

FreeBSD-10.3-Installation auf meinem Desktop-Rechner

ein Installationsprotokoll

0 Einleitung

Dies ist ein ausführliches Protokoll zur Installation von FreeBSD-10.3 auf meinem Desktop-Rechner.¹ Die meisten Angaben in den Konfigurationsdateien beziehen sich auf meinen PC und unser Netzwerk, ein Übertragen auf andere PC Situationen und Nutzerbedürfnisse sollte aber nicht allzu schwierig sein.

Dies ist kein Ersatz für das FreeBSD Handbuch, und bietet vermutlich FreeBSD-Profiles kaum neue Erkenntnisse. Da ich selbst noch Anfänger bin – wenn auch ein schon etwas fortgeschrittenerer, bitte ich um Nachsicht und auch Korrektur, falls hier einige Sachen nicht optimal oder gar fehlerhaft umgesetzt sind.

Folgende Kürzel und Layouts verwende ich:

Anm.	Anmerkung	<code>portsnap</code> <code>fetch</code>	Befehlseingaben
DebGL	Debian GNU/Linux	<code>/etc/rc.conf</code>	für Pfade
DLQ	Download-Quelle	<code>#!/bin/sh</code>	für Code
eHB	englisches Handbuch FreeBSD	<i>Achtung ...</i>	wichtige Hinweise
dHB	deutsches Handbuch FreeBSD	<i>Exkurs ...</i>	für Exkurse
Ref	Referenz/Verweis auf	Der PC soll ...	normaler Text
Vgl.	Vergleich	@Name	Name Forenuser

a Die Hardware²

Machine: Mobo: MSI // model: H97M ECO (MS-7817) // v: 5.0 //
Bios: American Megatrends // v: V26.5 // date: 12/23/2014

CPU: Quad core Intel Core i5-4690 (-MCP-) // cache: 6144 Kbclock //
speeds: max: 3900 MHz

Graphics: Card: NVIDIA GK208 [GeForce GT 730]

Audio: Card-1 VIA ICE1712 [Envy24] PCI Multi-Channel I/O Controller //
driver: snd_ice1712

Network: Card: Intel Ethernet Connection (2) I218-V // driver: e1000e //
F: eth0 // speed: 1000 Mbps // duplex: full // mac: xxxxxxxxxxxxxx

Drives: HDD Total Size: 376.1GB // ID-1: CT250BX100SSD1 size: 250.1GB //
ID-2: SanDisk_SDSSDP12 size: 126.0GB

¹ Anm.: Wenn erforderlich, gebe ich weitere Quellen als Referenzen an, z.B. das FreeBSD Handbuch (englische Fassung), Manuals z.B. von Warren Block, von @Rakor, oder verweise auf Diskussionen in den <https://www.bsdforen.de> oder <https://forums.freebsd.org/>

² Anm.: Auszug der Ausgabe von `inxi -F` unter DebGL

b Anforderungen

Der Rechner soll als Desktop-System in unser Netzwerk mit unixoiden Systemen (Clients und Server) integriert werden. Auf jedem der Clients gibt es neben dem Root Account die Konten holger und diane.

Software: Desktop-Environment (kde4), Dateimanagement (konsole / dolphin), Office (libreoffice), Internet (claws-mail, firefox), Audio (abcde, audacity, cdrtools, soundkonverter, vlc), Video (vlc, handbrake), Zugriff auf nfs Shares, Remote-Printing, Zugriff auf andere Rechner per ssh und weiteres

Als Experiment mache ich eine Parallel-Installation, basierend auf FreeBSD-11 Alpha3, dort soll ein aktueller Snapshot von plasma5 aus den inoffiziellen Area51 Ports gebaut werden.³

c Kurze Angaben zum Netzwerk

Alle Maschinen erhalten per DHCP ihre Adressen vom Router, jeder MAC ist im Router eine feste IP zugeordnet, damit gewährleistet ist, dass die verschiedenen Clients von Router immer die gleichen Adressen zugewiesen bekommen (Alternativ ist auch eine Deaktivierung von DHCP und statische Netzwerk-Konfiguration möglich).

```
taschenmaus.unix.pc mit 192.168.1.20 (Fileserver DebGL)
wdnas.unix.pc mit 192.168.1.10 (Backup-Server DebGL)
farbmaus.unix.pc mit 192.168.1.30 (Printserver turboprint DebGL)
biber.unix.pc mit 192.168.1.121 (um diesen PC geht es hier)
gerbil.unix.pc mit 192.168.1.111 (PC meiner Frau)
```

... und weitere

d Installationsmedium

Ich habe mich für ein bootonly Installationsmedium entschieden, laut FreeBSD Handbuch muss so ein Medium auf CD gebrannt werden⁴

Die ISO-Datei **FreeBSD-10.3-STABLE-amd64-20160518-r300092-bootonly**⁵

wird auf einen CD-Rohling geschrieben, von dieser CD wird der Rechner dann gestartet.

1 Booten des Installationsmediums

Der Rechner wird mit dem Installationsmedium gestartet, am Bootprompt wird mit <RETURN> FreeBSD im Multiuser-Modus gebootet.

Nach dem Booten kann gewählt werden zwischen <Install>, <Shell> und <LiveCD>, gewählt wird die Shell.

³ Anm.: Dieses Experiment erhält ein eigenes Kapitel.

⁴ Vgl. eHB https://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/bsdinstall-pre.html und dort das Unterkapitel **2.3.1. Prepare the Installation Media**

⁵ DLQ: <ftp://ftp.freebsd.org/pub/FreeBSD/snapshots/amd64/amd64/ISO-IMAGES/10.3/>

Um deutsches Tastaturlayout zu bekommen, wird **kbdmap** aufgerufen und folgendes Layout ausgewählt:

German ISO-8859-15 accent.keys

Mit dem Bestätigen steht deutsches Tastaturlayout sofort zur Verfügung und die Festplatte kann manuell partitioniert werden.

2 Partitionierung einer 128 GB großen SSD Platte⁶

Achtung: Beim Arbeiten mit gpart genau auf die Eingaben achten, sonst droht ggf. ungewollter Datenverlust

Auf die Platte soll ein FreeBSD-10.3 mit dem kde4-Desktopenvironment installiert werden (vgl. auch Kapitel **0 b Anforderungen**) und es soll noch Platz bleiben für eine experimentelle Installation von FreeBSD-11 mit plasma5.

Es handelt sich in meinem PC um die zweite Platte, also um **/dev/ada1** (FreeBSD) bzw. **/dev/sdb** (GNU/Linux), die erste Platte wird unter FreeBSD mit **/dev/ada0**, unter Linux mit **/dev/sda** bezeichnet.

Zunächst wird eine eventuell vorhandene alte Tabelle gelöscht:

```
gpart destroy -F ada1
```

Dann wird eine neue gpt Tabelle angelegt:

```
gpart create -s gpt ada1
```

Für den Bootcode wird nun eine 512 kb kleine 4K ausgerichtete Partition angelegt, sie erhält das Label **ssdboot**, und dann wird der Bootcode für FreeBSD dorthin geschrieben:

```
gpart add -t freebsd-boot -s 512k -a4k -l ssdboot ada1
gpart bootcode -b /boot/pmbr -p /boot/gptboot -i1 ada1
```

Was das weitere Partitionieren angeht, gibt es verschiedene Ansichten, möglich ist zum Beispiel ein Schema

boot + swap + Rest für Wurzel oder boot + swap + Wurzel + /var + /usr/ + /usr/home

Ich habe mich für folgendes Schema entschieden boot + Wurzel + /var + /usr + swapfile.⁷

Demgemäß folgen drei Partitionen vom Typ **freebsd-ufs** mit 1M Ausrichtung, die Wurzelpartition und /var erhalten jeweils 5 GB, /usr erhält 45 GB:

```
gpart add -t freebsd-ufs -l ssdrootfs -b 1m -s 5g ada1
gpart add -t freebsd-ufs -l ssdvarfs -a 1m -s 5g ada1
gpart add -t freebsd-ufs -l ssdusrfs -a 1m -s 45g ada1
```

Die Partitionen werden mit **ufs** formatiert, Trim (-t) wird eingeschaltet, Softupdates (-U)

⁶ Ref.: Siehe zur SSD Partitionierung auch das meines Erachtens ausgezeichnete Howto von Warren Block <http://www.wonkity.com/~wblock/docs/html/ssd.html>

⁷ Anm.: Mit **boot** ist hier die schon angelegte kleine Bootcode-Partition gemeint, mit dem Verzeichnis **/boot** hat das nichts zu tun.

werden eingeschaltet, aber Journaling der Softupdates (-SUJ) wird nicht aktiviert:

```
newfs -U -t /dev/gpt/ssdrootfs
newfs -U -t /dev/gpt/ssdvarfs
newfs -U -t /dev/gpt/ssdusrfs
```

Die Shell wird mit **exit** beendet.

3 Installation des Basis-Systems

Nach dieser Prozedur wird der FreeBSD Installer über die Auswahl <Install> gestartet.

a Auswahl der Tastatur

Das deutsche Tastaturlayout sollte noch geladen sein, zur Vorsicht wähle ich nochmals explizit **german.iso.acc.kbd** aus (beim Testen fällt vielleicht auf, dass äöü nicht korrekt angezeigt werden, aber das ist dieser Stelle nicht weiter schlimm, wichtig sind /, z, y @

b Vergabe des Rechner-Namens

Da alle Rechner in unserem Heimnetzwerk die Domain **.unix.pc** haben und ich gerne die Rechner nach Nagetieren benenne, verwende ich hier den Namen **biber.unix.pc**

c Weitere Komponenten für die Installation auswählen

Ich wähle hier <lib32> <doc> und <src>⁸

d Netzwerkkonfiguration

Da die bootonly CD keine weiteren Pakete enthält, muss für die bevorstehende Installation nun das Netzwerk konfiguriert werden. Der Installer bietet bei meinem Rechner die Konfiguration der Netzwerkkarte **em0 Intel PRO/1000 Network Connection** an.

Ich wähle IPv4 und DHCP⁹ aus, IPv6 aktiviere ich nicht.

In der folgenden Box steht dann

```
search: fritz.box
IPv4 DNS #1 192.168.1.1
IPv4 DNS #2 hier kann noch ein DNS eingefügt werden, z.B. 8.8.8.8
```

Nach Bestätigung mit <OK> geht es dann an die Wahl des Mirrors, ich nehme einen deutschen: <ftp://ftp.de.freebsd.org>

Mit <OK> gelangt man dann zum letzten Punkt vor der eigentlichen Installation, der

⁸ Anm.: Die Ports werden später durch **portsnap fetch extract** geholt.

⁹ Anm: Mein Router weist den Rechnern in unserem Netzwerk MAC-bezogen jeweils die gleichen IP Adressen zu, daher kann ich DHCP wählen, eine Alternative ist die Deaktivierung von DHCP beim Router und eine komplett manuelle Konfiguration des Netzwerkes, dies ist zu wählen, wenn der Router oben genanntes Feature nicht beherrscht und ein Netzwerk mit mehreren Rechnern aufgebaut werden soll, in welchem ssh und nfs zum Einsatz kommen sollen.

Partitionierung.

e Partitionieren

Zur Auswahl stehen <Auto>, <Manual>, <Shell> und <zfs>. Da die Platte schon per Konsole vorbereitet wurde, wähle ich die manuelle Partitionierung, navigiere zu ada1p2, gehe auf **<Modify>**, weise der Partition den Mountpoint **/** zu und bestätige mit **<OK>**. Die Partition ada1p3 setze ich auf **/var** und ada1p4 auf **/usr**. Mit **<Finnish>** und **<Commit>** kommt man zu einem Dialog, der darauf hinweist, dass ein schon bestehendes FreeBSD über ein Update und nicht über eine Neuinstallation aktualisiert werden soll, in solch einem Fall wird der Vorgang mit **<No>** abgebrochen, da hier FreeBSD neu installiert werden soll, wird mit **<Yes>** die Installation gestartet.

f Post-Installation

Nach der Installation des Basis-Systems vergebe ich ein Passwort für root, weitere Nutzer-Accounts füge ich später manuell hinzu. Nach Auswahl der Zeitzone (Timezone **UTC** → Region **Europe** → Country **Germany** → German Time Zones **most Locations**) und der Auswahl der zusätzlichen Services wie **ssh**, **powerd** und **dumpdev** ist die Basis-Installation beendet, nach einem Neustart geht es dann weiter mit der Einrichtung des Systems.

– Exkurs 1: Hinzufügen der FreeBSD-Installation in Grub –

Bei der oben erwähnten Partitionierung mit ada1p1 als freebsd-boot kann auf meinem PC das FreeBSD über die Boot-Device Liste (Start von der zweiten Festplatte) gebootet werden. Auf meinem PC ist auf der ersten Festplatte allerdings ein GNU/Linux installiert, und zwar das auf Debian Jessie basierende MX-Linux.¹⁰ Es folgt nun eine Anleitung zur Integration des installierten FreeBSD in das Grub-Menü der GNU/Linux Installation (bei dieser Startmethode könnte sogar die freebsd-boot Partition entfallen, sie ist meines Erachtens dennoch ratsam, denn geht mal was am Grub der GNU/Linux Installation kaputt, kann FreeBSD immer noch über die Boot-Device Liste gestartet werden).

*Bei den Vorbereitungen der Installation von MX-Linux auf der ersten Festplatte **/dev/sda** mit GPT als Partitionstabelle habe ich für die Grub-Installation an das Ende der Platte eine Partition erzeugt:¹¹*

```
gdisk /dev/sda
Command (? for help): n
Partition number (1-3): 128
First sector (xx-yyyymm, default = xxx) or {+-}size{KMGTP}: -3M
..
..
```

¹⁰ Anm.: MX-Linux, ein Gemeinschaftsprojekt der Entwickler von antix und der Mepis-Gemeinschaft, setzt auf das alte sysVinit System, systemd kann aber bei Bedarf nachgerüstet werden.

¹¹ Vgl. Arch-Wiki: <https://wiki.archlinux.de/title/GPT#Grub-Bootloader-Partition>

```
Last sector xx-yy,default = Sektorende) or {+-}size{KMGTP}: enter
Hex code or GUID (L to show codes, Enter = 8300): ef02
Changed type of partition to 'Bios boot partition'
Command (? for help): w
```

Bei der anschließenden Installation wird für Grub nun Partition Nr. 128 am Ende der SSD genutzt. Um dann FreeBSD dem Grub-Menü hinzu zu fügen, sind folgende Schritte erforderlich.

Zunächst muss der Datei `/etc/grub.d/40_custom` folgendes hinzu gefügt werden:¹²

```
menuentry "FreeBSD - Unix"{
    insmod ufs2
    set root='(hd1,2)'
    kfreebsd /boot/loader
}
```

Danach wird `update-grub` aufgerufen, dies gilt für jede Debian basierte Distribution, unter anderen GNU/Linux Distributionen `grub-mkconfig -o /boot/grub/grub.cfg`

Nach einem Neustart sollte im Grub-Menü FreeBSD zur Auswahl stehen.

– Ende Exkurs 1 –

4 Erzeugen einer Swap-Datei und Anpassen der `/etc/fstab`

Eine Swap-Datei von 4 GB Größe wird erzeugt:

```
mkdir /usr/swap
dd if=/dev/zero of=/usr/swap/swap bs=128k count=32768
```

Die `/etc/fstab` sieht dann wie folgt aus:

<code>/dev/gpt/ssdrootfs</code>	<code>/</code>	<code>ufs</code>	<code>rw</code>	<code>1</code>	<code>1</code>
<code>/dev/gpt/ssdvarfs</code>	<code>/var</code>	<code>ufs</code>	<code>rw</code>	<code>2</code>	<code>2</code>
<code>tmpfs</code>	<code>/tmp</code>	<code>tmpfs</code>	<code>rw,mode=01777</code>	<code>0</code>	<code>0</code>
<code>/dev/gpt/ssdusrfs</code>	<code>/usr</code>	<code>ufs</code>	<code>rw</code>	<code>2</code>	<code>2</code>
<code>md99</code>	<code>none</code>	<code>swap</code>	<code>sw,file=/usr/swap/swap,late</code>	<code>0</code>	<code>0</code>
<code>proc</code>	<code>/proc</code>	<code>procfs</code>	<code>rw</code>	<code>0</code>	<code>0</code>

Der proc Eintrag ist für den Betrieb von kde4 erforderlich.

5 Einrichten von Gruppen und Nutzern

Auf dem FreeBSD System sollen die Nutzerkonten **diane** und **holger** angelegt werden, wie auf den anderen PCs im Heimnetzwerk soll das Konto **diane** die ID **1001**, das Konto **holger** die ID **1000** erhalten, die Gruppe **users** mit **GID 5000** soll die Primärgruppe beider Nutzerkonten sein.

Zunächst wird die Gruppe users mit der GID 5000 angelegt:

```
pw groupadd users -g 5000
```

¹² Anm.: Bei Grub beginnt die Festplattenzählung mit 0, die Partitionszählung mit 1.

Dann folgt das Konto für holger:

```
adduser -g users -uid 1000
```

Den Schritten folgen und mit Return bestätigen, neben der Gruppe **users** das Konto noch zur Gruppe **wheel** und **operator** hinzufügen:

```
Invite user to further groups: wheel operator
```

Danach wird das Konto für diane angelegt und auch den Gruppen **wheel** und **operator** hinzugefügt:

```
adduser -g users -uid 1001
```

6 Grundkonfiguration Soundkarte¹³

Zunächst wird das Modul für meine M-Delta Audiophile in die **/boot/loader.conf** eingetragen:

```
snd_envy24_load="YES"
```

Nach einem Neustart wird eventuell noch die Soundkarte als primäres Device gesetzt, falls bei folgendem Befehl die M-Delta Audiophile nicht als pcm0 angezeigt wird:

```
dmesg|grep pcm
```

Ist zum Beispiel die Karte als pcm3 angegeben, muss folgendes in die **/etc/sysctl.conf**:

```
hw.snd.default_unit=414
```

Nach einem Neustart sollte nun die Karte einsatzbereit sein.

7 Grundkonfiguration locale-Einstellungen

Zwei Dateien sind bei jedem Account zu bearbeiten, um utf-8 und deutsch zu haben:

Die **~/profile** wird um folgende Zeiler ergänzt:

```
LANG=de_DE.UTF-8; export LANG
```

Die **~/login_conf** wird wie folgt editiert:

```
me:\n:charset=UTF-8:\n:lang=de_DE.UTF-8:\n:tc=default:
```

Nach einem erneuten Login zeigt der Befehl **locale de_DE.UTF-8** an, falls nicht, bitte die **~/profile** und **~/login_conf** auf Schreibfehler überprüfen und den Rechner ggf. komplett

¹³ Vgl. eHB <https://www.freebsd.org/doc/handbook/sound-setup.html>

¹⁴ Anm.: Es ist auch möglich, zunächst per **kldload snd_envy24** den Treiber testweise zu laden, allerdings kann dann ein Aufruf von **dmesg|grep pcm** die Soundkarte mit anderer pcm-Nummer anzeigen, als wenn sie in **/boot/loader.conf** eingetragen wird und dann ein Neustart durchgeführt wird. *Daher Achtung: Bitte erst **hw.snd.default_unit** ggf. anpassen nach Moduleintrag und Reboot.*

neu starten.¹⁵

8 Vorbereitungen der Ports

Bei allen Ports verwende ich meistens die Default-Optionen, wenn ich davon abweiche, erläutere ich das hier.

a Holen und entpacken der Ports¹⁶

```
portsnap fetch extract
```

b Installation der bash

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

Nach der Installation der bash muss folgender Eintrag der **/etc/fstab** hinzu gefügt werden:

```
fdesc /dev/fd    fdescfs    rw    0    0
```

Um die Shell jedes Nutzers und auch von root zu ändern bitte folgendes unter dem zu ändernden Account eingeben, den Anweisungen folgen und sich dann ab- und wieder anmelden:

```
chsh -s /usr/local/bin/bash
```

c Bauen von portmaster, bei den Optionen auf Kompatibilität mit der bash achten

```
cd /usr/ports/ports-mgmt/portmaster && make install clean
```

d Baunen des Editors nano

```
portmaster -d editors/nano
```

e Weitere Werkzeuge zum Arbeiten mit den Ports

Um noch ein paar nützliche Werkzeuge zur Port-Verwaltung zu bekommen, installiere ich noch einen Metaport:

```
portmaster -d ports-mgmt/port-maintenance-tools17
```

9 Grundkonfiguration Zugriffsrechte auf Devices

Da auch schon an der Konsole ohne installierte grafische Oberfläche einige sehr schöne Sachen gemacht werden können, zum Beispiel Rippen und Brennen von CDs, füge ich noch vor der Installation von Xorg und kde4 dieses Kapitel ein.

¹⁵ Vgl. eHB <https://www.freebsd.org/doc/handbook/using-localization.html>

¹⁶ **portsnap** ist ein Standardwerkzeug zum Aktuell halten der Ports, wichtige Optionen sind **fetch** (holt einen Tarball der aktuellen Ports), **extract** (entpackt den Tarball nach /usr/ports) und **update** (aktualisiert den Portstree /usr/ports). Zu **portsnap** und weiteren Werkzeugen zum Arbeiten mit den Ports vgl. auch <https://www.freebsd.org/doc/handbook/ports-using.html>

¹⁷ Zum Inhalt des Metaports vgl. <http://www.freshports.org/ports-mgmt/port-maintenance-tools/>

Standardmäßig hat unter FreeBSD nur root Zugriff auf optische Devices und auf USB-Devices (als root etwa mit **abcde** CDs zu rippen ist jedoch schon deshalb heikel, weil **abcde** über das Internet nach Tags sucht). Um dies zu ändern, und auch normalen Nutzern den Zugriff auf optische Laufwerke und USB-Devices zu erlauben, müssen in der Datei **/etc/devfs.conf** einige Einträge hinzu gefügt werden.

Zunächst prüfe ich mit **dmesg|grep da** die Anzahl und Bezeichnung der USB-Devices, bei mir sind es **da0**, **da1**, **da2** und **da3**.

Dementsprechend kommt in die **/etc/devfs.conf**

```
perm    /dev/da0      0666
perm    /dev/da1      0666
perm    /dev/da2      0666
perm    /dev/da3      0666
```

Mit **dmesg|grep cd** überprüfe ich das optische Laufwerk, da ich nur eines in meinem PC habe, taucht in der Ausgabe auch nur **cd0** auf, in die **/etc/devfs.conf** kommt

```
perm    /dev/cd0      0666
```

Nun kann es ja sein, dass nach dem Booten diverse externe Geräte erst angestöpselt werden. Um da eine Rundum-Sorglos-Konfiguration¹⁸ zu bekommen, kommt in die **/etc/devfs.rules** folgendes

```
[devfsrules_common=7]
add path 'ad[0-9]\*'    mode 0666
add path 'ada[0-9]\*'   mode 0666
add path 'da[0-9]\*'    mode 0666
add path 'acd[0-9]\*'   mode 0666
add path 'cd[0-9]\*'    mode 0666
add path 'mmcscd[0-9]\*' mode 0666
add path 'pass[0-9]\*'  mode 0666
add path 'xpt[0-9]\*'   mode 0666
add path 'ugen[0-9]\*'  mode 0666
add path 'usbctl'       mode 0666
add path 'usb/\*'       mode 0666
add path 'lpt[0-9]\*'   mode 0666
add path 'ulpt[0-9]\*'  mode 0666
add path 'unlpt[0-9]\*' mode 0666
add path 'fd[0-9]\*'    mode 0666
add path 'uscan[0-9]\*' mode 0666
add path 'video[0-9]\*' mode 0666
add path 'tuner[0-9]*'  mode 0666
add path 'dvh/\*'       mode 0666
add path 'cx88*'        mode 0660
add path 'cx23885*'     mode 0660
add path 'iicdev*'      mode 0660
add path 'uvisor[0-9]*' mode 0660
```

Der neue Satz an Regeln muss in der **/etc/rc.conf** noch aktiviert werden:

18 Vgl. dazu das meines Erachtens gute Howto von Nicole zu FreeBSD auf dem Desktop:
<https://cooltrainer.org/a-freebsd-desktop-howto/#device-permissions>

```
devfs_system_ruleset="devfsrules_common"
```

10 Einstellungen zur Integration in unser Netzwerk

Wie schon geschrieben soll dieser PC in ein Netzwerk integriert werden, er soll über **ssh** andere PCs erreichen und per **ssh** auch erreichbar sein, außerdem soll Remote-Zugriff auf unseren Printserver möglich sein, und dem PC werden **nfs**-Freigaben von unserem Datenserver zur Verfügung gestellt.

Zunächst wird die **/etc/hosts** um folgende Einträge¹⁹ ergänzt:

192.168.1.20	taschenmaus.unix.pc	taschenmaus
192.168.1.30	farbmaus.unix.pc	farbmaus
192.168.1.111	gerbil.unix.pc	gerbil
192.168.1.121	biber.unix.pc	biber
192.168.1.131	hamster.unix.pc	hamster

Auf **taschenmaus** (mit Debian GNU/Linux) ist Desktop **biber.unix.pc** für den **nfs** Zugriff in den **/etc/exports** frei gegeben (und in die **/etc/hosts.allow** eingetragen):

```
### biber mit 192.168.1.121
#
# gemeinsame Daten biber
/daten/share 192.168.1.121(rw,sync,subtree_check)
# daten holger biber
/daten/nutzer/holger
192.168.1.121(rw,sync,subtree_check,anonuid=1000,anongid=5000)
# daten diane biber
/daten/nutzer/diane
192.168.1.121(rw,sync,subtree_check,anonuid=1001,anongid=5000)
# daten pc biber
/daten/pc 192.168.1.121(rw,sync,subtree_check,no_root_squash)
```

Nun sind nur noch folgende Einträge in der **/etc/fstab** von **biber** nötig, die Mountordner müssen natürlich auch angelegt werden:

taschenmaus.unix.pc:/daten/share	/nas/share	nfs	rw	0	0
taschenmaus.unix.pc:/daten/pc	/nas/pc	nfs	rw	0	0
taschenmaus.unix.pc:/daten/nutzer/holger	/usr/home/holger/nas	nfs	rw	0	0
taschenmaus.unix.pc:/daten/nutzer/diane	/usr/home/diane/nas	nfs	rw	0	0

In der **/etc/rc.conf** wird dann **nfs** Zugriff aktiviert:

```
nfs_client_enable="YES"
rpc_statd_enable="YES"
rpc_lockd_enable="YES"
```

mount -a sollte nun die **nfs**-Freigaben mounten, und danach sollte **mount|grep nfs** folgendes ausgeben:

```
taschenmaus.unix.pc:/daten/share on /nas/share (nfs)
taschenmaus.unix.pc:/daten/pc on /nas/pc (nfs)
taschenmaus.unix.pc:/daten/nutzer/holger on /usr/home/holger/nas (nfs)
taschenmaus.unix.pc:/daten/nutzer/diane on /usr/home/diane/nas (nfs)
```

¹⁹ Zur Erklärung der Hostnamen: taschenmaus (Dateiserver), farbmaus (Printserver), biber (dieser Desktop), gerbil (Desktop meiner Frau), hamster (Testrechner für DragonFlyBSD)

11 Diverse Werkzeuge für die Konsole

Auch ohne eine grafische Oberfläche lassen sich zahlreiche Dinge unter FreeBSD erledigen, die nichts mit der Systemadministration zu tun haben, ich gehe im folgenden auf einige Dinge ein, die ich selber gerne benutze.²⁰

a Audio

Das sogenannte Schweizer Taschenmesser an der Konsole zum Bearbeiten von Audio-Dateien ist das Werkzeug **sox**, es lässt sich einfach aus den Ports bauen, bei den Optionen sind die jeweils favorisierten Audioformate zu berücksichtigen, mir ist eine Unterstützung von flac sehr wichtig:

```
portmaster -d audio/sox
```

Ein sehr schönes Werkzeug zum Rippen von CDs ist **abcde**, zum Brennen die **cdrtools** von Jörg Schilling²¹

```
portmaster -d audio/abcde
portmaster -d audio/cdrtools
portmaster -d audio/cdparanoia
```

Damit der normale Nutzer mit diesen Werkzeugen arbeiten kann, sind einige Konfigurationsschritte notwendig, die in einem vorherigen Kapitel schon abgearbeitet wurden.

... noch weitere Punkte hinzufügen ...

12 Xorg & Grafikkarte

Zunächst wird **xorg** aus den Ports gebaut (bei Bedarf werden weitere Ports hinzu gefügt): Ich bestätige hier auch die Default-Optionen:

```
portmaster -d x11/xorg
```

Um Schriften zu haben, die Arial, Times New Roman und Courier entsprechen, installiere ich noch die liberation-fonts:

```
portmaster -d x11-fonts/liberation-fonts-ttf
```

Dann wird der Nvidia-Treiber installiert²², vor der Installation ist es wichtig, das Modul **linux** zu laden, weil sonst die Installation mit einem Fehler abbricht:

```
kldload linux && pkg install nvidia-driver
```

Danach werden folgende Einträge der **/boot/loader.conf** hinzugefügt:

```
nvidia_load="YES"
linux_load="YES"
```

Um den Xserver für die Nvidia-Karte zu konfigurieren, sind folgende Schritte erforderlich:

²⁰ Anm.: Was Details zu den Werkzeugen betrifft, empfehle ich die einzelnen Manpages.

²¹ Vgl. <http://cdrtools.sourceforge.net/private/cdrecord.html>

²² Anm. vom 01.06.2016: Ein Bau aus den Ports ist zur Zeit nicht möglich

```
mkdir -p /usr/local/etc/X11/xorg.conf.d
nano -w /usr/local/etc/X11/xorg.conf.d/10-nvidia.conf
```

Der Inhalt der **/usr/local/etc/X11/xorg.conf.d/10-nvidia.conf**

```
Section "Device"
    Identifier "NVIDIA Card"
    VendorName "NVIDIA Corporation"
    Driver "nvidia"
EndSection
```

Nach einem Neustart ist der Xserver mit der Nvidia-Karte einsatzbereit und kann über **startx** angetestet werden.

13 Installation von kde4

Um ein komplettes kde4 zu bekommen, installiere ich den Metaport und für deutsch die entsprechende Lokalisierung:

```
portmaster -d x11/kde4
portmaster -d german/kde4-l10n
```

Bei der Vorkonfigurierung belasse ich es bis auf eine Ausnahme²³ bei den Default-Optionen.

Da ich auch gtk Anwendungen unter kde4 nutzen werden (Firefox, Audacity, Claws-Mail), benötige ich auch noch die gtk Konfiguration unter den Systemsettings:

```
portmaster -d x11-themes/kde-gtk-config
```

Der Metaport **kde4** zieht auch **cups** mit, in die **/etc/rc.conf** müssen folgende Einträge für **kde4** und **cups**:

```
dbus_enable="YES"
hald_enable="YES"
kdm4_enable="YES"
cupsd_enable="YES"
cups_browsed_enable="YES"
```

Der cups_browsed-Eintrag ist wichtig für den Remote-Zugriff auf den Printserver **farbmaus** und um diesen Printserver dem cups auf **biber** noch „bekannt“ zu machen, muss eine Datei **/usr/local/etc/cups/client.conf** mit folgendem Inhalt angelegt werden:

```
ServerName 192.168.1.30:631
```

Ein Neustart des Rechner oder aber der Services **cupsd** und **cups_browsed** aktiviert dann den Remote-Zugriff auf den Printserver und die dort frei gegebenen Drucker.

14 Bereinigung der **/usr/ports**

Nach dem Bau des Xservers und des großen Brockens kde4 hat sich mittlerweile bei den

²³ Für das Paket vlc-qt4 benötige ich **upnp**, weil in unserem Netzwerk über **minidlna** Multimediadateien allen Clients zur Verfügung gestellt werden. Im Gegensatz zu vielen GNU/Linux Distributionen ist unter FreeBSD upnp in vlc standardmäßig nicht einkompiliert.

Ports einiges an zusätzlichen Dateien angesammelt, das Verzeichnis **/usr/ports** dürfte mittlerweile einige Gigabyte umfassen, dies kann überprüft werden mit:

```
du -h /usr/ports
```

Um aufzuräumen, kann man das Werkzeug **portsclean**²⁴ verwenden:

```
portsclean -CDD
```

Allerdings wird hierbei nicht der umfangreiche Ordner **/usr/ports/distfiles** entfernt. Dies ist ungefährlich für die Ports-Konsistenz, ein kurzes Skript **/usr/local/bin/portputz** mit folgendem Inhalt macht gründlich sauber:

```
#!/bin/sh
du -h /usr/ports
echo Umfang des Verzeichnisses /usr/ports
sleep 3
portsclean -CDD
rm -rf /usr/ports/distfiles/*
du -h /usr/ports
echo Ports sind bereinigt
sleep 3
return 0
```

Skript für Root ausführbar machen, aufrufen und sich danach über mehr Platz freuen.

15 Office

Der Bau von Libreoffice aus den Ports dauert eine Weile, in der Zeit kann man eine gemütliche Kaffeepause einlegen:

```
portmaster -d german/libreoffice
```

Bei den Optionen gtk2-Unterstützung abwählen, dafür kde4- und Media-Unterstützung²⁵ auswählen.

16 Mail

Da ich das Kdepim-Paket aus kde4 nicht mag, und mir auch Thunderbird zu aufgeblasen ist, nutze ich schon seit einiger Zeit claws-mail. Diese schlanker Mail-Anwendung ist zur Zeit wegen eines Checksum Mismatches bei einem Tex Paket nicht aus den Ports installierbar²⁶, daher benutze ich hierfür ausnahmsweise den Paketmanager:

```
pkg install claws-mail claws-archiver claws-pdf_viewer  
claws-address_keeper
```

24 Dieses Werkzeug wurde durch den Bau des Metaports **ports-maintenance-tools** mitinstalliert., siehe auch die Manpage für verschiedene Optionen.

25 Die Media-Unterstützung bietet Audioausgabe bei entsprechenden Impress-Dateien mit integrierter Musik.

26 Der Mismatch trat am 07.06.2016 auf.

17 Browser

Für die Installation von **Firefox** samt deutscher Lokalisierung sind folgende Schritte erforderlich:

```
portmaster -d www/firefox-esr
portmaster -d www/firefox-esr-i18n
```

In den Browser-Einstellungen muss zunächst **german localisation** auf **enable** gesetzt werden. In der Adresszeile **about:config** eingeben und den Wert von **general.useragent.locale** auf **de** setzen. Nach einem Firefox-Neustart ist dieser in deutsch.

18 Weitere Anwendungen

```
portmaster -d audio/audacity
portmaster -d audio/easytag
portmaster -d audio/soundkonverter27
portmaster -d multimedia/handbrake
```

²⁷ Achtung: Das Einlesen von CDs unter kde4 Anwendungen wie k3b oder soundkonverter führt seit einiger Zeit zum Absturz, ein Fehlerbericht ist in Planung

19 Experimentell: plasma5

Zunächst wird nach der bisherigen Anleitung inklusive des 12. Kapitels, allerdings mit dem aktuellen FreeBSD 11 Alpha3²⁸, ein System samt Xserver aufgesetzt.

Achtung: Sollte diese Installation parallel zu einer FreeBSD-Installation erfolgen, müssen einige Angaben bei der Partitionierung entsprechend angepasst und auch andere Label vergeben werden: Zum Beispiel für das Wurzelverzeichnis `testrootfs`, für das Var-Verzeichnis `testvarfs` und für das Usr-Verzeichnis `testusrfs`.

Hier zunächst nochmal die bisherige Partitionierung meiner zweiten SSD `/dev/ada1`:

```
gpart -p -l show ada1
```

```
=>      34  246162605   ada1  GPT  (117G)
      34         6      - free -  (3.0K)
      40       1024  ada1p1  ssdboot (512K)
     1064       984      - free -  (492K)
     2048    10485760  ada1p2  ssdrootfs (5.0G)
    10487808    10485760  ada1p3  ssdvarfs (5.0G)
    20973568    94371840  ada1p4  ssdusrfs (45G)
    115345408   130817231      - free -  (62G)
```

Für die experimentelle Installation lege ich nun drei weitere Partitionen an, dies kann von der schon existierenden FreeBSD-Installation aus erfolgen:

```
gpart add -t freebsd-ufs -l testrootfs -b 1m -s 5g ada1
gpart add -t freebsd-ufs -l testvarfs -a 1m -s 5g ada1
gpart add -t freebsd-ufs -l testusrfs -a 1m -s 45g ada1
newfs -U -t /dev/gpt/testrootfs
newfs -U -t /dev/gpt/testvarfs
newfs -U -t /dev/gpt/testusrfs
```

Dann erfolgt, diesmal mit einer FreeBSD 11 Alpha3 Bootonly CD die Installation und Einrichtung, wie bis Kapitel 12 beschrieben.

Achtung: Bei den keymap-Einstellungen der `/etc/rc.conf` soll in aktuellen FreeBSD-Versionen für eine deutsche Tastatur nur noch folgendes eingetragen werden:²⁹

```
keymap="de"
```

Damit auch diese weitere FreeBSD-Installation über das Grub-Menü meiner GNU/Linux-Installation zugänglich ist, wird unter Linux der `/etc/grub.d/40_custom` folgendes hinzu gefügt:³⁰

```
menuentry 'FreeBSD 11' {
    insmod ufs2
    set root='(hd1,5)'
    kfreebsd /boot/loader
}
```

²⁸ Oder dem jeweils aktuellsten 11er Zweig.

²⁹ Vgl. im folgenden Thread den Kommentar von @bluescreen <https://www.bsdforen.de/threads/plasma5-unstimmigkeiten-mit-locale-keymap-stabilit%C3%A4t.32639/#post-281692>

³⁰ Anm.: `set root='(hd1,5)'` bezieht sich auf ein FreeBSD, dessen Root-Verzeichnis auf `ada1p5` liegt.

Zur nochmaligen Überprüfung folgt zunächst eine kurze Übersicht wichtiger Konfigurations-Dateien. Erst dann geht es weiter mit plasma5 und zusätzlichen Anwendungen.

Achtung: Bitte auf die Modifikationen achten, sie werden über Fußnoten erläutert.

Die **/etc/rc.conf** sollte nun wie folgt aussehen:

```
hostname="biber.unix.pc"
keymap="de"31
ifconfig_em0="DHCP"
sshd_enable="YES"
# Set dumpdev to "AUTO" to enable crash dumps, "NO" to disable
dumpdev="AUTO"
## Zusatzeintraege ##
linux_enable="YES"
dbus_enable="YES"
hald_enable="YES"
nfs_client_enable="YES"
rpc_statd_enable="YES"
rpc_lockd_enable="YES"
cupsd_enable="YES"
cups_browsed_enable="YES"
devfs_system_ruleset="devfsrules_common"
```

Der Inhalt der **/boot/loader.conf**:

```
snd_envy24_load="YES"
nvidia_load="YES"
linux_load="YES"
```

Die **/etc/fstab**³²:

# Device	Mountpoint	FStype	Options	Dump	Pass#
/dev/ada1p5	/	ufs	rw	1	1
/dev/ada1p6	/var	ufs	rw	2	2
/dev/ada1p7	/usr	ufs	rw	2	2
/dev/ada1p4	/usr-stable	ufs	rw	2	2
tmpfs	/tmp	tmpfs	rw,mode=01777	0	0
md99	none	swap	sw,file=/usr-stable/swap/swap,late	0	0
proc	/proc	procfs	rw	0	0
fdesc	/dev/fd	fdescfs	rw	0	0
taschenmaus.unix.pc:/daten/share			/nas/share		nfs rw 0 0
taschenmaus.unix.pc:/daten/pc			/nas/pc		nfs rw 0 0
taschenmaus.unix.pc:/daten/nutzer/holger			/usr/home/holger/nas		nfs rw 0 0
taschenmaus.unix.pc:/daten/nutzer/diane			/usr/home/diane/nas		nfs rw 0 0

Die **/etc/hosts**:

192.168.1.20	taschenmaus.unix.pc	taschenmaus
192.168.1.30	farbmaus.unix.pc	farbmaus
192.168.1.111	gerbil.unix.pc	gerbil
192.168.1.121	biber.unix.pc	biber
192.168.1.131	hamster.unix.pc	hamster

31 Seit kurzem soll nur noch dieser einfache Eintrag „de“ für die deutsche Tastatur verwendet werden, siehe auch Fußnote 29.

32 Anm.: Die fstab-Partitionseinträge bitte ggf. anpassen. Die /usr Partition ada1p4 der älteren FreeBSD-10.3 Installation mountete ich hier in FreeBSD 11 als /usr-stable, damit ich das swapfile von dort auch hier benutzen kann, wie am Eintrag md99 zu sehen ist.

Die **/etc/devfs.conf**:

```
perm    /dev/da0      0666
perm    /dev/da1      0666
perm    /dev/da2      0666
perm    /dev/da3      0666
perm    /dev/cd0       0666
```

Die **/etc/devfs.rules**:

```
[devfsrules_common=7]
add path 'ad[0-9]\*' mode 0666
add path 'ada[0-9]\*' mode 0666
add path 'da[0-9]\*' mode 0666
add path 'acd[0-9]\*' mode 0666
add path 'cd[0-9]\*' mode 0666
add path 'mmcsd[0-9]\*' mode 0666
add path 'pass[0-9]\*' mode 0666
add path 'xpt[0-9]\*' mode 0666
add path 'ugen[0-9]\*' mode 0666
add path 'usbctl' mode 0666
add path 'usb/\*' mode 0666
add path 'lpt[0-9]\*' mode 0666
add path 'ulpt[0-9]\*' mode 0666
add path 'unlpt[0-9]\*' mode 0666
add path 'fd[0-9]\*' mode 0666
add path 'uscan[0-9]\*' mode 0666
add path 'video[0-9]\*' mode 0666
add path 'tuner[0-9]*' mode 0666
add path 'dvb/\*' mode 0666
add path 'cx88*' mode 0660
add path 'cx23885*' mode 0660
add path 'iicdev*' mode 0660
add path 'uvisor[0-9]*' mode 0660
```

Und dann noch die **/usr/local/etc/cups/client.conf**:

```
ServerName 192.168.1.30:631
```

a Kopieren und Integrieren des plasma5 Ports³³

Achtung: Bei den Optionen der plasma5 Ports sollte immer DOCS und NLS aktiviert bleiben, weil sonst eventuell einige Ports nicht bauen.

```
mkdir /var/tmp/kde5
svnlike co http://area51.pcbsd.org/branches/plasma5 /var/tmp/kde5
cd /var/tmp/kde5/plasma5/Tools/scripts && ./kdemerge -a /usr/ports
```

Falls auf dem System schon ein kde4 installiert war, folgende Pakete zunächst deinstallieren:

```
pkg delete kde\* attica\*
```

Bitte dann den Rechner neu starten.

³³ Vgl. mit <http://area51.pcbsd.org/branches/plasma5/README>

b Bau von plasma5 samt Zusatzpaketen

```
portmaster -d x11/plasma5-plasma  
portmaster -d x11/konsole  
portmaster -d editors/kate
```

Wenn das deutsche Sprachpaket installiert werden soll, muss das Paket **kf5-helpcenter** zuvor entfernt werden wegen Dateikonflikten:

```
pkg delete kf5-helpcenter && portmaster -d german/kde5-l10n
```

Weitere interessante Ports sind in <http://area51.pcbsd.org/branches/plasma5> zu finden.