
jail_test1_rootdir="/mnt/jails/test1.chruetertee.ch"
```

## 2. OpenBSD

```
jail_test1_hostname="test1.chruetertee.ch"
jail_test1_ip="<IP>"
jail_test1_exec="/bin/sh /etc/rc"
jail_test1_devfs_enable="YES"
jail_test2_rootdir="/mnt/jails/test2.chruetertee.ch"
jail_test2_hostname="test2.chruetertee.ch"
jail_test2_ip="<IP>"
jail_test2_exec="/bin/sh /etc/rc"
jail_test2_devfs_enable="YES"
```

Diese Service Jails wurden mit FreeBSD 6.1 getestet und funktionieren bis jetzt hier auch. Allerdings ist diese Variante auf produktiven Systemen mit Vorsicht zu geniessen, da sowohl die neue unionfs-Implementierung als auch diese Variante noch nicht ausgiebig getestet worden sind!

Arbeitet man mit FreeBSD 6.1-RELEASE und mit den Jails sollte man unbedingt folgenden Hinweis beachten, da sich in FreeBSD 6.1-RELEASE ein Fehler in der `/etc/rc.d/jail` befindet: [ftp://ftp.freebsd.org/pub/FreeBSD/ERRATA/notices/FreeBSD-EN-06:01.jail.asc](http://ftp.freebsd.org/pub/FreeBSD/ERRATA/notices/FreeBSD-EN-06:01.jail.asc)

## 2. OpenBSD

### 2.1. OpenBSD von 3.7 auf 3.8 aktualisieren

[ OpenBSD ] by Beat @ 30.11.2005 20:03 CET

Vor dem Aktualisieren unbedingt folgende sehr gute Anleitung lesen: <http://www.openbsd.org/faq/upgrade38.html>

Ich habe hier die wichtigsten Schritte zusammengefasst, ein Upgrade sollte aber auf jeden Fall nach oben genannter Anleitung durchgeführt werden!

Vor dem Aktualisieren unbedingt ein Backup aller Daten durchführen!

Die benötigten Quelldateien von der CD kopieren oder von einem Mirror herunterladen:

```
# cd /root/
# ftp ftp://mirror.switch.ch/pub/OpenBSD/3.8/i386/bsd
# ftp ftp://mirror.switch.ch/pub/OpenBSD/3.8/i386/bsd.rd
# ftp ftp://mirror.switch.ch/pub/OpenBSD/3.8/i386/base38.tgz
# ftp ftp://mirror.switch.ch/pub/OpenBSD/3.8/i386/etc38.tgz
# ftp ftp://mirror.switch.ch/pub/OpenBSD/3.8/i386/comp38.tgz
# ftp ftp://mirror.switch.ch/pub/OpenBSD/3.8/i386/man38.tgz
# ftp ftp://mirror.switch.ch/pub/OpenBSD/3.8/i386/misc38.tgz
```

Danach eine Liste mit allen installierten Ports erstellen und diese deinstallieren:

```
# pkg_info > /root/pkg_lst
# pkg_delete -q /var/db/pkg/*
```

Alle zusätzlich aktivierten und nicht gebrauchten Daemonen in `/etc/rc.conf` und `/etc/rc.conf.local` deaktivieren und dann das Upgrade durchführen:

```
# cd /root
# rm -f /obsd ; ln /obsd /obsd && cp bsd /nbsd && mv /nbsd /bsd
# cp bsd.rd bsd.mp /
# cd /
# tar xzpf /root/base38.tgz "*/etc/firmware/*"
# reboot
# cd /
# tar xzpf /root/base38.tgz
# tar xzpf /root/comp38.tgz
# tar xzpf /root/man38.tgz
# tar xzpf /root/misc38.tgz
# cd /dev
# ./MAKEDEV all
# useradd -u86 -g=uid -c"HostAP Daemon" -d/var/empty -s/sbin/nologin _hostapd
# cd /tmp
# tar xzpf /root/etc38.tgz
# cd /tmp/etc
# cp hostapd.conf netstart pf.os rc services /etc
```

## 2. OpenBSD

```
# cp mtree/* /etc/mtree/
# cd /tmp/
# ftp ftp://mirror.switch.ch/pub/OpenBSD/3.8/src.tar.gz
# cd /usr/src/
# tar xzpf /tmp/src.tar.gz
# pkg_add
# ftp://mirror.switch.ch/pub/OpenBSD/3.8/packages/i386/mergemaster-1.46p0.tgz
# /usr/local/sbin/mergemaster
# mtree -qdef /etc/mtree/4.4BSD.dist -p / -u
# reboot
```

Danach alle Pakete wieder installieren, eine Liste der vorher installierten Pakete befindet sich noch unter `/root/pkg_lst`, und alle benötigten Dienste in `/etc/rc.conf` und `/etc/rc.conf.local` wieder aktivieren und gegebenenfalls den Rechner neu starten.

### 2.2. OpenOffice.org 2.0 auf OpenBSD

[ OpenBSD ] by Beat @ 12.12.2005 21:50 CET

Um OpenOffice.org 2.0 auf OpenBSD 3.8 laufen zu lassen, wird die Linuxemulation benötigt. Ist diese nicht bereits installiert, kann diese folgendermassen hinzugefügt werden:

```
# echo '/proc /proc procfs rw,linux 0 0' >> /etc/fstab
# mkdir /proc
# mount /proc
# /sbin/sysctl -w kern.emul.linux=1
# cd /usr/ports/emulators/redhat/base
# make install clean distclean
```

In `/etc/sysctl.conf` die Raute vor `kern.emul.linux=1` entfernen.

Danach die Linuxversion von OpenOffice.org herunterladen und diese installieren:

```
# cd /tmp
# ftp ftp://mirror.switch.ch/mirror/OpenOffice.localized/de/2.0.0/00o_2.0.0_LinuxIntel_install_de.tar.gz
# tar -xzf 00o_2.0.0_LinuxIntel_install_de.tar.gz
# mkdir /usr/local/emul/redhat/opt
# ln -s /usr/local/emul/redhat/opt /opt
# /emul/linux/bin/rpm --nodeps --ignoreos --ignorearch --force -ivh
# 000680_m3_native_packed-3_de.8968/RPMS/*.rpm
```

OpenOffice.org ist jetzt unter folgendem Pfad installiert: `/usr/local/emul/redhat/opt/openoffice.org2.0/program/soffice`

Quelle: [http://www.00f.net/php/show-article.php/openoffice\\_on\\_openbsd](http://www.00f.net/php/show-article.php/openoffice_on_openbsd)

### 2.3. OpenBSD Dateiset entfernen

[ OpenBSD ] by Beat @ 15.03.2006 15:37 CET

Möchte man auf OpenBSD ein installiertes Dateiset wieder entfernen, so kann folgender Einzeiler verwendet werden:

```
# tar -tzf <Dateiset> | sed -e 's/^\.//' | xargs rm -d
```

Da die Informationen, welche Dateien entfernt werden sollen, aus den Dateisets ausgelesen werden, müssen diese verfügbar sein. Möchte man zum Beispiel auf OpenBSD 3.8 die Spiele wieder entfernen und das `game38.tgz` Dateiset befindet sich im aktuellen Verzeichnis, so macht man dies wie folgt:

```
# tar -tzf game38.tgz | sed -e 's/^\.//' | xargs rm -d
```

### 2.4. Aktualität des OpenBSD-Portbaumes überprüfen

[ OpenBSD ] by Beat @ 03.05.2006 22:52 CEST

Auf OpenBSD kann man den Portbaum mit einem Perl-Skript auf Aktualität prüfen:



### 3. BSD Allgemein

```
# /usr/ports/infrastructure/build/out-of-date
```

Der Portbaum (oder Teile davon) kann danach mit CVS auf den aktuellen Stand gebracht werden. Einfach die CVSROOT Variable mit dem CVS-Server seines Vertrauens setzen und danach CVS mit dem passenden CVS-Tag ausführen:

```
# CVSROOT=anoncvs@anoncvs2.de.openbsd.org:/cvs
# cd /usr/port/<Verzeichnis das man aktualisieren möchte>
# cvs -d$CVSROOT up -P -rOPENBSD_3_9
```

## 2.5. OpenBSD Packages aktualisieren

[ OpenBSD ] by Beat @ 21.06.2006 21:48 CEST

Seit OpenBSD 3.9 lassen sich installierte Packages einfach mit pkg\_add(1) aktualisieren:

```
# pkg_add -ui -F update -F updatedepends
```

Dabei werden die installierten Packages mit denen verglichen, welche sich in einem Verzeichnis befinden, auf das die PKG\_PATH Umgebungsvariable zeigt.

Wird folgende Meldung angezeigt, so wurde PKG\_PATH nicht richtig gesetzt.

```
No packages available in the PKG_PATH
```

Mann kann PKG\_PATH auch gleich beim Aufruf von pkg\_add setzen:

```
# PKG_PATH=ftp://mirror.switch.ch/pub/OpenBSD/3.9/packages/i386 pkg_add -ui -F
update -F updatedepends
```

Im nachfolgenden Beispiel wurden alle Packages auf Aktualisierungen überprüft und dank der -i Option kann der Benutzer entscheiden, ob und auf welche Version ein Package aktualisiert wird. Hier wird dovecot von 1.0-beta3 auf 1.0-beta8 aktualisiert:

```
# PKG_PATH=ftp://mirror.switch.ch/pub/OpenBSD/3.9/packages/i386 pkg_add -ui -F
update -F updatedepends
Candidates for updating dovecot-1.0.beta3 - dovecot-1.0.beta3 dovecot-1.0.beta8
Ambiguous: dovecot-1.0.beta3 could be dovecot-1.0.beta3 dovecot-1.0.beta8
Choose one package
0:
1: dovecot-1.0.beta3
2: dovecot-1.0.beta8
Your choice: 2
Looking for updates: complete
Running the equivalent of pkg_add -r dovecot-1.0.beta8
dovecot-1.0.beta8 (extracting): complete
dovecot-1.0.beta3 (deleting): complete
dovecot-1.0.beta8 (installing): complete
```

## 3. BSD Allgemein

### 3.1. Dateien unter Free- und OpenBSD mit Flags schützen

[ FreeBSD ] [ OpenBSD ] by Beat @ 10.10.2005 22:24 CET

Auf Free- und OpenBSD gibt es die Möglichkeit, Dateien zusätzlich zu den User-, Group- und Othersattributen zu schützen. Diese Flags werden mit chflags(1) gesetzt und entfernt.

Setzen eines Flags:

```
# chflags <Flag> <Datei>
```

Um zum Beispiel das schg-Flag auf die testdatei zu setzen, verwende folgenden Befehl:

```
# chflags schg testdatei
```

Angezeigt werden die Flags mit ls -ol

### 3. BSD Allgemein

```
# ls -ol
-rw----- 1 beat users schg 618 Oct 5 15:20 testdatei
```

Entfernt werden die Flags indem man ein no vor die Flagbezeichnung setzt. Um zum Beispiel das schg Flag von der testdatei wieder zu entfernen, verwende:

```
# chflags noschg testdatei
```

Es gibt folgende Flags:

**schg** Niemand, auch nicht root, kann die Datei umbenennen, verändern oder löschen. Auch die Metadaten (Besitzer, Zuriffsrechte u.s.w.) können nicht verändert werden. Wird das Flag auf ein Verzeichnis gesetzt, so können keine neuen Dateien eingefügt werden. Kann nur von root gesetzt werden und nur von root und bei einem Securelevel < 1 wieder entfernt werden.

**uchg** Datei kann wie beim schg Flag nicht verändert werden, jedoch kann das uchg Flag von root und dem Dateibesitzer gesetzt und unabhängig vom Securelevel von beiden wieder entfernt werden.

**sappnd** Datei kann nicht verändert oder gelöscht werden, jedoch ist das Anfügen von Daten am Ende der Datei erlaubt. Kann nur von root gesetzt werden und nur von root und bei einem Securelevel < 1 wieder entfernt werden.

**uappnd** Es kann wie beim sappnd Flag nur angefügt werden, jedoch kann das uappnd Flag von root und dem Dateibesitzer gesetzt und unabhängig vom Securelevel von beiden wieder entfernt werden.

**sunlnk** Datei kann nicht gelöscht werden. Kann nur von root gesetzt werden und nur von root und bei einem Securelevel < 1 wieder entfernt werden. Dieses Flag existiert bei OpenBSD nicht.

**uunlnk** Datei kann wie beim sunlnk Flag nicht gelöscht werden, jedoch kann das uunlnk Flag von root und dem Dateibesitzer gesetzt und unabhängig vom Securelevel von beiden wieder entfernt werden. Dieses Flag existiert bei OpenBSD nicht.

**nodump** Datei mit gesetztem nodump Flag wird bei einem Backup mit dump(8) nicht gesichert. Dies ist die einzige Ausnahme indem der no Prefix das Flag nicht entfernt. Entfernt wird das Flag mit chflag dump <Datei> ! Kann von root oder dem Dateibesitzer gesetzt und entfernt werden.

**opaque** Wird mit unionfs verwendet. Unionfs ist aber noch nicht für den produktiven Einsatz gedacht. Zitat aus der Sektion BUGS aus der FreeBSD Manpage zu mount\_unionfs(8):

THIS FILE SYSTEM TYPE IS NOT YET FULLY SUPPORTED (READ: IT DOESN'T WORK) AND USING IT MAY, IN FACT, DESTROY DATA ON YOUR SYSTEM. USE AT YOUR OWN RISK. BEWARE OF DOG. SLIPPERY WHEN WET.

**arch** Dieses Flag hat keine Funktion und kann zur individuellen Kennzeichnung von Dateien verwendet werden.

Die Dateiflags sollten mit Bedacht eingesetzt werden, da sie zwar zusätzlichen Schutz bieten, jedoch das Administrieren unter Umständen erschweren können.

Weitere Informationen:

FreeBSD chflags(1)

OpenBSD chflags(1)

### 3.2. SMB Freigaben einfach mounten mit Sharity-Light

[ Befehl der Woche ] [ FreeBSD ] [ OpenBSD ] by Beat @ 03.12.2005 10:20 CET

Um eine SMB Freigabe einfach zu mounten gibt es Sharity Light:

```
/usr/local/sbin/shlight //<Server>/<Service> <Mountpunkt> -U <Benutzer> -C -u <UID>
-g <GID>
```

Danach muss noch das Passwort eingegeben werden und schon ist das Verzeichnis eingebunden.

-C bedeutet, dass das Passwort nicht in Grossbuchstaben konvertiert wird.

-u und -g weisen den eingebundenen Daten die User- und Group ID zu.

Um das eingebundene Verzeichnis wieder zu lösen, einfach folgenden Befehl benutzen:

```
/usr/local/sbin/ushlight <Mountpunkt>
```

Sharity Light befindet sich auf Open- und FreeBSD im Port net/sharity-light.

### 3. BSD Allgemein

#### 3.3. bsdtalk

[ BSD ] by Beat @ 18.02.2006 10:44 CET Auf <http://bsdtalk.blogspot.com/> findet man Interviews über BSD, zum Beispiel mit Kirk McKusick, Scott Long oder Dru Lavigne, als MP3s zum Herunterladen. Wirklich empfehlenswert!

#### 3.4. Zeit synchronisieren mit ntpd

[ Befehl der Woche ] [ FreeBSD ] [ OpenBSD ] by Beat @ 25.03.2006 12:57 CET

Um die lokale Zeit mit einem NTP-Server abzugleichen, kann ntpd(8) verwendet werden. Der ntpd kann auf FreeBSD einfach über die /etc/rc.conf gestartet werden.

```
ntpd_enable="YES"
ntpd_sync_on_start="YES"
```

ntpd\_sync\_on\_start bewirkt, dass die Zeit schon während des Aufstartens des Rechners das erste Mal abgeglichen wird. Die Konfiguration des ntpd wird in der /etc/ntp.conf festgelegt. Darin muss mindestens der NTP-Server bekanntgegeben werden. Dies kann zum Beispiel so aussehen:

```
server ntp.ethz.ch
```

Eine Übersicht über alle weiteren Konfigurationsmöglichkeiten in der /etc/ntp.conf findet man in der dazugehörigen Manpage ntp.conf(5). Alle Optionen des ntpd findet man in ntpd(8). Die Optionen können dem ntpd während des Startens mit folgendem Eintrag in der /etc/rc.conf übergeben werden:

```
ntpd_flags="<Optionen>"
```

Auf OpenBSD kann der ntpd durch folgenden Eintrag in der /etc/rc.conf.local gestartet werden:

```
ntpd_flags=""
```

Die Konfigurationsdatei für den ntpd befindet sich unter /etc/ntp.conf. Standardmässig werden zufällig zwei Server aus einem Serverpool ausgesucht. Alle Konfigurationsmöglichkeiten findet man in der ntpd.conf(5). Die Optionen des ntpd können in der ntpd(8) nachgelesen werden und ggf. mit dem ntpd\_flags Eintrag in der rc.conf.local übergeben werden.

#### 3.5. Kalender für die Kommandozeile

[ Befehl der Woche ] [ FreeBSD ] [ OpenBSD ] by Beat @ 15.04.2006 06:00 CEST

Passend zu Ostern dieser Befehl der Woche:

```
cal, ncal -- displays a calendar and the date of easter
```

cal(1) und ncal(1) befinden sich bereits im FreeBSD Basissystem. Auf OpenBSD ist nur cal(1) verfügbar. Wird cal ohne Optionen aufgerufen, so wird der aktuelle Kalendermonat angezeigt. ncal verwendet eine alternative Darstellung.

```
# cal
      April 2006
So Mo Di Mi Do Fr Sa
                        1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30

# ncal
      April 2006
Mo      3 10 17 24
Di      4 11 18 25
Mi      5 12 19 26
Do      6 13 20 27
Fr      7 14 21 28
Sa  1   8 15 22 29
So  2   9 16 23 30
```

### 3. BSD Allgemein

Für ncal lassen sich zusätzlich mit der -w Option die Wochennummer anzeigen.

Mit der -m Option lässt sich ein bestimmter Monat anschauen. Möchte man sich zum Beispiel den Oktober anschauen, so verwendet man cal wie folgt:

```
# cal -m 10
```

Mit der -y Option lässt sich ein bestimmtes Jahr anschauen. Die -e Option von ncal(1) zeigt das Datum von Ostern des aktuellen Jahres an:

```
# ncal -e
16 April 2006
```

Kombiniert man die -y und die -e Option, so zeigt ncal das Datum von Ostern des gewählten Jahres an.

Alle weiteren Optionen von cal und ncal findet man in der Manpage cal(1).

### 3.6. Befehle und Manpages finden mit apropos

[ Befehl der Woche ] [ FreeBSD ] [ OpenBSD ] by Beat @ 22.04.2006 11:29 CEST

Sucht man eine Manpage oder einen Befehl, weiss aber nicht genau wie er heisst, so kann einem apropos(1) weiterhelfen. apropos durchsucht die whatis Datenbank, welche wiederum aus den Namen und den Kurzbeschreibungen der Manpages besteht, und sucht darin nach dem Begriff. apropos kann ganz einfach verwendet werden:

```
# apropos <Begriff>
```

Möchte man zum Beispiel nach FreeBSD-Jails suchen, so verwendet man apropos wie folgt:

```
# apropos jail
jail(2), jail_attach(2) - imprison current process and future descendants
jail(8)                 - imprison process and its descendants
jexec(8)                - execute a command inside an existing jail
jls(8)                  - list active jails
prison_check(9)         - determine if two credentials belong to the same jail
```

Dadurch dass man für den Begriff auch Regular Expressions verwenden kann, kann man zum Beispiel wie folgt nach Manpages suchen, die mit port"beginnen:

```
# apropos ^port
ports(7)                 - contributed applications
portsnap(8)              - fetch and extract compressed snapshots of the ports tree
portsnap.conf(5)         - configuration file for portsnap(8)
port(1)                  - FreeBSD Port Tools front-end script
portaudit(1)             - system to check installed packages for known
    vulnerabilities
portcvswb(1)             - a tool to instantly browse a history via CVSweb or
    FreshPorts.org
portdowngrade(1)         - retrieve earlier port versions via CVS
portlint(1)              - a verifier for port directories
portsclean(1)            - a tool to clean ports/packages garbage
portsdb(1), ports_glob(1) - tools to manage and look up the ports database file
porttools(5), ~/.porttools(5) - FreeBSD Port Tools configuration file
portupgrade(1), portinstall(1) - tools to upgrade installed packages or install new
    ones via ports or packages
portversion(1)           - a tool to compare the versions of installed packages
    with the ones in the ports tree
```

Mehr Informationen zu apropos in der Manpage apropos(1) und zu Regular Expressions in der Manpage von grep(1).

### 3.7. Terminal sperren

[ Befehl der Woche ] [ FreeBSD ] [ OpenBSD ] by Beat @ 29.04.2006 10:41 CEST

Arbeitet man an einem Terminal und möchte seine Arbeit kurz unterbrechen, sich aber nicht ausloggen, so kann man mit lock(1) das Terminal sperren. lock ist sowohl in FreeBSD als auch in

### 3. BSD Allgemein

OpenBSD im Basissystem vorhanden. Gibt man einfach lock im Terminal ein, so wird man zweimal nach einem Schlüssel gefragt, mit dem man das Terminal für 15 Minuten sperren kann.

```
# lock
Key:
Again:
lock: /dev/tty1 on daedalus.network.local. timeout in 15 minutes.
time now is Thu Apr 27 22:57:15 CEST 2006
```

Verwendet man lock -p so wird nicht nach einem Schlüssel gefragt, sondern das Terminal kann mit dem Benutzerpasswort wieder entsperrt werden. Mit der -n Option wird das Terminal nach 15 Minuten nicht automatisch wieder entsperrt.

Wird auf FreeBSD zusätzlich die -v Option verwendet, so kann auch das virtuelle Terminal (mit Ctrl + Alt + Fx) nicht gewechselt werden, bis das Terminal entsperrt wird.

Verwendet man eine graphische Benutzeroberfläche und hat den xscreensaver installiert, so kann man auch die graphische Oberfläche mit folgendem Befehl sperren:

```
# xscreensaver -command -lock
```

Möchte man xscreensaver nicht installieren, so kann man die graphische Oberfläche auch mit xlock(1) sperren. Auf OpenBSD ist xlock bereits im Dateiset der graphischen Oberfläche enthalten, auf FreeBSD kann man xlock über den x11/xlockmore Port installieren.

### 3.8. Meldungen an eingeloggte Benutzer senden

[ Befehl der Woche ] [ FreeBSD ] [ OpenBSD ] by Beat @ 06.05.2006 09:46 CEST

Möchte man an andere eingeloggte Benutzer eine Nachricht senden, so kann man zuerst mit w(1) herausfinden, wer zur Zeit eingeloggt ist.

```
# w
6:08pm  up 18 mins, 5 users, load averages: 0,02 0,15 0,15
USER      TTY      FROM            LOGIN@   IDLE   WHAT
test      v0        -                6:07pm   -      -
beat      p1        :0.0             5:51pm   1      ssh chruetertee.ch
beat      p2        :0.0             6:07pm   -      w
beat      p3        :0.0             5:59pm   6      -
beat      p4        :0.0             6:04pm   3      csh
end{!stlisting}

Möchte man nun eine Nachricht an einen Benutzer schicken, so kann man dies mit
write(1)\index{write(1)} tun. Zum Starten gibt man

\begin{!stlisting}
# write <Benutzer>
```

ein. Anschliessend kann man die Nachricht eingeben und diese mit Ctrl + D senden.

```
# write test
Dies ist ein Test
^D
```

Der Empfänger erhält nun die Nachricht:

```
Message from beat@daedalus.network.local on tty2 at 18:28 ...
Dies ist ein Test
EOF
```

Möchte man die Nachricht an ein bestimmtes Terminal senden, so kann man dies wie folgt tun:

```
# write <Benutzer> [<tty>]
```

Mit mesg(1) kann man einstellen, ob man Nachrichten von anderen Benutzern empfangen will oder nicht. Ein mesg n zeigt keine Nachrichten von anderen Benutzern an, ein mesg y lässt Meldungen zu.

Möchte man eine Meldung an alle Benutzer senden, so kann man dies mit wall(1) machen:

```
# echo 'Hallo zusammen' | wall
```

### 3. BSD Allgemein

Die anderen Benutzer erhalten nun folgende Meldung:

```
Broadcast Message from beat@daedalus.network.local
(/dev/tty3) at 18:41 CEST...

Hallo zusammen
```

#### 3.9. Auf der Kommandozeile ans Beenden der Arbeit erinnert werden

[ Befehl der Woche ] [ FreeBSD ] [ OpenBSD ] by Beat @ 13.05.2006 11:58 CEST

Arbeitet man auf der Kommandozeile und hat nur noch eine gewisse Zeit zur Verfügung, bevor man die Arbeit beenden sollte, kann man mit `leave(1)` sichergehen, dass man die Zeit nicht vergisst. Wenn man weiss, man sollte zu einem bestimmten Zeitpunkt seine Arbeit beenden, so kann man `leave` mit der genauen Uhrzeit verwenden:

```
# leave [<hh>]<mm>
```

Für 11:00 wäre dies:

```
# leave 1100
Alarm set for Fr 12 Mai 2006 11:00:00 CEST. (pid 969)
```

Han man nur noch einen gewissen Zeitraum zur Verfügung, so kann man `leave` mit einem Plus verwenden:

```
# leave +[<hh>]<mm>
```

Für eine Viertelstunde wäre dies `leave +15` für eine Stunde und eine Minute ist dies `leave +101`. 5 Minuten bevor die Zeit abläuft, wird eine Warnung ausgegeben:

```
You have to leave in 5 minutes.
```

Auch eine Minute vor Ablauf der Zeit wird man nochmals darauf hingewiesen, dass man die Arbeit langsam abschliessen sollte:

```
Just one more minute!
```

Ist die Zeit abgelaufen, wird man immer wieder mit folgender Nachricht daraufhingewiesen, dass man jetzt doch gehen sollte:

```
Time to leave!
```

Jetzt hat man zwei Möglichkeiten, entweder wirklich zu gehen oder den `leave`-Prozess z.B. mit `kill(1)` zu beenden. Für das erste Beispiel wäre dies:

```
kill 969
```

#### 3.10. Terminalausgabe aufzeichnen

[ Befehl der Woche ] [ FreeBSD ] [ OpenBSD ] by Beat @ 08.07.2006 11:08 CEST

Möchte man die Terminalausgabe, zum Beispiel bei einem längeren `make`-Durchlauf, festhalten oder aufzeichnen, was bei einem Portupgrade gemacht wurde, so kann man das mit `script(1)` machen. Beim Start von `script` kann man mit der Option `-a` eine Datei angeben in welche die Ausgabe geschrieben wird:

```
# script -a /tmp/ausgabe
```

Nun kann man die gewünschten Kommandos eingeben und alles wird in die angegebene Datei aufgezeichnet. Um die Aufzeichnung zu stoppen, gibt man einfach `exit` im Terminal ein. Danach bestätigt `script`, dass die Ausgabe nach `/tmp/ausgabe` geschrieben wurde:

```
Script done, output file is /tmp/ausgabe
```

`script` ist sowohl auf OpenBSD als auch auf FreeBSD im Basissystem vorhanden. Auf FreeBSD hat `script` noch einige zusätzliche Optionen. Die `-k` Option schreibt alle gedrückten Tasten auch in die Datei und die `-q` Option unterdrückt die Start und die Stoppmeldung.

### 3. BSD Allgemein

#### 3.11. Komprimierte Textdateien lesen

[ Befehl der Woche ] [ FreeBSD ] [ OpenBSD ] by Beat @ 22.07.2006 10:32 CEST

Möchte man eine mit gzip oder bzip2 komprimierte Datei anschauen, so muss man diese nicht zuerst mit gunzip oder bunzip2 entpacken, sondern kann diese direkt mit bzcat(1) oder gzcat(1) auslesen:

```
# echo Dies ist eine Testdatei > test
# gzip test
# gzcat test.gz
Dies ist eine Testdatei

# echo Dies ist eine andere Testdatei > test
# bzip2 test
# bzcat test.bz2
Dies ist eine andere Testdatei
```

Mit bzip2 komprimierte Dateien können auch direkt mit Hilfe von bzgrep gegrept werden. bzcat, gzcat und bzgrep befinden sich auf FreeBSD im Basissystem. Auf OpenBSD muss bzcat und bzgrep mit dem archivers/bzip2 Port installiert werden.

#### 3.12. Herausfinden welche Benutzer angemeldet sind

[ Befehl der Woche ] [ FreeBSD ] [ OpenBSD ] by Beat @ 12.08.2006 16:03 CEST

Es gibt verschiedene Möglichkeiten herauszufinden, welche Benutzer auf einem Rechner angemeldet sind.

Mit users(1) werden die angemeldeten Benutzer in alphabetischer Reihenfolge angezeigt:

```
# users
beat test
```

Mit who(1) werden zusätzlich noch der tty-Name, die Zeit des Logins und gegebenenfalls der Name des Hosts, von dem der Benutzer aus angemeldet ist, angezeigt:

```
# who
test          ttyv0      11 Aug 19:18
beat          ttyv1      11 Aug 16:42 (:0.0)
beat          ttyv3      11 Aug 17:59 (:0.0)
```

Noch mehr Informationen erhält man mit w(1). Hier sieht man zusätzliche Informationen über das System, wie die Uptime und den System-Load. Zusätzlich wird bei allen Benutzern angezeigt, wie lange sie keine Eingabe mehr getätigt haben und welcher Befehl aktuell ausgeführt wird:

```
# w
7:38pm  up   4:05, 3 users, load averages: 0,03 0,12 0,14
USER      TTY      FROM          LOGIN@  IDLE WHAT
test      v0        -             7:18pm   19 -
beat      p1        :0.0          4:42pm   2 more
beat      p3        :0.0          5:59pm   - w
```

Möchte man herausfinden, welche Benutzer sich in letzter Zeit angemeldet haben, so kann man dies mit last(1) nachschauen:

```
# last
beat          ttyv5      localhost      Fr 11 Aug 19:50 - 19:51 (00:01)
test          ttyv0      -              Fr 11 Aug 19:18   still logged in
beat          :0         -              Fr 11 Aug 15:34   still logged in
reboot        ~         -              Fr 11 Aug 15:33
shutdown      ~         -              Fr 11 Aug 14:05
beat          :0         -              Fr 11 Aug 14:00 - shutdown (00:05)
reboot        ~         -              Fr 11 Aug 14:00
shutdown      ~         -              Fr 11 Aug 12:43
```

users, who, w und last sind sowohl auf FreeBSD als auch auf OpenBSD im Basissystem vorhanden.

## 4. Programme

### 3.13. Grösse eines Verzeichnisses bestimmen

[ Befehl der Woche ] [ FreeBSD ] [ OpenBSD ] by Beat @ 19.08.2006 08:33 CEST

Die Grösse eines Verzeichnisses lässt sich mit `du(1)` bestimmen. Mit der `-c` Option wird das Total aller Dateien angezeigt, mit der `-h` Option werden die Grössen in Kilo-, Mega- oder Gigabyte angezeigt. Beispiel:

```
# ls -l
total 2869856
-rw-r--r--  1 beat  users  529784832  7 Mai 20:50 6.1-RELEASE-i386-disc1.iso
-rw-r--r--  1 beat  users  548517888  5 Jun 15:25 7.0-CURRENT-200606-i386-disc1.iso
-rw-r--r--  1 beat  users  721241069 29 Sep  2005
      KNOPPIX_V4.0.2CD-2005-09-23-DE.iso.gz
-rw-r--r--  1 beat  users  208437248 12 Feb 16:45 frenzy_v03_release.iso
-rw-r--r--  1 beat  users  209682432  9 Feb  2006 i386cd-3.0.iso
-rw-r--r--  1 beat  users  703017311  5 Sep  2005
      install-x86-universal-2005.1.iso.gz
-rw-r--r--  1 beat  users  16425313 10 Jul 13:28 reactos0.3.0-RC1-live.zip
# du -ch
2,7G  .
2,7G  total
```

Möchte man nun in einem Verzeichnis herausfinden, welche Dateien am meisten Speicherplatz benötigen, so kann man `du` mit `sort(1)` kombinieren:

```
# du * | sort -n
```

Nun wird der Verzeichnissinhalt so sortiert, dass die grossen Dateien am Ende stehen:

```
# du * | sort -n
16064  reactos0.3.0-RC1-live.zip
203680 frenzy_v03_release.iso
204896 i386cd-3.0.iso
517648 6.1-RELEASE-i386-disc1.iso
535952 7.0-CURRENT-200606-i386-disc1.iso
686896 install-x86-universal-2005.1.iso.gz
704720 KNOPPIX_V4.0.2CD-2005-09-23-DE.iso.gz
```

`du(1)` und `sort(1)` stehen beide auf FreeBSD und OpenBSD im Basissystem zur Verfügung.

## 4. Programme

### 4.1. Schöner surfen mit Firefox

[ Befehl der Woche ] by Beat @ 22.10.2005 09:42 CET

Der Befehl der Woche ist diese Woche mal kein Befehl, sondern ein Plugin für den Browser meines Vertrauens, da die Werbung im Internet langsam Überhand nimmt. Zum Glück gibt es da das Adblock Projekt, das ein Plugin für den Firefox bereitstellt. Adblock ist in den FreeBSD Ports unter `www/adblock` zu finden.

Unter FreeBSD einfach mit folgendem Befehl installieren:

```
# cd /usr/ports/www/adblock && make install clean
```

Den Firefox neu starten und Adblock unter Tools -> Adblock -> Preferences konfigurieren. Meine Filter sehen so aus:

```
*.adboost.*
*.swf
*/ad.*
*/ad/*
*/ads/*
*/ads.*
*/banner/*
*/banners.*
*adserver*
*adtech*
```



## 4. Programme

```
*doubleclick*
*/easyad.*
```

Viel Spass beim werbefreien Surfen.

### 4.2. mySQL Datenbank sauber herunterfahren und auf Fehler prüfen

[ Allgemein ] by Beat @ 11.11.2005 12:20 CET

Um eine mySQL Datenbank sauber herunterzufahren ist folgender Befehl zu benutzen:

```
# mysqladmin shutdown -p
```

Danach muss das mySQL root Passwort eingegeben werden. Mehr Informationen zu mysqladmin mit:

```
# mysqladmin --help
```

Mit mysqlcheck kann die Datenbank auf Fehler überprüft werden:

```
# mysqlcheck -A -e -p
```

Die Option -A prüft alle vorhandenen Datenbanken und -e führt eine ausführliche Konsistenzüberprüfung durch. Mehr Informationen zu mysqlcheck mit:

```
# mysqlcheck --help
```

### 4.3. Jobkontrolle in der Shell mit bg, fg und jobs

[ Befehl der Woche ] by Beat @ 10.12.2005 08:14 CET

Mit Ctrl + z wird ein in der Shell aktiver Prozess in den Hintergrund verlagert. Dabei wird dieser gestoppt. Mit bg wird der gestoppte Prozess im Hintergrund wieder gestartet. jobs listet alle aktiven und gestoppten Prozesse im Hintergrund auf. Um einen Hintergrundprozess wieder in den Vordergrund zu holen, verwendet man fg.

Jeder Hintergrundprozess kriegt eine Jobnummer, damit können mehrere Prozesse in den Hintergrund verschoben werden und diese dann explizit wieder gestartet oder zurück in den Vordergrund gebracht werden. jobs listet zu jedem Hintergrundprozess die jeweilige Jobnummer auf. So kann z.B. mit fg <Jobnummer> der Hintergrundprozess mit dieser Jobnummer wieder in den Vordergrund gebracht werden.

### 4.4. E-Mails mit Attachment von der Konsole aus senden

[ Befehl der Woche ] by Beat @ 14.01.2006 11:00 CET

Möchte man ein E-Mail mit einem kurzen Text von der Konsole aus schreiben, so kann man einfach folgenden Befehl benutzen:

```
# echo "<Text>" | mail -s "<Betreff>" empfänger@domain
```

Möchten man einen längeren Text schreiben, so wird dieser in eine Textdatei geschrieben und dann so versendet:

```
# mail -s "<Betreff>" empfänger@domain < <Textdatei>
```

Möchte man dieses Mail an mehrere Benutzer mit CC senden, so verwendet man die -c Option und listet die Benutzer mit einem Komma abgetrennt auf:

```
# mail -s "<Betreff>" -c empfänger2@domain,empfänger3@anderedomain empfänger@domain
< <Textdatei>
```

Möchte man nun eine binäre Datei versenden, so muss man diese zuerst mit uuencode umwandeln.

```
# uuencode <Binärdatei> <Dateiname> | mail -s "<Betreff>" empfänger@domain
```

Mit folgendem Befehl versendet man einen Text und die binäre Datei.

## 4. Programme

```
# (more <Textdatei>; uuencode <Binärdatei> <Dateiname>) | mail -s "<Betreff>"  
empfänger@domain
```

Möchte man diese binäre Datei auf der Konsole aus dem E-Mail extrahieren, so verwendet man uudecode:

```
# uudecode <Pfad zu E-Mail>
```

Dies entpackt die Datei im aktuellen Verzeichnis mit dem Dateinamen der beim codieren angegeben wurde.

### 4.5. DOS Zeilenumbrüche CR/LF in Unix LF umwandeln

[ Befehl der Woche ] by Beat @ 25.02.2006 12:51 CET

Besitzt man Textdateien, die DOS-Zeilenumbrüche enthalten, so kann man diese mit dos2unix in Unix-Zeilenumbrüche umwandeln. Um DOS- in Unix-Zeilenumbrüche umzuwandeln benutzt man dos2unix:

```
# dos2unix <Textdatei>
```

Falls man Unix- in DOS-Zeilenumbrüche umwandeln will, benutzt man unix2dos:

```
# unix2dos <Textdatei>
```

Beide findet man in den FreeBSD Ports unter converters/unix2dos.

### 4.6. Konsolen verwalten mit GNU Screen

[ Befehl der Woche ] by Beat @ 11.03.2006 10:42 CET

Mit GNU Screen lassen sich mehrere Terminals parallel verwalten. So lassen sich zum Beispiel mit nur einer SSH Verbindung zu einem entfernten Rechner mehrere Konsolen öffnen, in diesen diverse Programme starten und einfach zwischen den Konsolen hin und herschalten. Ein weiterer sehr grosser Vorteil ist, wenn die SSH-Verbindung unterbrochen wird, so sind die verschiedenen gestarteten Konsolen nicht verloren, sondern laufen in Screen weiter, so dass man die Arbeit zu einem späteren Zeitpunkt fortsetzen kann.

GNU Screen befindet sich in den Ports auf FreeBSD unter sysutils/screen auf OpenBSD unter misc/screen.

Screen wird auf der Kommandozeile gestartet:

```
# screen
```

Danach kann Screen mit verschiedenen Tastenkombinationen bedient werden:

Ctrl + a ?	Zeigt alle Tastenkombinationen an
Ctrl + a c	Erzeugt eine weitere Konsole in Screen
Ctrl + a n	Wechselt zur nächsten Konsole
Ctrl + a p	Wechselt zur vorherigen Konsole
Ctrl + a A	Editiert den Namen der aktuellen Konsole
Ctrl + a w	Zeigt an, auf welcher Nummer sich welche Konsole befindet
Ctrl + a 0...9	Wechselt zur Konsole mit der jeweiligen Nummer
Ctrl + a N	Zeigt die Nummer der aktuellen Konsole an
Ctrl + a S	Splittet das aktuelle Fenster
Ctrl + a Ctrl + i	Wechselt zwischen den gesplitteten Fenstern
Ctrl + a \	Beendet Screen und alle Konsolen werden auch beendet
Ctrl + a d	Trennt Screen. Arbeit kann zu einem späteren Zeitpunkt an der gleichen Stelle wieder aufgenommen werden

Wurde eine Session mit Ctrl + a d geschlossen, kann man diese zu einem späteren Zeitpunkt wieder aufnehmen. Um herauszufinden, ob noch Sessions laufen, kann man folgende Option benutzen:

```
# screen -list
```

Um eine gestartete Screen-Session fortzusetzen, benutzt man die -r Option:

```
# screen -r
```

## 4. Programme

Möchten mehrere Benutzer an der gleichen Session arbeiten, so kann zusätzlich die `-x` Option verwendet werden, die es erlaubt, dass mehrere Benutzer die gleiche Session verwenden:

```
# screen -xr
```

Man kann der Screen-Session auch einen Namen geben, so dass man mehrere Sessions gestartet haben kann und diese dann einfach identifizieren kann

```
# screen -S <Sessionname>
```

Hat man diese Session wieder mit `Ctrl + a d` geschlossen, so kann man diese zu einem späteren Zeitpunkt wie folgt wieder aufnehmen:

```
# screen -r <Sessionname>
```

Eine ausführliche Beschreibung aller Optionen und Tastenkombinationen findet man in der Manpage `screen(1)`.

### 4.7. WebDAV Client für die Kommandozeile

[ Allgemein ] [ Befehl der Woche ] by Beat @ 08.04.2006 13:13 CEST

Möchte man auf Daten zugreifen, die per WebDAV freigegeben sind, so kann man dies auch auf der Kommandozeile mit `cadaver` machen. Auf FreeBSD befindet sich `cadaver` unter `www/cadaver`, auf OpenBSD befindet es sich unter `net/cadaver`. Auf FreeBSD installiert man `cadaver` wie folgt:

```
# cd /usr/ports/www/cadaver && make install clean
```

Danach kann man sich mit der WebDAV-Ressource verbinden und gegebenenfalls Authentifizieren:

```
# cadaver http[s]://<hostname>[:<port>]/<pfad>
Authentication required for Ressource on server 'WebDAV Server':
Username: <Benutzername>
Password: <Passwort>
```

Danach steht ein Prompt zur Verfügung:

```
dav:/<pfad>/>
```

Hier stehen einem verschiedene Kommandos zur Auswahl, die man auch von verschiedenen Kommandozeilen `ftp/sftp` Clients her kennt:

```
dav:/<pfad>/> help
Available commands:
ls      cd      pwd      put      get      mget     mput
edit    less    mkcol    cat      delete   rmcol    copy
move    lock    unlock   discover steal    showlocks version
checkin checkout uncheckout history label    propnames chexec
propget propdel propset   search   set      open     close
echo    quit    unset    lcd      lls      lpwd     logout
help    describe about
Aliases: rm=delete, mkdir=mkcol, mv=move, cp=copy, more=less, quit=exit=bye
```

Hier die wichtigsten Kommandos:

**pwd** Zeigt den Pfad auf dem WebDAV Server an

**lpwd** Zeigt den lokalen Pfad an

Mit **cd** und **lcd** werden die Verzeichnisse auf dem Server und auf dem lokalen Rechner gewechselt.

**ls** Listet den Inhalt des aktuellen Verzeichnisses auf dem WebDAV Server auf

**lls** Listet den Inhalt des aktuellen Verzeichnisses auf dem lokalen Rechner auf

**get <Datei>** Läd die Datei vom aktuellen Verzeichnis auf dem Server ins lokale Verzeichnis auf dem Rechner

**put <Datei>** Läd die Datei vom lokalen Verzeichnis auf dem Rechner ins aktuelle Verzeichnis auf dem WebDAV Server

Mit **mget** und **mput** können mehrere Dateien auf einmal hinauf- oder heruntergeladen werden.

Mit **close** wird die aktuelle Verbindung getrennt und mit **quit** wird `cadaver` beendet.

Alle Kommandos können in der Manpage `cadaver(1)` nachgelesen werden.

### 4.8. Leere Verzeichnisse und alte Symlinks löschen

[ Befehl der Woche ] by Beat @ 27.05.2006 11:21 CEST

Mit der Zeit können sich auf einem System leere Verzeichnisse und symbolische Links, die auf nicht mehr existierende Dateien verweisen, ansammeln. Mit `cleanlinks(1)` können diese automatisch gelöscht werden. `cleanlinks` sucht im aktuellen Verzeichnis nach solchen Daten und durchsucht auch rekursiv alle Unterverzeichnisse und löscht sie.

Beispiel:

```
# mkdir /tmp/test
# cd /tmp/test/
# mkdir leer
# mkdir voll
# touch voll/datei
# ln -s a b
# ln -s voll/datei hallo
# pwd
/tmp/test
# ls -l
total 4
lrwxr-xr-x  1 beat  wheel   1 27 Mai 09:59 b@ -> a
lrwxr-xr-x  1 beat  wheel  10 27 Mai 09:59 hallo@ -> voll/datei
drwxr-xr-x  2 beat  wheel 512 27 Mai 09:59 leer/
drwxr-xr-x  2 beat  wheel 512 27 Mai 09:59 voll/
```

Im Verzeichnis `/tmp/test` existiert nun ein leeres Verzeichnis `leer` und ein Symlink `b`, der auf eine nicht existierende Datei `a` verweist. Führen wir nun `cleanlinks` in diesem Verzeichnis `test` aus:

```
# cleanlinks
./b is a dangling symlink, removing
Removing empty directories ...
./leer
# ls -l
total 2
lrwxr-xr-x  1 beat  wheel  10 27 Mai 09:59 hallo@ -> voll/datei
drwxr-xr-x  2 beat  wheel 512 27 Mai 09:59 voll/
```

`cleanlinks` ist Teil der XFree86 und X.Org Projekte. Auf FreeBSD befindet sich `cleanlinks` im `imake` Port.

### 4.9. Rechnen auf der Kommandozeile

[ Befehl der Woche ] by Beat @ 03.06.2006 11:43 CEST

Auf der Kommandozeile kann man mit Hilfe von `bc(1)` auch Rechnen. Rechnet man mit Kommazahlen, sollte man die `-l` Option benutzen, so werden 20 Nachkommastellen angezeigt, ohne `-l` werden keine angezeigt. Ruft man `bc` auf, so kann man danach seine Rechnung eingeben und nach einem Enter wird das Ergebnis angezeigt. Danach kann eine weitere Rechnung eingegeben werden. `bc` kann mit `quit`, `exit` oder `Ctrl + d` beendet werden.

```
# bc -l
bc 1.06
Copyright 1991-1994, 1997, 1998, 2000 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type 'warranty'.
sqrt((3+2-1)*(9/3)-3)
3.00000000000000000000
9/4
2.25000000000000000000
10/3
3.33333333333333333333
exit
```

Die Operatoren `+` `-` `*` `/` sollten selbsterklärend sein, `sqrt()` ist die Quadratwurzel, `^` ist das Potenzzeichen, also  $2^3$  bedeutet 2 hoch 3:

## 4. Programme

```
# echo "2^3" | bc
8
```

Hier sieht man, wie man eine Rechnung mittels Pipe an bc übergeben kann.  
Auch Modulo-Operationen lassen sich mit

```
# echo "9%5" | bc
4
```

Mit bc lässt sich noch viel mehr machen, dies ist ausführlich in bc(1) beschrieben.

### 4.10. Verschiedene Komprimierungsmöglichkeiten

[ Befehl der Woche ] by Beat @ 24.06.2006 12:09 CEST

Auf Free- und OpenBSD gibt es verschiedene Komprimierungswerkzeuge.

#### **compress**

Nahezu auf jedem unixoiden Betriebssystem verfügbar. Mit compress(1) komprimierte Dateien tragen die Endung .Z.

Eine Datei wird wie folgt mit compress komprimiert und wieder entpackt:

```
# ls -aol test
-rw-r--r--  1 beat  wheel  3067 24 Jun 10:48 test
# compress test
# ls -aol test.Z
-rw-r--r--  1 beat  wheel  2117 24 Jun 10:48 test.Z
# uncompress test.Z
# ls -aol
-rw-r--r--  1 beat  wheel  3067 24 Jun 10:48 test
```

Mehrere Dateien müssen zuerst mit tar(1) zu einem Archiv gepackt werden, bevor sie mit compress komprimiert werden können.

compress ist sowohl auf Free- als auch auf OpenBSD im Basissystem verfügbar.

#### **bzip2**

Gute Kompressionsrate, dafür dauert das komprimieren länger. Mit bzip2(1) komprimierte Dateien tragen die Endung .bz2.

Eine Datei wird wie folgt mit bzip2 komprimiert und wieder entpackt:

```
# ls -aol test
-rw-r--r--  1 beat  wheel  - 3067 24 Jun 10:48 test
# bzip2 test
# ls -aol test.bz2
-rw-r--r--  1 beat  wheel  - 1689 24 Jun 10:48 test.bz2
# bunzip2 test.bz2
# ls -aol test
-rw-r--r--  1 beat  wheel  - 3067 24 Jun 10:48 test
```

Mehrere Dateien müssen zuerst mit tar(1) zu einem Archiv gepackt werden, bevor sie mit bzip2 komprimiert werden können. Im folgenden Beispiel wird das Verzeichnis Verzeichnis zusammen mit dem ganzen Inhalt komprimiert und mit dem zweiten Befehl wieder entpackt.

```
# tar -cvpyf Verzeichnis.tar.bz2 Verzeichnis
# tar -xvyf test.tar.bz2
```

bzip2 ist auf FreeBSD bereits im Basissystem enthalten, auf OpenBSD findet man bzip2 im archivers/bzip2 Port.

#### **gzip**

Beliebtes Kompressionsprogramm. Mit gzip(1) komprimierte Dateien tragen die Endung .gz.

Eine Datei wird wie folgt mit gzip komprimiert und wieder entpackt:

```
# ls -aol test
-rw-r--r--  1 beat  wheel  - 3067 24 Jun 10:48 test
# gzip test
# ls -aol test.gz
-rw-r--r--  1 beat  wheel  - 1529 24 Jun 10:48 test.gz
# gunzip test
```

## 4. Programme

```
#ls -aol test
-rw-r--r--  1 beat  wheel  - 3067 24 Jun 10:48 test
```

Mehrere Dateien müssen zuerst mit `tar(1)` zu einem Archiv gepackt werden, bevor sie mit `gzip` komprimiert werden können. Im folgenden Beispiel wird das Verzeichnis `Verzeichnis` zusammen mit dem ganzen Inhalt komprimiert und mit dem zweiten Befehl wieder entpackt.

```
# tar -cvpzf Verzeichnis.tar.gz Verzeichnis
# tar -xvzf test.tar.gz
```

`gzip` ist sowohl auf Free- als auch auf OpenBSD im Basissystem verfügbar.

### zip

Um Daten mit Windowsbenutzern auszutauschen, empfiehlt sich `zip(1)`. Mit `zip` komprimierte Dateien tragen die Endung `.zip`.

Eine Datei wird wie folgt mit `zip` komprimiert und wieder entpackt:

```
# ls -aol test
-rw-r--r--  1 beat  wheel  - 3067 24 Jun 10:48 test
# zip test.zip test
  adding: test (deflated 51%)
# ls -aol test.zip
-rw-r--r--  1 beat  wheel  - 1646 24 Jun 11:34 test.zip
# unzip test.zip
Archive:  test.zip
replace test? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
  inflating: test
# ls -aol test
-rw-r--r--  1 beat  wheel  - 3067 24 Jun 10:48 test
```

Mehrere Dateien lassen sich direkt mit `zip` komprimieren. Im folgenden Beispiel wird das Verzeichnis `Verzeichnis` zusammen mit dem ganzen Inhalt komprimiert:

```
# zip -r Verzeichnis.zip Verzeichnis
```

`zip` findet man auf Free- und OpenBSD im `archivers/zip` Port.

### 4.11. MP3 und Ogg-Dateien aus Musik CD's erstellen

[ Befehl der Woche ] by Beat @ 01.07.2006 10:05 CEST

Mit `abcde` (A Better CD Encoder) lassen sich einfach MP3 oder Ogg-Dateien aus Musik CD's erstellen. Dabei wird automatisch eine CDDb abgefragt und die ID3-Tags der MP3's gesetzt. `abcde` findet man auf FreeBSD in den Ports unter `audio/abcde`:

```
cd /usr/ports/audio/abcde/ && make install clean
```

Danach kann `abcde` über die `/usr/local/etc/abcde.conf` konfiguriert werden. Wichtig ist, dass man zuerst in der `abcde.conf` das CD-ROM Laufwerk angibt. Zum Beispiel:

```
CDROM=/dev/acd0
```

Standardmässig werden von `abcde` Ogg-Dateien erstellt. Möchte man MP3's erstellt haben, wechselt man den `OUTPUTTYPE` in der `abcde.conf`:

```
OUTPUTTYPE=mp3
```

Eine Beschreibung aller Optionen findet man in der Manpage `abcde(1)`.

Nun kann man einfach eine Musik CD ins CD-ROM Laufwerk einlegen und in dem Verzeichnis, in dem man den Ordner mit den Musik-Dateien haben möchte, `abcde` eingeben und warten, bis `abcde` beendet ist. Die nun erzeugten MP3's kann man dann, wenn man möchte, gleich auf den iPod laden.

## 5. vi(1)

### 4.12. Anzeigen welcher Host wieviel Bandbreite benutzt

[ Befehl der Woche ] [ FreeBSD ] by Beat @ 26.08.2006 08:36 CEST

Mit iftop(8) lassen sich die Verbindungen einer Netzwerkschnittstelle im Stile von top(1) anzeigen. Mit der -i Option kann man die Netzwerkschnittstelle angeben, die man beobachten will:

```
# iftop -i <Netzwerkschnittstelle>
```

Nun werden alle Verbindungen untereinander angezeigt und ein schwarzer Balken visualisiert die zur Zeit verwendete Bandbreite jeder Verbindung, welche jedoch in gesendete und empfangene Daten aufgeteilt ist. Zusätzlich werden am rechten Rand das Total der empfangenen und gesendeten Daten der letzten 40, 10 und 2 Sekunden angezeigt.

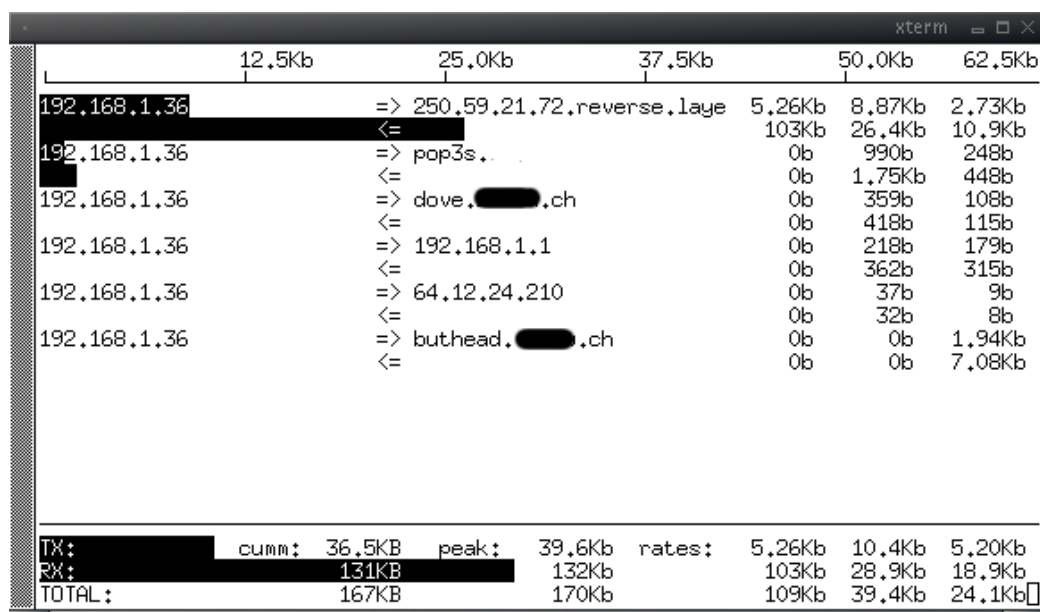


Abbildung 4: iftop

Am unteren Rand wird das Total aller Verbindungen der angezeigten Schnittstelle angezeigt. Da iftop die IP-Adressen in Hostnamen auflöst, ist es möglich, dass das Ergebnis von iftop verfälscht wird, da iftop nun selber Netzwerkverkehr generiert. Das Auflösen der IP-Adressen kann mit der -n Option unterbunden werden.

iftop kann nur als root ausgeführt werden. Auf FreeBSD befindet sich iftop unter net-mgmt/iftop in den Ports. Alle Optionen von iftop findet man in der Manpage iftop(8).

## 5. vi(1)

### 5.1. vi Kurzreferenz

[ Befehl der Woche ] by Beat @ 26.11.2005 05:25 CET

Ich werde öfters gefragt, wieso ich den vi als Editor brauche. Ganz einfach, der kann viel und man editiert so schnell, wie mit keinem anderen Editor. Deshalb möchte ich den Befehl der Woche dem vi widmen.

Bewegen	
h, j, k, l	links, runter, hinauf, rechts
Pfeiltasten	Hoffentlich selbsterklärend

## 5. vi(1)

w, b	Springt jeweils ein Wort vorwärts/rückwärts
E	Springt ans Ende des aktuellen Wortes
0, \$	Springt zum Anfang/Ende der Zeile
H, M, L	Springt an den Anfang, in die Mitte oder ans Ende des Fensters
G	Springt ans Ende der Datei
1G	Springt an den Anfang der Datei
<b>Suchen</b>	
/ <Wort> [Enter]	Sucht vorwärts nach <Wort> und setzt den Cursor auf den ersten Buchstaben
? <Wort> [Enter]	Sucht rückwärts nach <Wort> und setzt den Cursor auf den ersten Buchstaben
n, N	Sucht nach <Wort> vorwärts, rückwärts fortsetzen
<b>Schreiben</b>	
i, a	Fügt den nachfolgenden Text vor/nach dem Buchstaben unter dem Cursor ein
I, A	Fügt den nachfolgenden Text am Anfang/Ende der Zeile ein
o, O	Fügt eine neue Zeile nach/vor der aktuellen Zeile ein
r	Ersetzt das aktuelle Zeichen unter dem Cursor
<b>Löschen</b>	
x	Löscht das aktuelle Zeichen unter dem Cursor
dh	Löscht das Zeichen links des Cursors
dw	Löscht das aktuelle Wort
db	Löscht das Wort links des aktuellen Wortes auf dem sich der Cursor befindet
dd	Löscht aktuelle Zeile
d0, d\$	Löscht die Zeile von der aktuellen Cursorposition zum Zeilenanfang/-ende
dG	Löscht alle folgenden Zeilen, einschliesslich der aktuellen, bis zum Ende der Datei
J	Löscht den Zeilenumbruch zwischen der aktuellen und der nächsten Zeile
d/ <Wort>	Löscht von der aktuellen Cursorposition alles bis zu <Wort>
<b>Navigieren</b>	
: <Ziellenummer>	Geht zur angegebenen Zeile
<b>Suchen und ersetzen</b>	
:s/ <suche> / <ersetze> /	Sucht <suche> auf der aktuellen Zeile und ersetzt den ersten Treffer durch <ersetze>
:s/ <suche> / <ersetze> /g	Sucht <suche> auf der aktuellen Zeile und ersetzt alle durch <ersetze>
:%s/ <suche> / <ersetze> /g	Sucht <suche> in der ganzen Datei und ersetzt es durch <ersetze>
:%s/ <suche> / <ersetze> /gc	Sucht <suche> in der ganzen Datei und fragt nach, ob er es durch <ersetze> ersetzen soll
<b>Weiteres</b>	
u	Macht den letzten ausgeführten Befehl rückgängig
. <- Punkt	Führt den letzten ausgeführten Befehl nochmals aus
~	Ändert Grossbuchstaben in Kleinbuchstaben und umgekehrt
<b>Kommandomodus</b>	
[Esc]	Wechselt in den Kommandomodus
:w / :w <Dateiname>	Speichert Datei / Speichert Datei in <Dateiname>
:q / :q!	Verlässt den vi / Verlässt vi ohne geänderte Datei zu speichern



## 5. vi(1)

:wq oder :x	Speichert Datei und verlässt den vi
:! <Befehl>	Führt <Befehl> in der Shell aus
:e <Dateiname>	Erstellt neue Datei namens <Dateiname>
Kopieren und Einfügen	
yy	Kopiert aktuelle Zeile
p, P	Fügt kopierte Zeile nach/vor der aktuellen Zeile ein

Lösch- und Bewegungsbefehle können auch noch mit Zahlen kombiniert werden, so löscht zum Beispiel ein 3dw gleich 3 Wörter auf einmal oder ein 5h bewegt den Cursor gleich 5 Zeichen nach links.

Leider weiss ich auch nicht alle Befehle auswendig, aber dafür hab ich ja meine vi Referenz Kaffeetasse.

### 5.2. vi Makros benutzen

[ Allgemein ] by Beat @ 23.03.2006 15:21 CET

Im vi(1) lassen sich oft gebrauchte vi-Kommandofolgen zu einem Makro zusammenfassen. Makros werden mittels des ex-Befehles map oder map! definiert. Werden Makros mit map erstellt, so kann das Makro aus dem Kommandomodus aufgerufen werden, map! Makros werden aus dem Textmodus aufgerufen. Um die ex-Befehle map/map! ausführen zu können, muss man sich zuerst im Kommandomodus befinden, dies erreicht man indem man im vi die Esc-Taste drückt. Danach kann mittels des Doppelpunktes in den ex-Modus umgeschaltet werden und das Makro definiert werden:

```
map[!] <Taste> <Kommandofolge>
```

Um ein Makro zu definieren, welches im Kommandomodus durch die q-Taste aufgerufen wird und an der aktuellen Cursorposition subsection einfügt, danach ans Ende der Zeile springt und dort ein einfügt, kann folgender Befehl gebraucht werden:

```
map q i\subsection{^[a}
```

Die Kommandofolge wird dabei so eingegeben, wie wenn man das Kommando direkt im vi benutzt:

i	Wechselt in den Textmodus
\subsection{	Fügt an der aktuellen Cursorposition \subsection{ ein
^[	Esc-Taste, um in den Kommandomodus zu wechseln. Muss als Ctrl + v Esc eingegeben werden
\$	Springt ans Zeilenende
a	Wechselt in den Textmodus, das nächste Zeichen wird nach der aktuellen Cursorposition eingefügt
}	Fügt eine schliessende, geschweifte Klammer ein

Wie in diesem Beispiel gesehen wurde, müssen Steuerzeichen wie Esc, Return, Backspace mit Ctrl + v in den Makros maskiert werden.

Makros können auch auf die Funktionstasten F1 - F12 gelegt werden. Diese Tasten werden durch eine Raute # und die jeweilige Nummer symbolisiert. Folgendes Makro wird aus dem Kommandomodus mit F1 aufgerufen, fügt eine neue Zeile ein, fügt auf dieser \begin{lstlisting} ein und setzt den Cursor im Textmodus auf eine neue Zeile:

```
map #1 o\begin{lstlisting}^M
```

o	Fügt eine neue Zeile nach der aktuellen Cursorposition ein und wechselt danach in den Textmodus
---	---

## 5. vi(1)

<code>\begin{lstlisting}</code>	Dieser Text wird auf der neuen Zeile eingefügt
<code>^M</code>	Return-Taste. Ein Zeilenumbruch wird vorgenommen. Wird als Ctrl + v Return eingegeben

Dieses Makro kann nur aus dem Kommandomodus aufgerufen werden. Es kann jedoch für die gleiche Taste noch ein anderes Makro definiert werden, das aber aus dem Textmodus aufgerufen wird. So kann zum Beispiel jetzt noch ein Makro auf die F1 Taste gelegt werden, dass im Textmodus aufgerufen wird, welches in den Kommandomodus wechselt und dort F1 aufruft:

```
map! #1 ^[[OP
```

<code>^[</code>	Esc-Taste um in den Kommandomodus zu wechseln. Wird als Ctrl + v Esc eingegeben
<code>^[OP</code>	F1-Taste um das Kommandomodus-Makro auf F1 auszuführen. Wird als Ctrl + v F1 eingegeben

So lassen sich natürlich interessante Sachen wie folgende zwei Makros machen:

```
map #1 i^[OP
map! #1 ^[[OP
```

Das erste Makro wechselt durch das Drücken der F1-Taste in den Textmodus und wählt die F1-Taste, welche wiederum durch das zweite Makro in den Kommandomodus wechselt und dort die F1-Taste auswählt... Drückt man nun die F1 Taste, lässt sich dieses Spielchen dann nur noch durch ein `kill -9 <vi-PID>` beenden.

Eine Kurzreferenz von vi Kommandos findet man hier: <http://www.chruetertee.ch/blog/archive/2005/11/26/vi-kurzreferenz.html>

Anstatt Makros jedes mal wieder neu einzugeben, lassen sich diese auch in der `.exrc` dauerhaft speichern. Einfach die Makrozeile in die `.exrc` des Benutzers einfügen oder in der `/etc/vi.exrc` definieren, dann stehen die Makros allen Benutzern zur Verfügung.

### 5.3. Abkürzungen im vi benutzen

[ Allgemein ] by Beat @ 29.03.2006 13:48 CEST

Im vi(1) lassen sich Abkürzungen (**abbreviations**) definieren. So kann man für oft gebrauchte Wörter, Satzteile oder Sätze Abkürzungen definieren, welche dann automatisch vom vi umgewandelt werden. Abkürzungen werden mittels des `ex`-Befehls `ab` festgelegt. Dazu muss man sich zuerst im Kommandomodus befinden (dies erreicht man, indem man die Esc-Taste drückt) und danach mit dem Doppelpunkt in den `ex` Modus wechseln und die Abkürzung eingeben:

```
ab <Abkürzung> <Satz oder Wort, das die Abkürzung ersetzt>
```

Um eine Abkürzung zu nutzen, wird im Textmodus die Abkürzung geschrieben, wenn man dann nach der Abkürzung die Leer-oder Entertaste drückt, wird die Abkürzung durch das vorher definierte Wort oder den Satz ersetzt.

Im folgenden Beispiel wird eine Abkürzung definiert die *mfg* durch *Mit freundlichen Grüßen* ersetzt:

```
ab mfg Mit freundlichen Grüßen
```

Danach muss im vi nur noch `mfg` geschrieben werden, wenn man danach noch die Leertaste oder Enter drückt, wird dies ersetzt.

Oft gebrauchte Abkürzungen lassen sich auch in der `.exrc` oder in der `/etc/vi.exrc` dauerhaft speichern.

## 6. tcsh(1)

### 5.4. vi Fenster teilen

[ Allgemein ] by Beat @ 23.08.2006 13:46 CEST

Im vi(1) lässt sich das Fenster teilen, so kann man gleichzeitig mehrere Dokumente bearbeiten. Um das Fenster zu teilen, muss man sich zuerst im Kommandomodus befinden (dies erreicht man, indem man die Esc-Taste drückt) und danach mit dem Doppelpunkt in den ex Modus wechselt und anschliessend E drückt. Nun wird das Fenster geteilt und in der neuen Hälfte das gleiche Dokument geöffnet.

Mit Ctrl + w kann man zwischen den beiden Fensterhälften wechseln. Nun kann man zum Beispiel mit :e <Dateiname> im aktiven Fenster eine neue Datei, mit dem angegebenen Dateinamen, anlegen.

## 6. tcsh(1)

### 6.1. Umlaute in der (t)csh verwenden

[ FreeBSD ] by Beat @ 02.12.2005 15:01 CET

Um Umlaute in einer (t)csh verwenden zu können, muss die LC\_ALL Umgebungsvariable angepasst werden:

```
# setenv LC_ALL de_CH.ISO8859-15
```

Um dies global zu aktivieren, einfach obige Zeile in die /etc/csh.cshrc eintragen.

Mehr zur Benutzung von Umlauten auf FreeBSD hier: <http://user.cs.tu-berlin.de/~eserte/FreeBSD/doc/umlaute/umlaute.html>

### 6.2. Piepston der tcsh deaktivieren

[ FreeBSD ] by Beat @ 13.01.2006 13:59 CET

Um den Piepston der tcsh zu deaktivieren, einfach folgenden Befehl eingeben:

```
# set nobeep
```

Möchte man das Piepsen generell ausschalten, kann man diesen Eintrag in die /.cshrc eines Benutzers oder in die /etc/csh.cshrc für alle Benutzer eintragen.

### 6.3. DEL-Taste in der tcsh benutzen

[ FreeBSD ] by Beat @ 03.04.2006 15:13 CEST

Möcht man in der tcsh die DEL-Taste benutzen können, um Zeichen zu löschen, so kann man folgenden Abschnitt der /etc/csh.cshrc anfügen.

```
# DEL:
bindkey ^[[3~ delete-char
```

Gefunden bei bsdforen.de

## 7. X.Org

### 7.1. Auflösung unter X.Org während des Betriebes ändern

[ Befehl der Woche ] by Beat @ 15.10.2005 02:01 CET

Um die Auflösung unter X.Org während des Betriebes zu ändern, muss zuerst herausgefunden werden, welche Auflösungen verfügbar sind. Dies wird mit folgendem Befehl gemacht:

```
# xrandr -q
```

Danach kann die gewünschte Auflösung gewählt werden:

```
# xrandr -s <SZ Index>
```

## 7. X.Org

<SZ Index> ist der SZ Wert der gewünschten Auflösung. Die aktuelle Auflösung erkennt man durch einen Stern vor dem Index.

Die Frequenz kann mit der -r Option angepasst werden:

```
# xrandr -r <Frequenz>
```

Um die Auflösung auf 1280 x 1024 zu ändern, verwende ich:

```
# xrandr -q
      SZ:      Pixels      Physical      Refresh
*0      1400 x 1050      ( 474mm x 356mm )      *50
  1      1280 x 1024      ( 474mm x 356mm )      50
  2      1024 x 768       ( 474mm x 356mm )      50
  3      800 x 600        ( 474mm x 356mm )      50
  4      640 x 350        ( 474mm x 356mm )      50
  5      640 x 400        ( 474mm x 356mm )      50
  6      720 x 400        ( 474mm x 356mm )      50
  7      640 x 480        ( 474mm x 356mm )      50
  8      1152 x 864       ( 474mm x 356mm )      50
  9      1280 x 960       ( 474mm x 356mm )      50
 10      832 x 624        ( 474mm x 356mm )      50
 11      1152 x 768       ( 474mm x 356mm )      50
Current rotation - normal
Current reflection - none
Rotations possible - normal
Reflections possible - none
# xrandr -s 1
```

Auf einem FreeBSD System befindet sich xrandr im x11/xorg-clients Port.

### 7.2. X.Org: could not open default font "fixed"

[ FreeBSD ] by Beat @ 17.10.2005 16:31 CET

Verweigert X.Org das Starten mit folgender Fehlermeldung

```
Could not open default font 'fixed'
```

so muss unter FreeBSD noch der x11-fonts/xorg-fonts-miscbitmaps Port installiert werden:

```
# cd /usr/ports/x11-fonts/xorg-fonts-miscbitmaps && make install clean
```

### 7.3. Umlaute nach X.Org Update von 6.8.2 auf 6.9.0

[ FreeBSD ] by Beat @ 29.01.2006 12:31 CET

Nach den Update von X.Org auf 6.9.0 wurden beim Schreiben mit einer deutschschweizer Tastatur statt Umlaute irgendwelche Sonderzeichen angezeigt. Um dies zu beheben, einfach in der /etc/X11/xorg.conf in der InputDevice Section folgenden Eintrag

```
Option "XkbLayout" "de_CH"
```

in folgenden ändern:

```
Option "XkbLayout" "ch"
```

Danach den X-Server mit Ctrl + Alt + Backspace neu starten.

### 7.4. Hosts erlauben den X Server zu benutzen

[ Befehl der Woche ] by Beat @ 11.02.2006 11:13 CET

Ist man als ein bestimmter Benutzer in X eingeloggt und möchte als ein anderer Benutzer (z.B. root) eine X Applikation öffnen, so schlägt dies meistens mit folgender Fehlermeldung fehl:

```
# xterm
Xlib: connection to ":0.0" refused by server
Xlib: No protocol specified

xterm Xt error: Can't open display: :0.0
```

## 7. X.Org

Mit dem Befehl `xhost(1)` können nun Verbindungen auf den X Server zugelassen werden.

```
# xhost
access control enabled, only authorized clients can connect
```

Im Moment können keine anderen Rechner oder Benutzer auf den X Server zugreifen. Wollen wir nun allen lokalen Benutzern erlauben auf den X Server zuzugreifen, so kann dies folgendermassen gemacht werden:

```
# xhost localhost
localhost being added to access control list
# xhost
access control enabled, only authorized clients can connect
INET:localhost
INET6:localhost
```

Dies erlaubt allen lokalen Benutzern den X Server zu benutzen und ist deshalb für die meisten Mehrbenutzersysteme nicht geeignet. Will man nur ausgewählten Benutzern die Erlaubnis erteilen, sollte man lieber `xauth` verwenden: <http://www.theparallax.org/dcoul/user2root/xhost.shtml>  
Möchte man nun die Berechtigung der lokalen Benutzer wieder rückgängig machen, so kann dies mit einem Minus vor dem Host gemacht werden:

```
# xhost -localhost
```

So können auch entfernte Rechner autorisiert werden, auf dem Rechner den X Server zu benutzen, indem man diese mittels der IP Adresse freigibt. Aber auch hier ist zu beachten, dass dann alle Benutzer dieses Rechners den X Server benutzen können.

```
# xhost <IP-Adresse>
```

Möchte man aus irgend einem Grund die Zugangskontrolle für den X Server abschalten, so kann man dies so machen:

```
# xhost +
access control disabled, clients can connect from any host
```

Um die Zugangskontrolle wieder zu aktivieren, einfach folgenden Befehl benutzen:

```
# xhost -
access control enabled, only authorized clients can connect
```

## *A. Danksagung*

### **A. Danksagung**

Ganz herzlich danken möchte ich Nina (<http://www.w-h-a-t-e-v-e-r.de>), für ihre Unterstützung, das Korrekturlesen der Artikel und die verschiedenen Graphiken sowie Alain (<http://www.sequenz.ch>), der mir eine Jail für chruetertee.ch zur Verfügung stellt und regelmässig Support für das Flux CMS bietet.

Auch danken möchte ich allen, die regelmässig hier lesen, Kommentare geschrieben oder mich kontaktiert haben, ohne nach Support zu fragen ;-)

### **B. Lizenz**

Creative Commons Deed

Namensnennung 2.5

Sie dürfen:

- den Inhalt vervielfältigen, verbreiten und öffentlich aufführen
- Bearbeitungen anfertigen
- den Inhalt kommerziell nutzen

Zu den folgenden Bedingungen:

- Sie müssen den Namen des Autors/Rechtsinhabers nennen.
- Im Falle einer Verbreitung müssen Sie anderen die Lizenzbedingungen, unter die dieser Inhalt fällt, mitteilen.
- Jede dieser Bedingungen kann nach schriftlicher Einwilligung des Rechtsinhabers aufgehoben werden.

# Stichwortverzeichnis

## A

abcde(1) .....	43
Adblock .....	37
Apache .....	1, 11
apropos(1) .....	33
ath(4) .....	3
Audio CD .....	17, 43

## B

bc(1) .....	41
bg .....	38
BSD Daemon .....	5
bsdtalk .....	32
build(7) .....	7
burncd(8) .....	1
bzcat(1) .....	36
bzgrep(1) .....	36
bzip2(1) .....	36, 42

## C

cadaver(1) .....	40
cal(1) .....	32
camcontrol(8) .....	15
CD-Brenner .....	1
cdcontrol(1) .....	17
chflags(1) .....	30
cleanlinks(1) .....	41
compress(1) .....	42
cp(1) .....	11
cpdup(1) .....	11
CPU throttling .....	2
cpufreq(4) .....	2
CR/LF .....	39
csup(1) .....	15, 19
CVSup(1) .....	10, 15, 19

## D

Dateiflags .....	1, 30
de_CH .....	49
devfs.rules(5) .....	14, 18
DHCP .....	3
dialog(1) .....	20
dos2unix(2) .....	39
du(1) .....	37

## E

E-Mail .....	12, 38
--------------	--------

## F

fastest_cvsup(7) .....	15
fg .....	38
Firefox .....	6, 37
firewall(7) .....	7
Flashplugin .....	6, 22
Flux CMS .....	11
fluxbox .....	22
fstab(5) .....	14
FUSEFS-SSHFS .....	24

## G

gnome_upgrade212.sh .....	4
gnupod .....	15
grep(1) .....	33
gzcat(1) .....	36
gzip(1) .....	36, 42

## H

hier(7) .....	7
hostname(1) .....	18
httpd.conf .....	11

## I

ifconfig(8) .....	3
iftop(8) .....	44
iPod .....	14, 43
ISO-Image .....	1, 9

## J

Jail .....	1, 5, 7, 18, 25
jailaudit .....	5
jls(8) .....	5
jobs .....	38

## K

Kernelmodul .....	2
kill(1) .....	35
kldload(8) .....	2
Konsole .....	18
kqemu-kmod .....	9

## L

last(1) .....	36
LC_ALL .....	48
leave(1) .....	35

## Stichwortverzeichnis

LF ..... 39  
libmap.conf(5) ..... 6  
Linuxpluginwrapper ..... 6  
Live-CD ..... 9  
loader.conf(5) ..... 2, 5, 9  
lock(1) ..... 33  
login.conf(5) ..... 7

### M

mail(1) ..... 38  
make(1) ..... 16  
make.conf(5) ..... 7, 20  
man(1) ..... 7  
Manpage ..... 7  
mesg(1) ..... 34  
MP3 ..... 43  
mySQL ..... 11, 38  
mysqladmin(1) ..... 38  
mysqlcheck ..... 38

### N

ncal(1) ..... 32  
NFS ..... 25  
nologin(5) ..... 25  
NTP ..... 32  
ntp.conf(5) ..... 32  
ntpd(8) ..... 32  
ntpd.conf(5) ..... 32

### O

Ogg ..... 43  
OpenBSD Dateiset ..... 29  
OpenOffice.org ..... 29

### P

Packages ..... 30  
PHP5 ..... 11  
pkg\_add(1) ..... 30  
pkg\_cutleaves(1) ..... 2 f.  
pkg\_deinstall(1) ..... 5  
pkg\_delete(1) ..... 5  
PKG\_PATH ..... 30  
pkg\_tree(7) ..... 21  
pkg\_which(1) ..... 3  
pkgdb(1) ..... 3  
portaudit(1) ..... 12  
portconf ..... 20  
portdowngrade(1) ..... 8  
portmanager ..... 20  
portmaster ..... 20  
Ports ..... 2 f., 5, 12 f., 15, 19, 21

ports(7) ..... 7, 20  
portscan(1) ..... 3, 20  
portsearch ..... 13, 20  
portsnap(8) ..... 10  
portupgrade(1) ..... 3, 5, 20  
portversion(1) ..... 5  
powerd(8) ..... 2

### Q

qemu(1) ..... 9

### R

rc.conf(5) ..... 2 f., 7, 11 f., 14, 18, 24, 32

### S

scp(1) ..... 24  
screen(1) ..... 39  
script(1) ..... 35  
Securelevel ..... 31  
security(7) ..... 7  
security.jail.set\_hostname\_allowed ..... 18  
sendmail(8) ..... 12  
setuid ..... 7  
Sharity Light ..... 31  
shutdown(8) ..... 25  
SMB ..... 24, 31  
snmp ..... 23  
sort(1) ..... 37  
SSH ..... 24  
SSL Zertifikat ..... 1  
ssmtp(8) ..... 12  
style(9) ..... 7  
sysctl(8) ..... 2, 18  
sysctl.conf(5) ..... 14, 24  
systat(1) ..... 6

### T

tar(1) ..... 42 f.  
tcsh(1) ..... 48  
top(1) ..... 44  
tuning(7) ..... 7

### U

unionfs ..... 25  
Update  
    FreeBSD Basissystem ..... 16  
    FreeBSD Kernel ..... 16  
    OpenBSD 3.7 auf 3.8 ..... 28  
usermount ..... 14  
users(1) ..... 36  
uuencode(1) ..... 38



**V**

vi(1)	
ab.....	47
Fenster teilen.....	48
Kurzreferenz .....	44
map.....	46
vidcontrol(1).....	18

**W**

w(1).....	23, 34, 36
wall(1).....	34
watch(8).....	23
WebDAV .....	40
WEP.....	3, 20
which(1) .....	3
who(1).....	36
WLAN.....	3, 20
wlan(4) .....	3
wpa_supPLICant(8).....	20
wpa_supPLICant.conf(5).....	20

**X**

X.Org .....	22, 48 f.
xhost(1) .....	50
Xinerama.....	22
xlock(1) .....	34
xorg.conf .....	22
xrandr(1) .....	48
xscreensaver .....	34

**Z**

zip(1).....	43
-------------	----