

FreeBSD auf dem Desktop mit zahlreichen Beispielen

1. April 2017

Vorwort

Erst seit Mitte 2015 befaße ich mich mit FreeBSD¹. Davor habe ich lange Zeit GNU/Linux² als Haupt-System genutzt, mit S.u.S.E 5.3³ habe ich angefangen, dann bin ich wenige Jahre später bei Debian gelandet, und dies sehr lange unter Nutzung verschiedener Zweige als auch Derivate. Gelegentliche Ausflüge zu PCLOS, Mageia, Manjaro und Arch hat es auch gegeben.

Da ich kein großer Freund des neuen Init-Systems **systemd**⁴ bin, das sich mittlerweile bei vielen GNU/Linux-Distributionen quasi zum Standard etabliert hat, nutze ich als primäres GNU/Linux System seit einigen Monaten MX-Linux⁵.

Letztes Jahr habe ich dann doch aus Neugier damit begonnen, mir FreeBSD näher anzuschauen (ein Ausflug zu PC-BSD⁶ davor war nur von kurzer Dauer, dieses System ist mir etwas zu überladen, und wenn man zu FreeBSD lernen möchte, eignet sich wohl das Original besser, als ein Derivat)⁷. Nach einiger Lektüre und einigen Installationsversuchen, die auch zu laufenden benutzbaren Systemen geführt haben, habe ich mich dann beim englischsprachigen Forum⁸ angemeldet, um eine Kommunikationsplattform zu diesem unixartigen System zu haben. Neben Themen mit Fragen hat es dann auch einige Auseinandersetzungen gegeben. Folgenden Fehler habe ich dabei gemacht:

Ich bin mit der Erwartung an FreeBSD heran gegangen, dass es ähnlich zu funktionieren hat wie GNU/Linux.⁹ Das sehe ich mittlerweile anders, davon abgesehen hat es aber bei den Diskussionen einen Aspekt gegeben, den sogar der Foren-Admin Warren Block¹⁰ aufgegriffen und verteidigt hat:

¹Siehe <https://www.freebsd.org/de/>

²Siehe <https://www.gnu.org/gnu/linux-and-gnu.de.html>

³Anm.: Diese Distribution aus Nürnberg kam mit mehreren CD-Roms und einem dicken Benutzerhandbuch. Meines Erachtens bot sie einen sehr schönen Einstieg zu diesem unixartigen Betriebssystem.

⁴Anm.: Zu diesem Init-System - siehe auch <https://de.wikipedia.org/wiki/Systemd> - gibt es diverse Auseinandersetzungen, neben sachlichen Pro- und Kontra-Meinungen aus Entwickler- wie auch Nutzersicht gibt es ausgesprochen unproduktive Schlammschlachten. Ich glaube einerseits durchaus, dass **systemd** interessante Features hat, die Dinge erleichtern können. Allerdings halte ich es andererseits für eine Fehlentwicklung, wenn man für einfach zu handhabbare Dinge ein hochkomplexes Werkzeug benutzt. Davon abgesehen finde ich es bedenklich, wenn sich allmählich in größeren Teilen der GNU/Linux-Szene eine Auffassung durchsetzt, dass Kritik an diesem Init-System eigentlich nur von Volltrotteln kommen kann. Da ich in Sachen **systemd** trotz meiner Vorbehalte gerne auf dem Laufenden bleiben möchte, habe ich noch ein Antergos laufen, das ist weitgehend ein Vanilla-Arch plus einigen Zusätzen, das sich dank des Installers Calamares - siehe <https://calamares.io/> - rasch installieren lässt.

⁵Siehe <http://www.mepiscommunity.org/mx>

⁶Anm.: PC-BSD heißt mittlerweile TrueOS und basiert auf FreeBSD, die Entwickler haben den Anspruch, ein installationsfreundliches BSD für Desktop-Systeme und Server-Systeme zu schaffen, neben FreeBSD kommen Eigenentwicklungen hinzu, siehe dazu auch <https://web.pcbbsd.org/>. Eine meines Erachtens spannende Eigenentwicklung, der ein eigenes Kapitel dieses Buches gewidmet sein wird, ist der unter einer BSD-Lizenz stehende Desktop Lumina, siehe dazu <https://lumina-desktop.org>

⁷Anm.: Ich möchte PC-BSD keinesfalls schlecht machen, wenn das schlanke Desktop-Environment Lumina zum Einsatz kommt, ist das Argument mit der Überladenheit wohl nicht mehr zutreffend.

⁸Siehe <https://forums.freebsd.org/>

⁹Anm.: Ich habe Beispiele gebracht, bei denen Schlagworte wie „xyz + Arch“ in Suchmaschinen sofort brauchbare Treffer liefern, man bei FreeBSD hingegen manchmal zunächst im Regen steht. Nun hat Arch ein spitzenmäßiges Wiki, es wird oftmals auch von Leuten anderer GNU/Linux-Distributionen bei kniffligen Angelegenheiten genutzt.

¹⁰Anm.: Warren hat eine Seite mit einer Sammlung von Howtos - siehe <http://www.wonkity.com/~wblock/docs/> - ich kann sie nur empfehlen und werde darauf noch zurück kommen.

Dokumentationen zu FreeBSD können auch so gemacht sein, dass sie von Anfängerinnen und Anfängern ohne große Vorkenntnisse abzarbeiten sind und dass dabei dann funktionierende Systeme heraus kommen.

Zwei sehr hilfreiche Sachen habe ich zu jener Zeit allerdings noch nicht gekannt, und zwar eine Installationsanleitung zu FreeBSD auf dem Desktop von Nicole¹¹ und die Seiten zur FreeBSD-Installation von Martin.¹²

Abgesehen davon ist es ja schön, zu fordern, dass etwas so gemacht sein kann, aber noch etwas anderes, nicht nur zu fordern, sondern es dann auch zu machen. Ich habe meine Installationen von FreeBSD bisher immer stichpunktartig auf Papier notiert, heraus gekommen ist eine lose Sammlung von dahin gekritzelten Seiten, ein Teil dieser Sammlung ist dann von einem unserer Kater genüsslich zerfleddert worden. Der Kater lebt noch, ich bin dem Tierchen sogar ein wenig dankbar, denn danach habe ich überlegt, ob ich nicht meine Installationen mit dem PC protokollieren und dann auch als Dokumentation zur Verfügung stellen soll. Zunächst habe ich geplant, zur FreeBSD-Installation auf meinem Desktop-Rechner eine Dokumentation zu schreiben, inklusive einem experimentellen Kapitel zu plasma5.¹³ Diese Dokumentation sollte schon über eine reine Beschreibung hinaus gehen, aber nicht allzu umfangreich werden. Eine erste Rohfassung habe ich dann im deutschsprachigen BSD-Forum zur Diskussion gestellt, allen Beteiligten bin ich dankbar für die freundlichen Rückmeldungen, sachlichen Ergänzungen und kritischen Anmerkungen.

Dieses Feedback hat Konsequenzen, und zwar den fast vollständigen Umbau der Dokumentation. Im Vordergrund steht nach wie vor mein Bemühen, einsteigerfreundlich über FreeBSD auf dem Desktop zu schreiben, die eigenen Installationen spielen auch weiterhin eine Rolle, sie stehen aber nicht mehr zu sehr im Mittelpunkt, sondern dienen eher als anschauliche Beispiele. Vom Umfang her wird es nun auf ein Buch hinauslaufen. Wichtig ist mir dabei folgendes (das habe ich auch schon während einer Forum-Diskussion geäußert):

„[...]Was ich da schreibe, möchte ich auch ordentlich verstanden haben und was ich schreibe, soll auch für andere nachvollziehbar sein und dann bei den anderen auch funktionieren. Ich finde es sehr abschreckend, wenn Leute, die mit unixoiden Systemen beginnen, auf Dokumentationen stoßen, die schwierig nachzuvollziehen sind und die viel zu viel voraus setzen. Daher meide ich auch weitgehend während meiner Erklärungen zu einer Sache Querverweise, ich möchte nicht, dass man zu sogenanntem Howto-Hopping gezwungen ist, um bei Installationen weiter zu kommen (Dokumentation a verweist bei einer Sache auf Handbuch x und dort muss dann bei Manpage zu z nachgeschaut werden, so etwas mag für Profis einfach zu bewerkstelligen sein, aber für Anfängerinnen und Anfänger kann das frustrieren), sondern anhand dieser Dokumentation soll jede / jeder schrittweise ein System aufsetzen können und auch dabei begreifen, was sie / er jeweils tut (Hinweise auf Handbuch, Manpages oder Howtos sind bei mir immer als Ergänzung und Wissenserweiterung zu verstehen).[...]“

Anfangs habe ich geplant, und auch schon damit begonnen, diese Dokumentation mit LibreOffice zu schreiben, zumal ich mit diesem Office-Paket ein wenig vertraut bin. Nach einigen Diskussionen im deutschsprachigen BSD-Forum habe ich mich dann doch für \LaTeX ¹⁴ bzw. den \LaTeX -Editor LyX¹⁵ entschieden. \LaTeX und ähnliche professionelle Software ist eindeutig mäch-

¹¹Siehe <https://cooltrainer.org/a-freebsd-desktop-howto/>

¹²Siehe <http://denkrobot.de/doku.php>

¹³Anm.: Unter FreeBSD ist es dank des inoffiziellen area51 Projektes möglich, sehr aktuelle Versionen von plasma5 samt KDE-Applikationen zu bekommen, siehe dazu <https://freebsd.kde.org/news.php> und <http://area51.pcbsd.org/branches/plasma5/README>

¹⁴Siehe <https://www.latex-project.org/>

¹⁵Siehe <https://www.lyx.org/>

tiger und wohl auch besser für mein Vorhaben geeignet. Es bedarf zwar einer gewissen Einarbeitungszeit, aber die lohnt sich, das Layout solcher Dokumente ist sehr schön.

Weiterhin basieren die hier nun verwendeten Installationen von FreeBSD auf der nun veröffentlichten Version FreeBSD 11.0 und nicht, wie zu Beginn geplant, auf FreeBSD 10.3.

Einleitung

Das Buch soll ein ausführlicher Leitfaden zur Installation von FreeBSD auf Desktop-Rechnern oder Notebooks sein, einer Bündelung von wichtigen Informationen, angereichert mit zahlreichen Beispielen,¹⁶ und richtet sich an Anfängerinnen und Anfänger.¹⁷

Bevor ich richtig loslege, sei vorweg gestellt, was das Buch nicht sein soll:

Dieses Buch habe ich nicht als Konkurrenz zum FreeBSD-Handbuch konzipiert, es bietet vermutlich für FreeBSD-Profies kaum neue Erkenntnisse. Falls hier einige Sachen nicht optimal oder gar fehlerhaft umgesetzt sind, und es in jenen Fällen deutlich bessere Möglichkeiten einer Konfiguration gibt, bitte ich um Nachsicht.

Der Aufbau ist sukzessiv, es soll ein umfangreicher sogenannter roter Faden zu FreeBSD auf dem Desktop entstehen. Manche Kapitel beginnen mit theoretischen Abschnitten, wer es eher eilig hat, kann natürlich einiges überspringen, ich empfehle dies aber nicht, weil ich mich bemühe, in diesen Abschnitten Sachen auch so zu erläutern, dass sie in den dazu passenden praktischen Teilen als hilfreiches Hintergrundwissen fungieren. Einige Kapitel sind vielleicht für manche nicht weiter interessant, wer sich zum Beispiel für UFS2 als Dateisystem entschieden hat, mag sich eventuell nicht weiter mit dem Kapitel zu ZFS befassen.

Im ersten Teil des Buches geht es um die **Basiseinrichtung des Systems**:

- Neben Bezugsquellen von FreeBSD erläutere ich eine Vorbereitung von Installationsmedien.
- Weiter geht es dann mit dem ersten Start von FreeBSD über ein Installationsmedium und dem Prüfen der eigenen Hardware mit verschiedenen Werkzeugen, die FreeBSD schon auf dem Installationsmedium mitbringt.
- Danach stelle ich den Standardinstaller **bsdinstall**¹⁸ vor, ich gehe dann einige Installationen des Basissystems anhand meines Desktop-Rechners durch,¹⁹dabei kommen auch Fragen zur Verwendung des Dateisystems und Partitionierung zur Sprache.²⁰
- Eine Erläuterung der weiteren Schritte einer Basis-Installation bis zum ersten Start des neu installierten FreeBSD-Systems rundet den ersten Teil ab.²¹

¹⁶Anm.: Die Beispiele beziehen sich auf unsere PCs, Notebooks und unser Netzwerk, ein Übertragen auf andere PC Situationen und Nutzerbedürfnisse sollte aber nicht allzu schwierig sein.

¹⁷Anm.: 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/>

¹⁸Manpage <https://www.freebsd.org/cgi/man.cgi?bsdinstall>

¹⁹Anm.: Neben verschiedenen Installationsszenarien auf UFS2- und ZFS-Dateisystemen spielt auch Multiboot eine Rolle.

²⁰Anm.: Bei der Verwendung des Dateisystems ZFS kann man nicht mehr im klassischen Sinn von Partitionieren sprechen. Mehr zu Dateisystemen im weiteren Verlauf.

²¹Anm.: Der FreeBSD-Installer ist weitgehend selbsterklärend und meines Erachtens so gut gemacht, dass ich dazu nicht sehr ausführlich schreiben werde.

Der zweite große Teil umfasst dann den **weiteren Aufbau des Systems nach der Basisinstallation:**

- Ich beginne mit Erläuterungen zu wichtigen Konfigurationsdateien und den Möglichkeiten der Edition dieser Dateien, die die Basisinstallation schon mitbringt.
- Dann ist die Einrichtung neuer Gruppen und Nutzerkonten dran, samt der Befugnisse, die man neuen Nutzern gestattet.
- Umfangreich fallen die Erläuterungen zum Paketmanagement aus und welche Möglichkeiten man unter FreeBSD hat, neue Software hinzu zu installieren.²²

Im dritten großen Teil widme ich mich dann dem **Desktop:**

- Nach Vorbemerkungen zur Unterstützung verschiedener Grafikkarten fange ich mit der Installation und Konfiguration des Xservers an.
- Weiter geht es mit der Soundkarte. Die bisher von mir verwendeten Soundchips ließen sich problemlos mit FreeBSD zum Erklingen bringen. Allerdings bedarf es zum Teil einiger Feineinstellungen, um gewisse Sonderfunktionen auch zum Laufen zu bringen.²³
- Das folgende sehr umfangreiche Kapitel handelt von der jeweiligen Installation und Konfiguration einzelner Desktop-Umgebungen.²⁴ Ausgewählt habe ich KDE4, Xfce4 und Lumina. Eventuell kommt noch Gnome hinzu, allerdings bin ich mit Gnome nicht sehr vertraut und habe es bisher auch unter GNU/Linux eher gemieden.²⁵
- Weiter geht es mit typischer Software auf einem Desktop-Rechner wie Mailprogrammen, Browsern, Office-Suiten, ich gehe dabei auch auf Software rund um Audio, Video und Bilder ein.
- Zu einem Desktop-System gehört oft die Verwendung eines oder sogar mehrerer lokal angeschlossener Drucker. Daher befaße ich mich auch mit der Einrichtung von Cups und der Druckerkonfiguration unter FreeBSD.²⁶
- Weiterer Peripherie wie Scannern, Webcams, Anschluss von Digitalkameras und Smartphones widme ich ebenfalls ein Kapitel.

Im vierten Teil behandle ich **fortgeschrittene Themen:**

- Zunächst geht es um die Integration eines FreeBSD-Desktop-Rechners in ein bestehendes Netzwerk, ich thematisiere ausführlicher ssh-Zugriff auf andere Rechner, gefolgt von einem Unterkapitel zu Remote-Printing, also der Anbindung an einen Print-Server.
- Ich dokumentiere dann die Migration unseres Daten-NAS von Debian GNU/Linux nach FreeBSD, dafür richte ich wieder ein Root-on-ZFS System ein, allerdings diesmal mit Redundanz, weil ich den Pool spiegeln möchte.

²²Anm.: In diesem Unterkapitel behandle ich **pkg** als zentrales Installationswerkzeug von FreeBSD, gefolgt von der Nutzung nur vorkompilierter Pakete für eine Installation (packages-only) oder das Einbeziehen von Ports (builds+packages) oder dem Komplettbau aus Ports (builds-only). Schon hier sei erwähnt, dass für die eigentliche Installation immer **pkg** zuständig ist, daher hat man bei allen Installationsvarianten immer eine **pkg**-Datenbank, in der alle installierten Pakete vermerkt sind und installiert daher nichts am Paketsystem vorbei.

²³Anm.: Dabei denke ich an Mikrophoneingänge, Kopfhörerausgänge und ähnliches.

²⁴Anm.: Ausführlich eingehen werde ich auf KDE4, Xfce4 und Lumina.

²⁵Anm.: Damit meine ich nicht, dass ich es mir gar nicht erst angeschaut habe, nach kurzer Zeit musste bisher Gnome2 / Gnome3 bei mir meistens wieder KDE4 oder manchmal Xfce4 weichen, weil ich mit diesem Desktop-Environment einfach nicht richtig warm werde.

²⁶Anm.: Wer unter GNU/Linux mit Cups ordentlich eingerichtete Drucker laufen hatte, muss unter FreeBSD mit Cups keine Abstriche machen. Auf andere Möglichkeiten, Drucker einzurichten, gehe ich nur am Rande ein.

- Breit angelegt sein wird das Unterkapitel zur Einrichtung und Pflege eines lokalen Repositoriums. Zunächst werde ich zeigen, wie man mit **poudriere**²⁷ einzelne Pakete oder Paketserien sauber bauen, lokale Repos mit diesen Paketen befüllen und Client-Rechnern in einem Netzwerk zur Verfügung stellen kann. Es folgt als Beispiel der Bau sämtlicher Plasma5-Pakete aus dem noch inoffiziellen Area51-Port.²⁸

²⁷Siehe <https://www.freebsd.org/doc/handbook/ports-poudriere.html> und <https://wiki.bsdforen.de/kategorie:howto:poudriere>

²⁸Siehe <https://freebsd.kde.org/> und <http://area51.pcbbsd.org/branches/plasma5/README>

Inhaltsverzeichnis

I Die Basisinstallation von FreeBSD	9
1 Vor der Installation	10
1.1 Bezugsquellen	10
1.2 Der erste Start – Zur Hardware-Unterstützung	11
2 Auf eine Festplatte damit: Zur Installation	13
2.1 Das Tastatur-Layout	13
2.2 Wie soll der Rechner heißen	14
2.3 Auswahl verschiedener Komponenten für die Installation	15
2.4 Die Partitionierung	15
2.4.1 Dateisysteme: Der Klassiker UFS2	16
2.4.1.1 UFS2 - Automatische Einrichtung durch den Installer	16
2.4.1.2 Manuelle Partitionierung einer Festplatte mit der Shell	18
2.4.2 ZFS: Ein innovatives Dateisystem	19
2.4.2.1 ZFS - Automatische Einrichtung durch den Installer	19
2.4.2.2 Root-on-ZFS: Manuelle Einrichtung	19
2.5 Die Paßwortvergabe für den Administrator	21
2.6 Das Netzwerk konfigurieren	21
2.7	22
2.8	22
2.9 Abschluss der Installation	22

II Die Grundkonfiguration des Systems	24
III FreeBSD als Desktop-System	25
IV Fortgeschrittene Themen	26
3 Integration eines FreeBSD-Desktop-Rechners in ein Netzwerk	27
4 Migration eines Daten-Nas von Debian GNU/Linux nach FreeBSD	28
5 Erstellen eines lokalen Repos mit plasma5	29
5.1 Aufsetzen und Einrichten von poudriere ²⁹	29
5.1.1 poudriere aufsetzen	29
5.1.2 Konfiguration	30
5.1.2.1 Wichtige Dateien	30
5.1.2.2 Ports einlesen	30
5.1.2.3 Jail mit poudriere erstellen	31
5.1.3 Bau von Paketen	31
5.1.3.1 Die Paketliste	31
5.1.3.2 Optionen	31
5.1.3.3 Der Bau	31
5.1.3.4 Weitere Hinweise zum Umgang mit poudriere	32
5.2 Der Bau von plasma5 mit Hilfe von poudriere...	32
5.3 Installation von plasma5 auf einem FreeBSD-System	33

Teil I

Die Basisinstallation von FreeBSD

Kapitel 1

Vor der Installation

1.1 Bezugsquellen

Die diesem Buch zugrunde liegende Referenz-Hardware (verschiedene Rechner und Notebooks von meiner Frau und mir mit Intel- und Nvidia-Grafik) wird ordentlich von FreeBSD unterstützt, zu Grafikkarten von Ati / AMD kann ich mangels Hardware keine Angaben machen. Was mir fremde Hardware angeht, sei folgendes angemerkt:

Da diese Dokumentation sehr auf meine Erfahrungen mit eigenen Installationen ausgerichtet ist, kann ich natürlich auch nur über solche Hardware ausführlich schreiben, die ich selbst habe. Mit Vermutungen über mir fremde Hardware, zur der ich nur etwas gelesen habe, möchte ich mich eher zurückhalten, es steht leider auch einiges im Internet dazu, was entweder veraltet ist, ungeprüft abgeschrieben oder nur gemutmaßt wurde. Es ist meines Erachtens ausgesprochen ärgerlich, Zeit mit Hinweisen und Anleitungen zu verbringen, die dann nicht greifen und ich möchte mich mit meiner Dokumentation nicht in diese gerade für Anfängerinnen und Anfänger frustrierende Kategorie von Anleitungen einreihen. Wenn mir allerdings Leute von ihrer mir fremden Hardware detailliert berichten, dass sie ordentlich unter FreeBSD zum Laufen zu bekommen ist, nehme ich diese Hinweise gerne auf, selbstverständlich mit Nennung der Quelle.

Ein Installationsmedium mit FreeBSD eignet sich auch gut dafür, die eigene Hardware auf FreeBSD-Tauglichkeit zu testen. Daher geht es im Folgenden zunächst um die Bezugsquellen:

Info Werkzeuge Falls bisher nur mit Windows gearbeitet wurde: Auch dafür gibt es zahlreiche Werkzeuge zum Erstellen von bootbaren USB-Sticks.¹ (da ich kein Betriebssystem von Microsoft nutze, kann ich hier mangels Erfahrungen und Testmöglichkeiten keine praktischen Tipps dazu geben).² Unter GNU/Linux oder einem anderen unix-artigen System benötigt ihr zum Erstellen eines Installationsmediums auf USB-Stick das Programm **dd**.³

Damit nicht unnötig Rohlinge vergeudet werden – vielleicht kommt FreeBSD nach einigen Tests ja gar nicht mehr in Frage – und damit die Tests nervenschonend sind (heulende DVD-Laufwerke sind verzichtbar), empfehle ich die Verwendung eines USB-Sticks.⁴ Passende Images sind unter

¹siehe z. B. <http://alternativeto.net/software/suse-studio-imagewriter/>

²Anm.: Hier könnte ich auch nur eine Suchmaschine bemühen.

³Manpage <http://man7.org/linux/man-pages/man1/dd.1.html>

⁴Anm.: Falls ihr nicht von USB-Medien booten könnt, habt ihr auch die Möglichkeit, optische Medien wie CDs oder DVDs zur Installation zu verwenden. Wichtig beim Brennen der Images auf die Medien ist die Auswahl „Brennen eines ISO-Abbildes“ oder ähnliches in eurem Brennprogramm. Kopiert ihr die Images bloß auf CD oder DVD, erzeugt ihr sogenannte Daten-CDs oder Daten-DVDs und diese Medien sind nicht bootbar.

folgender URL zu finden:

<http://www1.de.freebsd.org/freebsd/releases/ISO-IMAGES/11.0/>

Nach dem Herunterladen des Images xxx-amd64-memstick.img und der Überprüfung der CHECKSUMME⁵ wird ein USB-Stick mit dem Image wie folgt beschrieben:

```
# dd if=<name-des-images> of=/dev/<name-des-USB-Stick-Devices> bs=4M
```

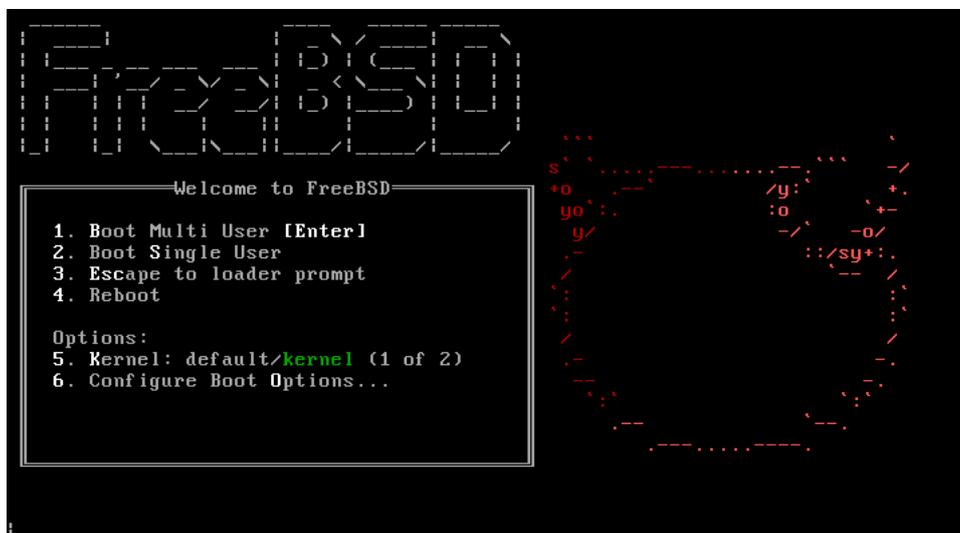
Achtung Mit dd sorgsam umgehen, nicht, dass ggf. die Festplatte statt des Sticks beschrieben wird. Das Device ist in der Regel kurz nach dem Einstöpseln des Sticks mit dmesg in den letzten Zeilen zu erkennen.

Stimmen die Input- und Output-Angaben überein, wurde der Stick ordentlich beschrieben und ist dann einsetzbar.

1.2 Der erste Start – Zur Hardware-Unterstützung

Info Werkzeuge Es kommen `kbdmap`⁶ und `dmesg`⁷ zum Einsatz.

Der Rechner muss im BIOS so eingestellt sein, dass er von USB-Stick booten kann, alternativ bieten die meisten Systeme an, kurz nach dem Start ein Bootdevice-Menü über eine Taste aufrufen zu können. Ist der Stick in Ordnung, wird nach kurzer Zeit ein Bildschirm mit rotem FreeBSD-Logo und verschiedenen Bootoptionen angezeigt.



Mit **[RETURN]** wird dann der eigentliche Bootvorgang des Systems gestartet. Es rasen eine Menge an Informationen über den Bildschirm und es erscheint dann eine Auswahlbox mit Begrüßungstext und der Auswahlmöglichkeit zwischen **<INSTALL>**, **<SHELL>** und **<LIVE-CD>**.

Für Tests ist **<LIVE-CD>** zu wählen, an der Shell **root** einzugeben, gefolgt von **[RETURN]**, das Live-Medium ist nun Ready-to-use.

⁵Anm.: In der Regel werden SHA512-Summen angegeben, und ihr braucht zum Überprüfen das – auch abhängig vom Betriebssystem - entsprechende Werkzeug, unter GNU/Linux zum Beispiel `sha512sum`.

⁶Manpage <https://www.freebsd.org/cgi/man.cgi?query=kbdmap>

⁷Manpage <https://www.freebsd.org/cgi/man.cgi?query=dmesg>

Zunächst das Tastaturlayout an die eigene Tastatur anzupassen, dazu wird das Programm **kbdmap** aufgerufen und das entsprechende Layout - in den meisten Fällen wohl das deutsche - auszuwählen (ob mit oder ohne Akzent-Tasten ist an dieser Stelle nicht so wichtig).

FreeBSD unterstützt verschiedene Architekturen,⁸ ich beschränke mich hier allerdings nur auf die amd64-Architektur (64 Bit), bis auf wenige Ausnahmen sollte alles hier Geschriebene auch auf die i386-Architektur (32 Bit) zu übertragen sein.

Wer mit GNU/Linux vertraut ist, kennt vermutlich schon **dmesg**

⁸Siehe auch <https://www.freebsd.org/de/where.html>

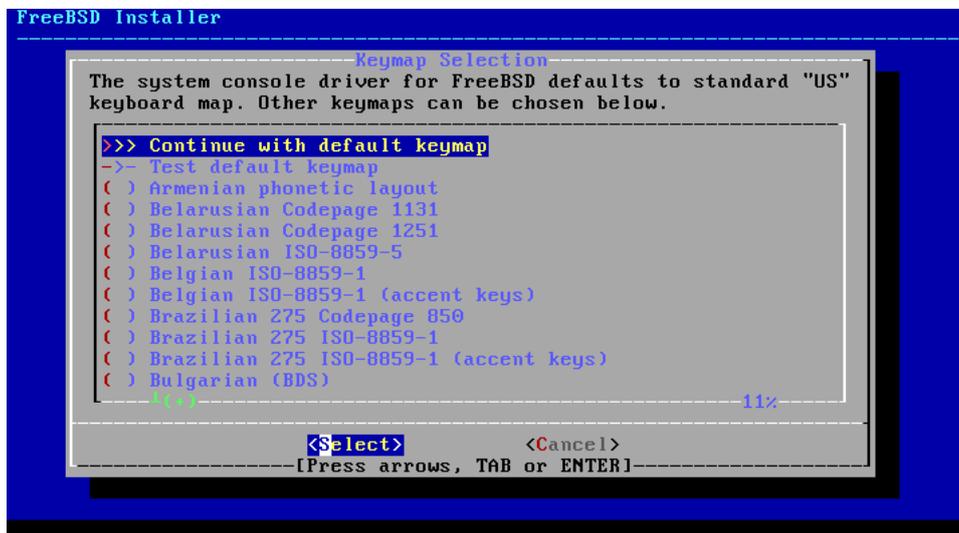
Kapitel 2

Auf eine Festplatte damit: Zur Installation

Falls nun das Interesse groß genug ist, FreeBSD zu installieren, wird wie schon oben erläutert mit dem Installationsmedium gebootet, aber diesmal <INSTALL> ausgewählt. Hiermit wird der Standard-Installer `bsdinstall`¹ aufgerufen, im folgenden gehe ich die einzelnen Schritte zur Installation des Basis-Systems durch.²

2.1 Das Tastatur-Layout

In der ersten Eingabemaske des Installers kann das Tastatur-Layout eingestellt werden.



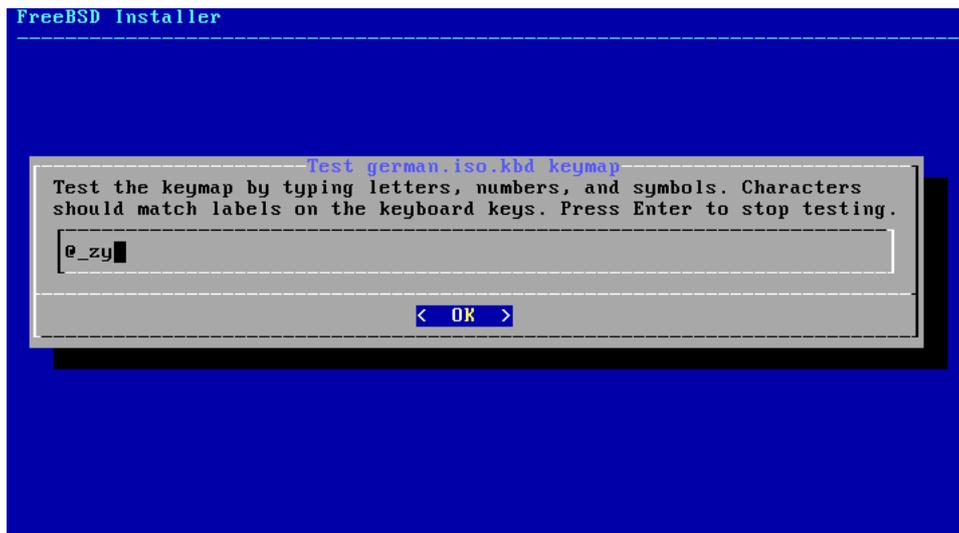
Standardmäßig ist US-Layout eingestellt, was auch immer hier ausgewählt wird: eine Feinjustierung ist gegen Ende der Installation noch möglich.³ Das neue Layout kann hier auch getestet

¹Manpage <https://www.freebsd.org/cgi/man.cgi?bsdinstall>

²Anm.: Falls ihr bisher mit GNU/Linux gearbeitet habt, könnt ihr in der Regel bei den dort verwendeten Installern ein komplettes Betriebssystem samt Xserver, Desktop-Oberflächen, Office, Browser, Mail und weiteren Programmen zur Installation auswählen. Dies ist bei `bsdinstall` anders, hiermit könnt ihr nur das Basis-System FreeBSD installieren.

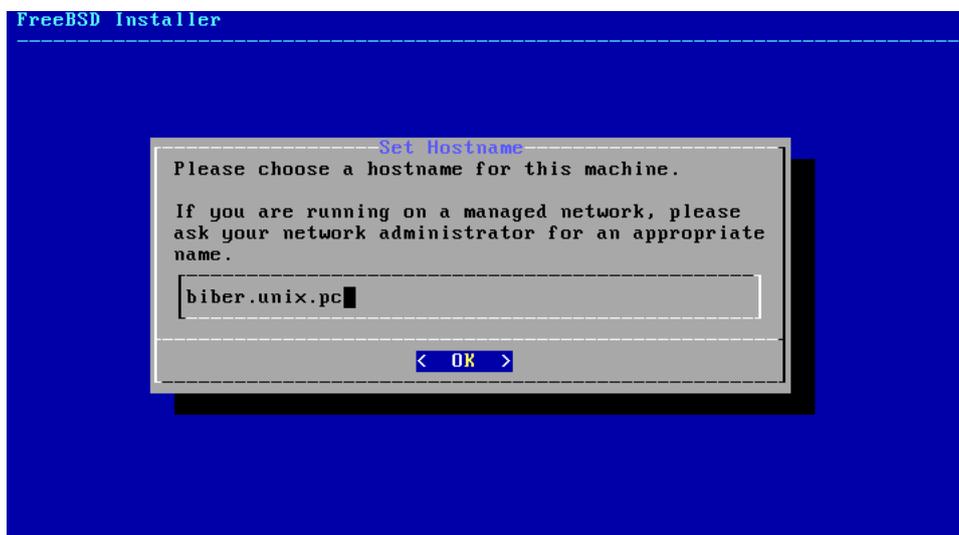
³Anm.: Natürlich können am installierten System - von bestimmten Festplattenaufteilungen einmal abgesehen - sämtliche Einstellungen bei Bedarf modifiziert werden.

werden, wichtig ist die korrekte Zuordnung von z, y, @, -, _ u.s.f. Es kann sein, dass ä, ü, ö oder ß noch nicht funktionieren, das ist an dieser Stelle aber noch nicht so wichtig.⁴



2.2 Wie soll der Rechner heißen

Im Folgeschritt geht es um die Vergabe eines Rechnernames, hier ist nicht nur der reine Hostname gemeint, sondern `<hostname>.<domainname>`.

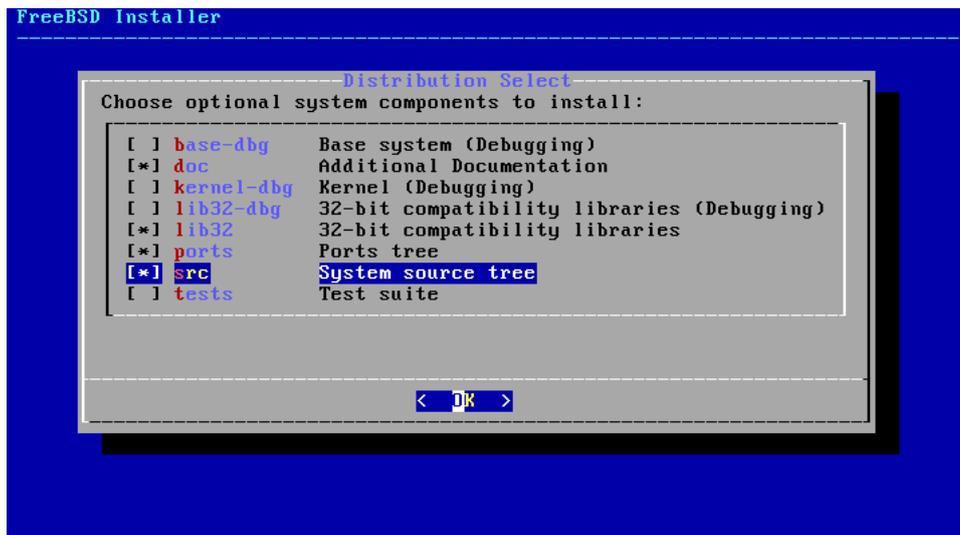


Die Abbildung zeigt die Namesvergabe für meinen Desktop-PC **biber.unix.pc**.⁵ Falls der Rechner in ein Netzwerk integriert werden soll, ist es sinnvoll, den Domain-Namen hier auch einzusetzen.

⁴Anm.: Zur Lokalisierung mit korrekter Tastatur-Konfiguration und locale-Einstellungen gibt es ein extra Unterkapitel.

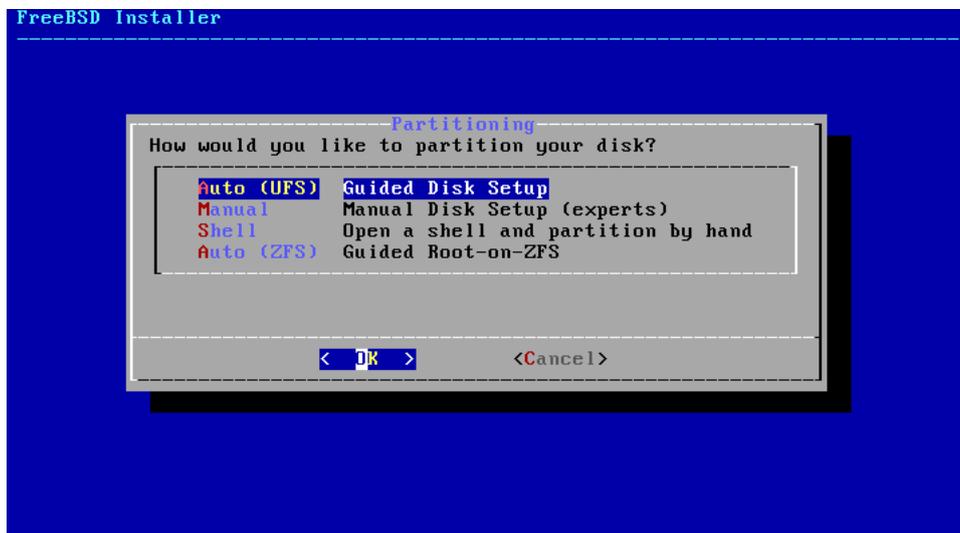
⁵Anm.: Für unser Heimnetzwerk habe ich die Domain **unix.pc** vergeben, die verschiedenen Rechner sind nach Tieren – bevorzugt Nagetieren – benannt.

2.3 Auswahl verschiedener Komponenten für die Installation



2.4 Die Partitionierung

Das nächste Fenster des Installers bietet die Vorbereitung der Festplatte / Festplatten für die nachfolgende Installation an.



Wer eine Suchmaschine zum Thema „richtige Partitionierung“ füttert, dürfte für einige Tage Lesestoff kredenzt bekommen. Es gibt sicherlich Festplattenaufteilungen, die als suboptimal zu bezeichnen sind, aber von groben Schnitzern abgesehen hängt es auch sehr von eigenen Vorlieben ab, wie partitioniert wird.

In den vergangenen Jahren bin ich unter GNU/Linux immer mehr dazu über gegangen, mit wenig Partitionen zu arbeiten.⁶ Eigentlich reicht eine Partition für das Betriebssystem aus, Swap kann über ein Swapfile eingerichtet werden und ein regelmäßiges Backup der eigenen Dateien macht eine extra Partition für **/home** auch überflüssig, zumal in Fall unseres Heimnetzwerkes

⁶Anm.: Damit meine ich wenige Partitionen pro System.

alle persönlichen Daten per NFS-Share auf die Desktop-Rechner eingebunden werden und von diesen Daten auf dem NAS einmal pro Nacht auch noch ein inkrementelles Backup auf einem weiteren NAS erstellt wird.

Als ich im deutschsprachigen BSD-Forum meine erste Rohfassung dieser Dokumentation zur Diskussion gestellt habe, wurde mir tendenziell auch zu möglichst wenig Partitionen auf einem Desktop-System geraten. Daher halte ich es auch unter FreeBSD so, mit möglichst wenig Partitionen (bzw. bei Verwendung des Dateisystems ZFS mit wenig Data-Sets) zu arbeiten.

Arbeitet ihr mit Windows, ist das Standarddateisystem wohl heutzutage nur noch NTFS, ob es überhaupt noch möglich ist, ein Windows 8 oder 10 unter VFAT zu installieren, kann ich mangels Erfahrungen mit dem Aufsetzen und Warten dieser Betriebssysteme nicht sagen.

Hattet ihr auch schon mit GNU/Linux zu tun, bietet sich eine ganz andere Situation: Was hättet ihr denn gern für ein Dateisystem? EXT3 / EXT4, XFS, JFS, REISER, REISER4, BTRFS? Seit Jahren nutze ich unter GNU/Linux das robuste EXT4-Dateisystem, XFS und JFS habe ich aber auch schon ohne Schwierigkeiten verwendet. Meines Erachtens sind es heutzutage vor allem EXT4, XFS und BTRFS, die für eine Installation von GNU/Linux in Frage kommen, zumal an diesen Dateisystemen und Werkzeugen zur Wartung aktiv entwickelt wird.

Unter FreeBSD ist es noch etwas überschaubarer, eigentlich kommen nur zwei Dateisysteme in Frage, diese unterscheiden sich aber dermaßen voneinander, dass jedem dieser Dateisysteme ein eigenes Unterkapitel gewidmet ist.

2.4.1 Dateisysteme: Der Klassiker UFS2

UFS ist laut Wikipedia das Urgestein unter den Dateisystemen für Unix.⁷ Es wurde in den 80er Jahren entwickelt, kommt aber in seiner ursprünglichen Form heutzutage unter FreeBSD nicht mehr zum Einsatz. UFS2, entwickelt im Jahr 2002 von Kirk McKusick, ist das Nachfolge-Dateisystem und wurde zuerst mit FreeBSD 5 eingeführt.

Das Werkzeug zum manuellen Anlegen von UFS2 Dateisystemen heißt **newfs**.⁸ In der Manpage findet ihr diverse Einstellungsmöglichkeiten, es gibt eine Journalfunktion und die Möglichkeit, Softupdates zu aktivieren.

2.4.1.1 UFS2 - Automatische Einrichtung durch den Installer

Wer kein großes Interesse daran hat, sich mit der Partitionierung zu befassen, dem „kommt“ der Installer mit der Möglichkeit „entgegen“, dies automatisch erledigen zu lassen. Die Einstellungen für diese automatische Partitionierung sind in der Regel gut gesetzt.

Im Partitionierungsdialog wird zunächst der erste Eintrag gewählt:

<Auto (UFS) Guided Disk Setup>

Im Folgedialog kann bestimmt werden, ob die gesamte Festplatte verwendet werden soll, oder eine manuelle Einrichtung einzelner Partitionen bevorzugt wird. Ist die Verwendung der ganzen Platte gewünscht ohne persönliche Vorlieben einer speziellen Festplattenaufteilung, ist **<Entiere Disk>** zu wählen:

⁷Siehe https://de.wikipedia.org/wiki/Unix_File_System

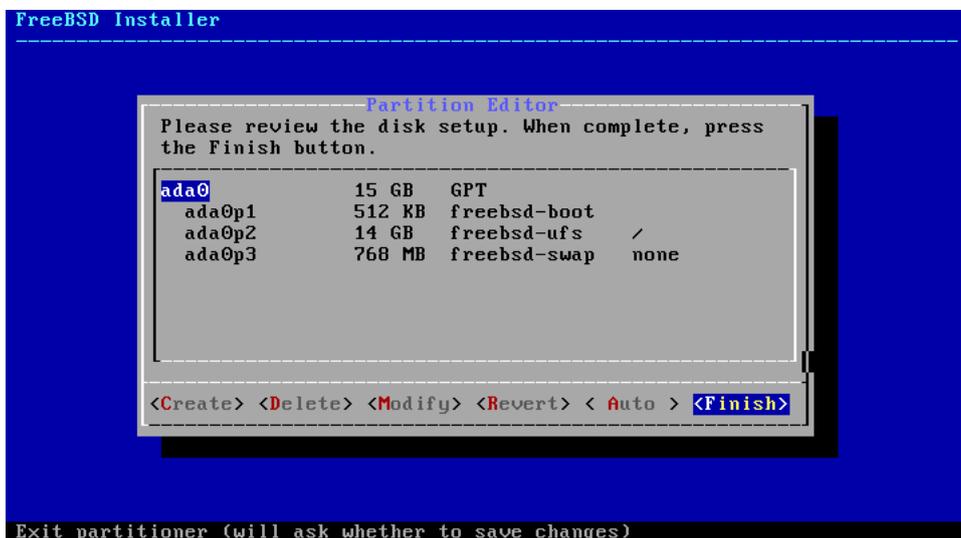
⁸Manpage <https://www.freebsd.org/cgi/man.cgi?newfs>



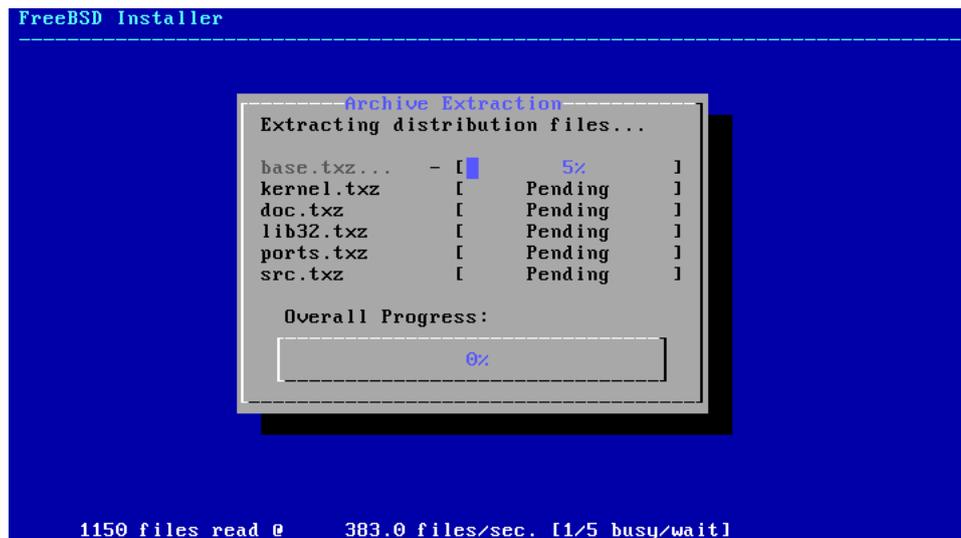
Im folgenden Dialog werden zahlreiche Typen von Partitionstabellen angeboten, hier ist <GPT> eine gute Wahl:



Der Installer zeigt nun nochmals eine Zusammenfassung mit der Möglichkeit, hier noch nachträglich eingreifen zu können. Soll nichts geändert werden, ist die Partitionierung mit <Finnish> zu bestätigen



und der Folgedialog mit **<Commit>**, damit wird dann die Partitionierung und die Installation des Basissystems ausgelöst:



Falls kein Interesse an der Erläuterung einer manuellen Festplatteneinrichtung oder dem Dateisystem ZFS besteht, geht es nun mit **Unterkapitel 2.5** weiter.

2.4.1.2 Manuelle Partitionierung einer Festplatte mit der Shell

Info Werkzeuge: Es kommen **gpart**⁹ und **newfs**¹⁰ zum Einsatz.

Ihr habt spezielle Wünsche bei der Aufteilung der Festplatte oder ihr wollt diese Arbeit schon aus didaktischen Gründen gerne manuell mit einer Shell durchführen? Dann seid ihr bei diesem Unterkapitel genau richtig.

Im folgenden beschreibe ich die manuelle Partitionierung einer SSD¹¹ unter Verwendung des Dateisystems UFS2.

Es handelt sich bei dieser Beschreibung um die erste Platte, also um **/dev/ada0** (FreeBSD) bzw. **/dev/sda** (GNU/Linux), die zweite Platte wird unter FreeBSD mit **/dev/ada1**, unter Linux mit **/dev/sdb** bezeichnet. Vermutlich wollt auch ihr FreeBSD auf der ersten Platte installieren, falls nicht, passt bitte die Angaben entsprechend an.¹² Beim Abschnitt zur Partitionierung wählt ihr die Shell aus und lasst euch zunächst mit

```
# gpart show -p ada0
```

Informationen zu eurer Festplatte anzeigen.

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

Dann geht ihr, falls auf dieser Platte keine mehr für euch erhaltenswerten Daten sind, bitte wie folgt vor:

⁹Manpage <https://www.freebsd.org/cgi/man.cgi?gpart>

¹⁰Manpage <https://www.freebsd.org/cgi/man.cgi?newfs>

¹¹Siehe dazu auch <http://www.wonkity.com/~wblock/docs/html/ssd.html>

¹²Anm.: Nehmt ihr die zweite Platte, ersetzt bitte alle ada0-Angaben durch ada1.

1. **# gpart destroy -F ada0** Damit löscht ihr die Partitionstabelle
2. **# gpart create -s gpt ada0** Damit legt ihr eine neue gpt Tabelle an
3. **# gpart add -t freebsd-boot -s 512k -a4k -l boot0 ada0** Hierdurch legt ihr eine 512 kb kleine 4k ausgerichtete auf **boot0** gelabelte Partition an
4. **# gpart bootcode -b /boot/pmbr -p /boot/gptboot -i 1 ada0** Schreiben des Bootcodes für FreeBSD
5. **# gpart add -t freebsd-swap -l swap0 -b 1m -s 8g ada0** Dies erzeugt eine 8 GB große Swap-Partition mit 1M Ausrichtung und dem Label **swap0**
6. **# gpart add -t freebsd-ufs -l system0 -a 1m -s 15g ada0**
7. **# newfs -U -t /dev/gpt/system0** Mit Punkt 6 legt ihr eine 15 GB große UFS-Partition mit 1M Ausrichtung und dem Label **system0** an, hierhin soll FreeBSD installiert werden. Die Partition formatiert ihr, wie bei Punkt 7 zu sehen ist, durch das Werkzeug newfs mit UFS2, mit -t schaltet ihr Trim ein, 1 mit -U Softupdates, ein Journaling der Softupdates (-SUJ) wird nicht aktiviert
8. **# gpart add -t freebsd-ufs -l daten0 -a 1m -s 45g ada0**
9. **# newfs -U -t /dev/gpt/daten0** Damit legt ihr eine 45 GB große UFS-Partition mit 1M Ausrichtung und dem Label **daten0** an, und formatiert sie mit UFS. Hierhin können persönliche Daten geschrieben werden. Punkte 8 und 9 sind nicht erforderlich für die Installation. Falls euch eine UFS-Partition für das System und eure Daten reicht, passt bitte die Größe der Systempartition **system0** eurer Festplattengröße entsprechend an und überspringt das Anlegen einer Datenpartition **daten0**
10. **# mount /dev/gpt/system0 /mnt && mkdir /mnt/boot** Der Installer installiert nach **/mnt**, daher sind diese Schritte erforderlich
11. **# exit**

Nach dem Verlassen der Shell wird automatisch die Installation von FreeBSD gestartet. Falls an ZFS kein Interesse besteht, geht es mit dem **Unterkapitel 2.5** weiter.

2.4.2 ZFS: Ein innovatives Dateisystem

2.4.2.1 ZFS - Automatische Einrichtung durch den Installer

Auch bei der Verwendung von ZFS braucht ihr nichts großartig manuell zu machen, der Installer bietet Euch eine automatische Einrichtung einer ZFS-Umgebung für FreeBSD an, bei der ihr allerdings einige Dinge modifizieren könnt.

2.4.2.2 Root-on-ZFS: Manuelle Einrichtung

Info Werkzeuge: Zum Einsatz kommen **gpart**, **zpool**¹³ und **zfs**¹⁴

Der FreeBSD-Installer ist wie schon geschrieben in der Lage, eine ZFS-Umgebung automatisch zu kreieren für eine Root-on-ZFS Installation. Bei solchen Systemen gibt ein

```
# zfs list
```

¹³Manpage <https://www.freebsd.org/cgi/man.cgi?zpool>

¹⁴Manpage <https://www.freebsd.org/cgi/man.cgi?zfs>

eine große Anzahl verschiedener Data-Sets aus, die sicherlich gut vorkonfiguriert sind, allerdings ist mir diese Fülle von Data-Sets zu unübersichtlich und ähnlich wie im Kapitel zur manuellen Einrichtung einer Festplatte mit UFS2 reichen mir eigentlich neben der freebsd-boot-, der freebsd-swap- und der freebsd-zfs-Partition im ZFS-Pool ein Data-Set für das System und eines für Daten.

Der FreeBSD-Installer erlaubt wie auch schon im Kapitel zu UFS2 gesehen beim Abschnitt zur Partitionierung eine komplett manuelle Einrichtung und ähnlich wie bei UFS2 beschreibe ich nun eine manuelle Root-on-ZFS Einrichtung am Beispiel meines Desktop-Rechners **biber.unix.pc**.

Die Einrichtung erfolgt auf einer 256 GB großen SSD, auf der neben einer freebsd-boot- und einer freebsd-swap- eine freebsd-zfs-Partition für einen Pool angelegt wird, ich verzichte also zunächst auf Redundanz, die Einrichtung erfolgt aber so, dass ich später durch den Einbau einer weiteren gleichgroßen SSD die Möglichkeit habe, den ZFS-Pool zu spiegeln. Ich wähle dazu im Abschnitt zur Festplatteneinrichtung die <Shell> für die manuelle Partitionierung aus und führe folgende Schritte durch – falls die Platte neu ist, entfallen die Schritte 1 und 2:

1. **# zpool import -f -N <poolname>** Hiermit wird ein eventuell vorhandener alter Pool importiert, die Option -N sorgt dafür, dass die Data-Sets dieses Pools nicht gemountet werden. Wenn keinerlei Daten mehr von diesem alten Pool benötigt werden, kann der Pool gelöscht werden.
2. **# zpool destroy -f <poolname>** Dies zerstört den alten Pool vollständig.
3. **# gpart destroy -F ada0** Dies zerstört das Geom ada0 samt Partitionstabelle.
4. **# gpart create -s gpt ada0** Erzeugen einer GPT Partitionstabelle
5. **# gpart add -t freebsd-boot -l boot0 -a4k -s 512k ada0** Anlegen einer 512 KB großen Bootpartition mit korrekter Ausrichtung und der Labelvergabe boot0
6. **# gpart add -t freebsd-swap -l swap0 -b 1m -s 8g ada0** Mein Rechner hat 8 GB Ram, die Größe der Swap-Partition von 16 GB ist Ansichtssache, sie erhält das Label swap0. Die Partition wird durch die Angabe **-b 1m** ordentlich ausgerichtet.
7. **# gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i 1 ada0** Dies schreibt den Bootcode, wichtig ist hier die Angabe **/boot/gptzfsboot** statt **/boot/gptboot**, sonst bootet das System nicht
8. **# gpart add -t freebsd-zfs -l system0 -a 1m -s 235g ada0** Die ZFS-Partition wird hier auf system0 gelabelt, die Größe richtet sich nach dem verbleibenden Platz auf der SSD und sollte etwas kleiner als dieser verbleibende Platz gewählt werden, die Blockgröße wird durch **-a 1m** auf 1 MB gesetzt.
9. **# zpool create -m /zbiber zbiber /dev/gpt/system0** Hiermit wird ein Pool zbiber erzeugt für die mit system0 gelabelte ZFS-Partition und auf das Verzeichnis /zbiber gemountet, die Meldung cannot mount /zbiber: failed to create mountpoint. . . kann ignoriert werden
10. **# zfs create -o mountpoint=none zbiber/ROOT** Dies erzeugt das Data-Set zbiber/ROOT ohne Mountpunkt
11. **# zfs create -o mountpoint=/mnt zbiber/ROOT/default** Dies erzeugt das Data-Set zbiber/ROOT/default, hierher erfolgt gleich die Installation des FreeBSD-Systems, da der Installer in den Pfad /mnt installiert, muss demnach der Mountpunkt temporär für die Installation auf /mnt gesetzt werden
12. **# zfs create -o mountpoint=/local zbiber/local** Dieses Data-Set möchte ich für diverse Daten wie den Inhalt von /home und auch für poudriere nutzen, es ist für die Installation entbehrlich, die Meldung filesystem successfully created, but not mounted. . . kann ignoriert werden
13. **# zfs set compression=lz4 zbiber/ROOT/default** Hier setze ich die Option compression=lz4 für das Data-Set, wohin gleich FreeBSD installiert wird. Zum Setzen von Optionen und verschiedenen Kompressionsverfahren für Data-Sets siehe die Manpage zu zfs
14. **# zpool set bootfs=zbiber/ROOT/default zbiber** Hier wird das Verzeichnis festgelegt, von welchem gebootet werden soll, die Syntax ist: zpool set bootfs=<Name-des-Datasets-von-Root> <Poolname> Vor dem nächsten Schritt wird noch per mount geprüft, ob zbiber/ROOT/default auf /mnt gesetzt ist

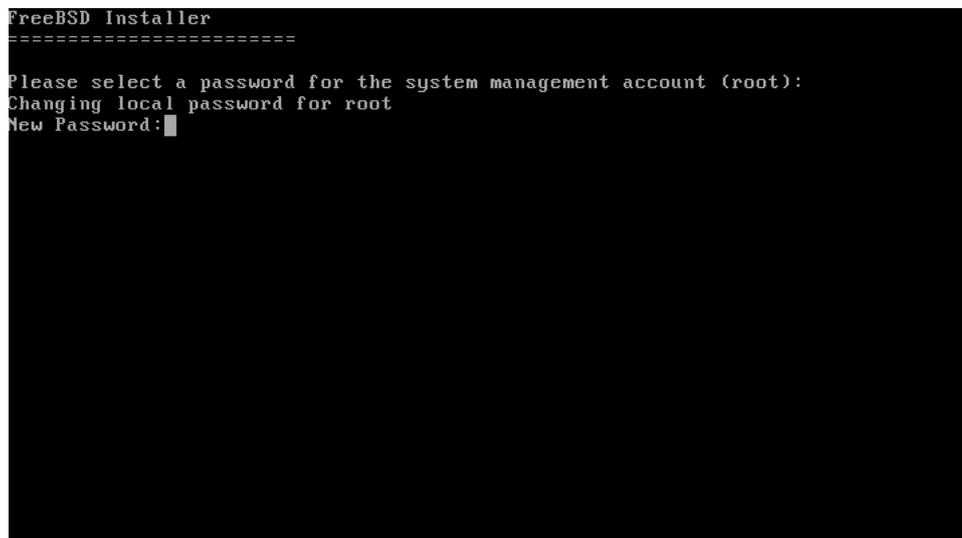
15. **# mkdir -p /mnt/boot/zfs** Dies erzeugt das Bootverzeichnis, in das für den ZFS-Betrieb notwendige Sachen installiert werden

16. **# exit**

Nach dem Verlassen der Shell wird automatisch die Installation von FreeBSD gestartet.

2.5 Die Paßwortvergabe für den Administrator

Sobald das Basissystem installiert wurde, öffnet sich ein neuer Dialog, hier habt ihr die Möglichkeit, ein Paßwort für den Administrator zu vergeben. Eine Mischung aus Zahlen, Zeichen und Buchstaben ist zu empfehlen, ihr solltet euch das Paßwort allerdings gut merken können:

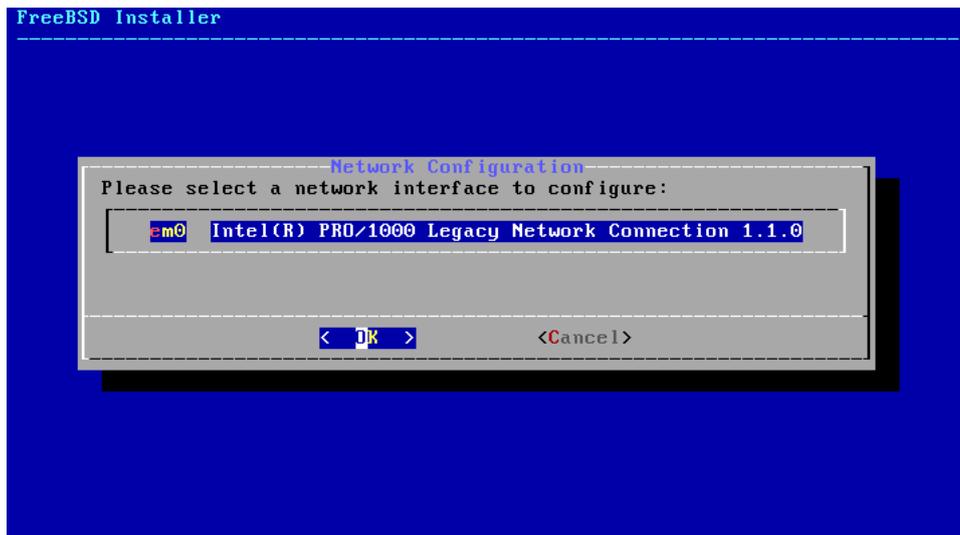


```
FreeBSD Installer
=====
Please select a password for the system management account (root):
Changing local password for root
New Password: █
```

Das Paßwort ist zwei Mal einzugeben, ihr seht die Eingabe nicht, auch keine Sternchen.

2.6 Das Netzwerk konfigurieren

Nach der Paßwortvergabe für den Administrator geht es nun an die Netzwerkkonfiguration. Alle von FreeBSD gefundenenen Netzwerkschnittstellen werden hier angezeigt - meistens ist es eine Netzwerkkarte - bitte wählt die für die Netzwerkanbindung vorgesehene Schnittstelle aus:



2.7 ...

2.8 ...

2.9 Abschluss der Installation

Vor dem Abschluss bietet euch der Installer noch an, per Shell in das neue System zu wechseln – dies nennt man auch **chroot**¹⁵ – um weitere Einstellungen zu machen.

Sofern ihr FreeBSD auf UFS2 installiert, oder Euch für eine automatische ZFS-Installation entschieden habt, könnt ihr dieses Angebot ablehnen, das System neu starten und mit Teil II weiter machen.

Habt Ihr hingegen eine manuelle ZFS-on-Root Installation gewählt, nehmt dieses Angebot bitte an.

Info Werkzeuge: Ihr braucht den Editor **ee**.¹⁶

Ich ergänze, bezogen auf meinen Rechner, mit dem Editor **ee** die Datei **/boot/loader.conf** um folgende Einträge, passt die Einträge bitte Eurer Installation entsprechend an:

```
# zfs_load="YES"
# vfs.root.mountfrom="zfs:zbiber/ROOT/default"
```

In die Datei **/etc/rc.conf** kommt:

```
# zfs_enable="YES"
# keymap=de.acc
```

¹⁵Siehe <https://de.wikipedia.org/wiki/Chroot>

¹⁶Anm.: Ausführlicher gehe ich auf diesen unter FreeBSD standardmäßig installierten Editor später noch ein, die Bedienung ist simpel, mit **# ee** /Pfad-zu-Datei öffnet ihr die Datei, dann könnt ihr schreiben, mit [Esc] beendet ihr den Eingabemodus, mit [Return] wird der Editor verlassen (ist vorgewählt) und mit einem weiteren [Return] werden die Änderungen in die Datei geschrieben (ist vorgewählt), zu weiteren Befehlen siehe auch die Manpage <https://www.freebsd.org/cgi/man.cgi?query=ee>

Danach wird der PC komplett herunter gefahren und vor der weiteren Einrichtung noch ein letztes Mal mit dem Installationsmedium gebootet. Diesmal wird nicht der Installer, sondern die <Shell> gewählt, deutsches Tastatur-Layout eingestellt (**kbdmap**) und dann per

```
# zpool import -f -N zbiber
```

der neue Pool importiert, ohne die Data-Sets zu mounten. Das Data-Set mit der FreeBSD-Installation hat als Mountpunkt noch **/mnt**, das wird nun geändert, mit

```
# zfs set mountpoint=legacy zbiber/ROOT/default
```

wird der Mountpunkt auf legacy gesetzt. Dann wird der Pool exportiert mit

```
# zpool export -f zbiber
```

danach kann per Reboot in das neue System gestartet werden.

Teil II

Die Grundkonfiguration des Systems

Teil III

FreeBSD als Desktop-System

Teil IV

Fortgeschrittene Themen

Kapitel 3

Integration eines FreeBSD-Desktop-Rechners in ein Netzwerk

Kapitel 4

Migration eines Daten-Nas von Debian GNU/Linux nach FreeBSD

Kapitel 5

Erstellen eines lokalen Repos mit plasma5

... Achtung noch nicht gründlich überarbeitete Rohfassung ...

In diesem Kapitel geht es um den Bau eines lokalen Repos, ohne das System, was zum Bau verwendet wird, mit unnötigen Abhängigkeiten zu belasten oder gar dessen Konsistenz zu gefährden. Da plasma5 zur Zeit noch nicht im offiziellen Repo von FreeBSD und den offiziellen Ports zu haben ist, bietet es sich hier dafür regelrecht an.¹

Zunächst beginne ich mit einer Anleitung zum Aufsetzen und Konfigurieren von poudriere², es folgt dann die Vorbereitung der Ports für den Bau von plasma5 inklusive der Bauanleitung, und zum Schluss geht es an das Aufsetzen eines FreeBSD mit plasma5 unter Verwendung des lokalen selbst gebauten Repos.

Folgendes wird voraus gesetzt:

- ein Rechner mit einem per root-on-zfs aufgesetztem FreeBSD, wie in **2.4.2.2 Root-on-ZFS: Manuelle Einrichtung** beschrieben
- ein Datenset für die Verwendung von poudriere
- ein Datenset für die parallele Installation des FreeBSD Systems mit plasma5

5.1 Aufsetzen und Einrichten von poudriere³

5.1.1 poudriere aufsetzen

```
pkg install poudriere
zfs create -o mountpoint=/poudriere <poolname>/poudriere
```

¹Anmerkung: Bei Erscheinen dieser Doku kann das natürlich anders, und auch plasma5 dann per pkg direkt installierbar sein.

²Manpage <https://www.freebsd.org/cgi/man.cgi?query=poudriere>

³siehe auch <https://wiki.bsdforen.de/kategorie:howto:poudriere>

5.1.2 Konfiguration

5.1.2.1 Wichtige Dateien

In `/usr/local/etc` eine Datei `poudriere.conf` anlegen mit folgendem Inhalt:

```
ZPOOL=<poolname>
ZROOTFS=/poudriere
FREEBSD_HOST=http://ftp.de.freebsd.org/
RESOLV_CONF=/etc/resolv.conf
BASEFS=/poudriere/base
POUDRIERE_DATA=${BASEFS}/data
USE_PORTLINT=no
DISTFILES_CACHE=/poudriere/distfiles
CHECK_CHANGED_OPTIONS=verbose
CHECK_CHANGED_DEPS=yes
WRKDIR_ARCHIVE_FORMAT=txz
NOLINUX=yes
USE_TMPFS=yes
ALLOW_MAKE_JOBS=yes
PARALLEL_JOBS=4
PREPARE_PARALLEL_JOBS=4
(Anm.: Je nach Anzahl der CPU-Kerne)
```

Wichtig: Der Ordner `/poudriere/distfiles` muss angelegt werden.

In `/usr/local/etc/poudriere.d` eine Datei `<jailname>-make.conf` anlegen:

```
# build ports allways without X
WITHOUT_X11=yes
# default options
OPTIONS_UNSET= BONJOUR DEBUG X11
```

5.1.2.2 Ports einlesen

Der jeweils stabile Quartalsbranch, hier am Beispiel von 2017Q2, kann verwendet werden (mit dem Schalter `-p` kann man nach Geschmack einen Namen vergeben):

```
poudriere ports -c -B branches/2017Q2 -m svn+https -p 2017Q2
```

Ich verwende für plasma5 als Quelle `head`:

```
poudriere ports -c -B head -m svn+https -p plasma5head
```

Ein `poudriere ports -l` listet die eingelesenen Ports auf (hier am Beispiel meiner Portsammlungen):

PORTSTREE	METHOD	TIMESTAMP	PATH
lumina	portsnap	2017-03-11 04:36:02	/poudriere/base/ports/lumina
plasma5	svn+https	2017-03-23 19:34:46	/poudriere/base/ports/plasma5
plasma5head	svn+https	2017-03-27 20:01:20	/poudriere/base/ports/plasma5head

5.1.2.3 Jail mit poudriere erstellen

```
poudriere jail -c -j <jailname> -v 11.0-RELEASE -a amd64
```

Die Syntax ist eigentlich einfach:

- das -c (create) erzeugt eine Jail
- auf -j folgt der Jailname
- das -v steht für die in der Jail verwendete FreeBSD-Version
- und -a bestimmt die Architektur.

Auf meinem PC habe ich wie folgt eine Jail angelegt:

```
poudriere jail -c -j 110default -v 11.0-RELEASE -a amd64
```

5.1.3 Bau von Paketen

5.1.3.1 Die Paketliste

In `/usr/local/etc` wird z.B. eine **110default-liste** mit den zu bauenden Ports angelegt:

```
audio/flac
mail/claws-mail
www/firefox-esr
shells/bash
x11/xorg
x11/lumina
german/libreoffice
...
```

5.1.3.2 Optionen

Modifikation eines Ports im Ports-Tree 2017Q2:

```
poudriere options -p 2017Q2 -c x11/xorg
```

Zurücksetzen der Optionen eines Ports im Ports-Tree 2017Q2 auf Standard:

```
poudriere options -p 2017Q2 -r x11/xorg
```

5.1.3.3 Der Bau

Ein Skript in `/usr/local/bin` z.B. mit dem Namen **poudriere-build-packages** kann den Bau erleichtern (hier am Beispiel der Liste **110default-liste** unter Verwendung des Ports-Trees **2017Q2** und der jail **110default**):

```
#!/bin/sh
cd /usr/local/etc
poudriere bulk -f 110default-liste -j 110default -p 2017Q2
return 0
```

5.1.3.4 Weitere Hinweise zum Umgang mit poudriere

Update der Ports:

```
poudriere ports -u -p <portname>
```

Update der jails:

```
poudriere jail -u -j <jailname>4
```

Löschen von ports:

```
poudriere -d -p <portname>
```

Speicherort der Pakete ist in **/poudriere/base/data/packages/<portname>**.

Sehr praktisch: Jeder Ports-Tree erhält sein eigenes Repo, wird etwa der Ports-Tree **2017Q2** verwendet, entsteht in Abhängigkeit von der Paket-Liste ein Repo mit dem Namen:

```
/poudriere/base/data/packages/<jailname>-2017Q2
```

Nimmt man einen Ports-Tree **plasma5head**, entsteht ein eigenes Repo mit dem Namen:

```
/poudriere/base/data/packages/<jailname>-plasma5head
```

Außerdem: Die Repos sind ready-to-use, darum kümmert sich schon poudriere, man muss nicht händisch noch

```
pkg repo .
```

im Paketverzeichnis ausführen, sondern kann die Repo-Pfade direkt in **/usr/local/etc/pkg/repos/FreeBS** für die Nutzung unter **pkg** eintragen (zur Syntax komme ich später).

5.2 Der Bau von plasma5 mit Hilfe von poudriere...

Zunächst wird für den Bau der Ports-Tree head eingelesen und entsprechend aussagekräftig benannt:

```
poudriere ports -c -B head -m svn+https -p plasma5head
```

Dann wird um der besseren Recherchierbarkeit mit „whereis“ folgender Link gesetzt (Achtung, es darf kein normaler Portstree **/usr/ports** vorhanden sein):

```
ln -sf /poudriere/base/ports/plasma5head /usr/ports
```

Für das Holen der inoffiziellen Ports zu plasma5:

```
pkg install subversion (Anm.: Falls noch nicht installiert)
mkdir /tmp/plasma5stuff && cd /tmp/plasma5stuff
svn co http://area51.pcbsd.org/branches/plasma5
plasma5/Tools/scripts/kdemerge -a /usr/ports
cd && rm -R /tmp/plasma5stuff
```

⁴Anm.: Hiernach werden immer sämtliche Pakete der Paket-Liste neu gebaut!

Achtung Zur Zeit muss noch devel/icu gepatcht werden, damit alle Anwendungen mit Datei- und Ordernamen klar kommen, die Umlaute enthalten⁵

Eine Liste `/usr/local/etc/110default-plasma5head-liste` erstellen (dies ist ein Ausschnitt meiner Liste, bei Bedarf weitere Ports für den Bau hinzu fügen)

```
devel/dbus
devel/icu
german/kde-l10n
german/libreoffice
graphics/okular
mail/claws-mail
mail/claws-mail-address_keeper
mail/claws-mail-pdf_viewer
multimedia/gstreamer
multimedia/gstreamer-plugins
multimedia/gstreamer1
multimedia/gstreamer1-plugins
multimedia/qt5-phonon4-gstreamer
ports-mgmt/pkg
print/cups
shells/bash
sysutils/htop
sysutils/powerdxx
sysutils/cdrtools
sysutils/eject
sysutils/hal
sysutils/k3b
textproc/ibus
www/firefox-esr
www/firefox-esr-i18n
x11-fonts/liberation-fonts-ttf
x11/xorg
x11/kde5
x11/sddm
```

Ich habe ein ausführbares Script für den Bau erstellt `/usr/local/bin/pbplasma5head` mit folgendem Inhalt:

```
#!/bin/sh
cd /usr/local/etc
poudriere bulk -f 110default-plasma5head-liste -j 110default -p plasma5head
return 0
```

Damit wird dann der Bau des Repos angestoßen. Wenn alles glatt geht, liegt je nach Rechnerleistung in einigen Stunden das fertige Repo, benannt nach `<jailname>-<ports-tree>` unter `/poudriere/base/data/packages/110default-plasma5head`

5.3 Installation von plasma5 auf einem FreeBSD-System

Vorausgesetzt wird

- dass auf dem Rechner zfs und ein Pool verwendet wird

⁵https://bugs.freebsd.org/bugzilla/show_bug.cgi?id=216372

- eine frische Installation des Basis-Systems von FreeBSD 11 inklusive angelegten Nutzern und angepasster `/etc/rc.conf` an die jeweiligen Bedürfnisse
- das Dataset mit dem Mountpunkt `/poudriere`, in welchem sich das zuvor gebaute Repo **110default-plasma5head** befindet

Nach dem Aufsetzen des Systems wird zunächst ein Verzeichnis `/usr/local/etc/pkg/repos` erzeugt, dann eine **FreeBSD.conf** mit folgenden Einträgen:

```
local: {
  url: "file:///poudriere/base/data/packages/110default-plasma5head/",
  enabled: yes }
FreeBSD: { enabled: no }
```

Mit **pkg update** wird pkg aus dem lokalen Repo installiert und dieses dann eingelesen. Nun kann die Installation der weiteren Komponenten beginnen:

```
pkg install -x bash hal dbus ibus xorg kde5 german/kde-l10n
  german/libreoffice fire claws liberation k3b u.s.f.
```

Einige wichtige Dateien in `/etc` auf meinem Rechner (bitte anpassen):

/etc/fstab

```
tmpfs /tmp tmpfs rw,mode=01777 0 0
proc /proc procfs rw 0 0
fdesc /dev/fd fdscfs rw 0 0
/dev/ada0p2 none swap sw 0 0
```

/etc/rc.conf

```
zfs_enable="YES"
keymap=de
hostname="biber.unix.pc"
ifconfig_em0="DHCP"
ifconfig_em0_ipv6="inet6 accept_rtadv"
sshd_enable="YES"
moused_enable="YES"
moused_device="dev/ums0"
dumpdev="NO"
powerdxx_enable="YES"
hald_enable="YES"
dbus_enable="YES"
cupsd_enable="YES"
sendmail_enable="NO"
sendmail_submit_enable="NO"
sendmail_outbound_enable="NO"
sendmail_msp_queue_enable="NO"
devfs_system_ruleset="devfsrules_common"
```

/etc/devfs.conf

```
perm  /dev/da0      0666
own   /dev/da0      root:operator
perm  /dev/da1      0666
own   /dev/da1      root:operator
```

```

perm   /dev/da2      0666
own    /dev/da2      root:operator
perm   /dev/da3      0666
own    /dev/da3      root:operator
perm   /dev/da4      0666
own    /dev/da4      root:operator

```

/etc/devfs.rules

```

[devfsrules_common=7]
add path 'ada[0-9]\*' mode 0666
add path 'da[0-9]\*' mode 0666
add path 'cd[0-9]\*' mode 0666
add path 'mmc[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 'ulpt[0-9]\*' mode 0666
add path 'usbctl' mode 0666
add path 'usb/\*' mode 0666
add path 'fd[0-9]\*' mode 0666
add path 'uscan[0-9]\*' mode 0666
add path 'video[0-9]\*' mode 0666

```

/etc/sysctl.conf

```

vfs.usermount=1
vfs.zfs.min_auto_ashift=12

```

Die Nutzer bekommen bei mir die bash.

Um plasma5 zu starten, empfehle ich zunächst den manuellen Start. Für jeden Nutzer wird eine `~/.xinitrc` mit folgendem Inhalt angelegt:

```

export LANG=de_DE.UTF-8
exec ck-launch-session startkde

```

... WEITER AUSARBEITEN ...