Inhaltsverzeichnis

Einrichten eines Linux-Servers für den Betrieb der Thera-Pi-Datenbank.

Teil 1, die Installation bei mir zu Hause

1. Vorwort:

Solange man nur eine kleine Praxis mit 1-3 Thera-Pi-Installationen betreibt, reicht es wahrscheinlich aus, auf dem Hauptrechner die MySQL-Datenbank laufen zu lassen und mit den anderen PC's darauf zu zugreifen.

Hat man jedoch mehrere Thera-Pi-Installationen, die dazu auch noch an verschiedenen Orten (zu Hause, Zweitpraxis etc.) stehen und zu verschiedenen (Nacht)Zeiten auf die Datenbank zugreifen müssen, sollte man sich Gedanken über einen eigenen Datenbankserver machen. So ist es auch bei mir, 2 Praxen mit jeweils 1 Rezeptions-PC und Zugriff von zu Hause von 2 weiteren PC's.

Nun könnte ja jemand sagen, warum ein selbstständiger PC nur für eine Datenbank????

Ich lass einfach den Hauptrechner laufen und fertig. Gegen diese Annahme spricht, dass der übliche Bürorechner

- hardwareseitig nicht auf Dauerbetrieb ausgelegt ist
- ein aktuelles Windows zwar als ServerOS laufen kann, die Sicherheitseinstellungen aber häufig für die tägliche Arbeit so löchrig gestaltet wurden, dass ich starke Bedenken über die Sicherheit meiner Daten habe

Also steht mein Entschluss, einen eigenen Server für den Betrieb der Datenbank aufzustellen. Aufgeführte Herstellerlinks geben nur meine Einkaufsentscheidungen wieder und sollen keinesfalls als Werbung verstanden werden!

2. Die Aufgabenstellung

Folgende „Dinge“ soll mein Server können:

- auch wenn es komisch klingt: er soll stabil und sicher laufen!
- Er soll mich über Zwischenfälle informieren
- Ich will meine Daten regelmäßig (extern) gesichert haben
- Ansonsten will ich nichts von ihm hören

Das klingt erstmal nicht nach sonderlich großen Ansprüchen, wirft jedoch eine Reihe von Überlegungen und notwendigen Tätigkeiten auf, auf die ich im Nachhinein eingehen werde. Ich verwende nachfolgend die statische IP 192.168.0.2.

3. Die Hardware

Bei mir fiel die Wahl auf den HP ProLiant N36L MicroServer Hersteller-Seite

Ein kleiner, billiger Server, der aber für den Betrieb der MySQL-Datenbank und einiger weiterer Dienste (dazu später mehr) völlig ausreichend ist. Dazu ist sämtliche verbaute Technik vollständig Linuxkompatibel. Einziger Minuspunkt: ziemlich laut (für meine Ohren), aber da das Ding ja nicht auf'm Schreibtisch steht, noch okay.

3.1. Interne Erweiterung der Hardware

Diesen habe ich mit einer weiteren Festplatte und zusätzlichem Arbeitsspeicher ausgestattet.

Meine erste Überlegung war, insgesamt 3 Festplatten einzubauen, eine (SSD) für das OS und 2 weitere für einen RAID1-Verbund für die Daten. Aus Kostengründen habe ich mich dagegen entschieden. Ich habe jetzt also 2 x 250 GB Festplattenspeicher und 3 GB RAM.

In Zeiten der billigen TB-Platten kommt wohlmöglich die Frage auf, warum ich „so wenig" Festplattenplatz habe? Die Antwort ist ziemlich einfach:

- Serverfestplatten sind teuer und TB-Platten für den Heimgebrauch ungeeignet.
- 250 GB reichen so was von aus....
- durch die Verwendung von LVM ist es einfach, weitere Platten bei Bedarf nachträglich hinzuzufügen

3.2. Externe Erweiterung der Hardware

Vor den Server habe ich eine **U**nterbrechungsfreie**S**trom**V**ersorgung geschaltet, damit im Falle eines kurzzeitigen Stromausfalles die PC's weiterlaufen und bei längerem Stromausfall die Datenbank ordnungsgemäß runtergefahren werden kann. Und damit auch hier der reibungslose Betrieb unter Linux sichergestellt ist, nehme ich eine USV des Herstellers des apcups-dämons http://www.apcupsd.com/ APC Hersteller-Seite

4. Die Software bzw. das ServerOS

Ich habe verschiedene Linuxdistributionen ausprobiert und schlussendlich eine in einigen Punkten sicherlich sehr subjektive Entscheidung getroffen.

Folgene Distris haben ihren Weg auf den Server und bis auf eine auch wieder herunter gefunden:

- Mandriva 2010.2: mein langjähriger Begleiter mit einer hervorragenden Unterstützung fast aller Verwaltungsaufgaben durch grafische Assistenten, sehr komfortabel! Leider ist durch verschiedene wirtschaftliche Entscheidungen des Managements das Fortbestehen und damit auch das Aktualisieren von Sicherheitslücken nicht gesichert und so muss ich leider von meinem Favoriten Abstand nehmen.
- Mageia 1: der Communityfork von Mandriva. Bietet die Vorteile von Mandriva, muss sich jedoch erst noch beweisen. Schade.
- Ubuntu LTS Server: Die aktuelle Version ist leider veraltet (ein kleiner, aber feiner Widerspruch in sich....)
- CentOS: der freie Clon des Serverbetriebssystems von RedHat. Irgendwie scheinen dieser interessanten Distri die Entwickler wegzulaufen
- SuSE: mir persönlich irgendwie unsympatisch. Ich weiß nicht, ob es an dem bekifften Reptil liegt oder meinen Erinnerungen an YAST1 oder an was sonst.
- Debian in der aktuellen stabilen version 6.02. Weltweit tausendfach als Server getestet, für gut befunden und im Einsatz, eine große Community (wichtig für den support) und gute Dokumentation. Die soll es sein!

5. Die Installation

5.1. Das Medium

Wir brauchen also ein Installationsmedium. Der Server ist ein 64-bit-System, also spricht nichts dagegen, auch ein 64-bit-OS zu verwenden (es spricht auch nur wenig dafür, unser Server ist so klein, dass die Vorzüge eines 64-bit-OS nicht wirklich zum Tragen kommen). Es kann also auch ein 32-bit-OS verwendet werden. Ich beziehe mich nachfolgend auf Debian 6.02 64bit.

http://cdimage.debian.org/debian-cd/6.0.2.1/amd64/iso-dvd/debian-6.0.2.1-amd64-DVD-1.iso

Um das System auch Installieren zu können, brauchen wir weiterhin einen USB-Stick mit mindestens 4,5 GB Speicher (ACHTUNG: Alle Daten auf dem Stick werden gelöscht!) sowie ein Tool, welches uns das ISO auf den USB-Stick schiebt. Hierzu kann Mandriva Seed für Windows verwendet werden.

ftp.mandrivauser.de/mandriva_isos/2010.0/

Sollte dies unter einem aktuellen 64bit-Win nicht funktionieren, kann wohl auch

http://www.chip.de/downloads/SelfImage_30991577.html verwendet werden

5.2. Installationsvorgang

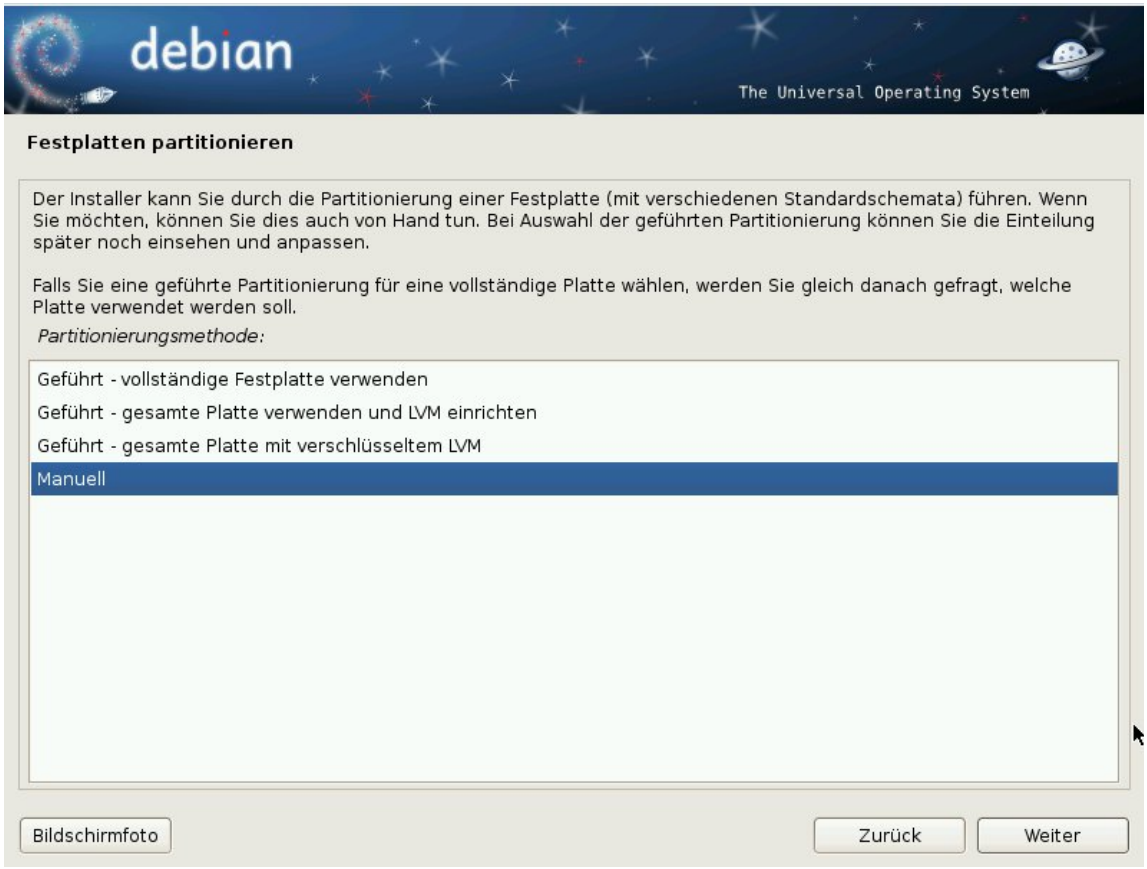
Im BIOS die Bootpriorität USB auf high einstellen und Rechner neu starten.

Ich beschreibe nur die Schritte, wo ich von den vorgeschlagenen Werten abweiche, ansonsten übernehme ich die Voreinstellungen.


debian
The Universal Operating System
Installer boot menu
Install
Graphical install
Advanced options >
Help


debian
The Universal Operating System
Advanced options
Back.. <
Expert install
Rescue mode
Automated install
Graphical expert install
Graphical rescue mode
Graphical automated install
Alternative desktop environments >

einige Extra-Worte zur Partitionierung



Ich habe 2 Festplatten, die ich als RAID1 verwenden will. Allerdings müssen nicht alle Daten gespiegelt werden und zusätzlich ist es hilfreich, wenn die Bootpartition kein RAID ist.

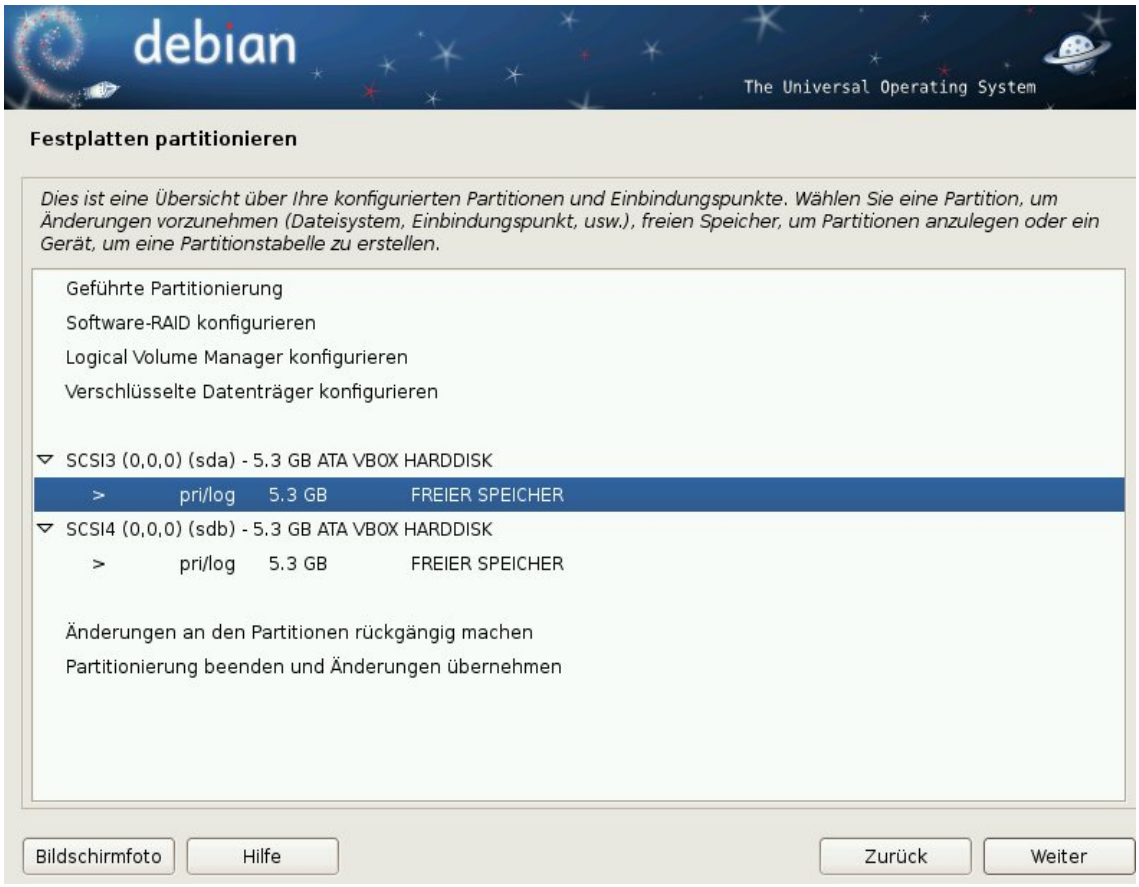
Also werde ich folgende Partitionen direkt auf die 2 Platten verteilen:

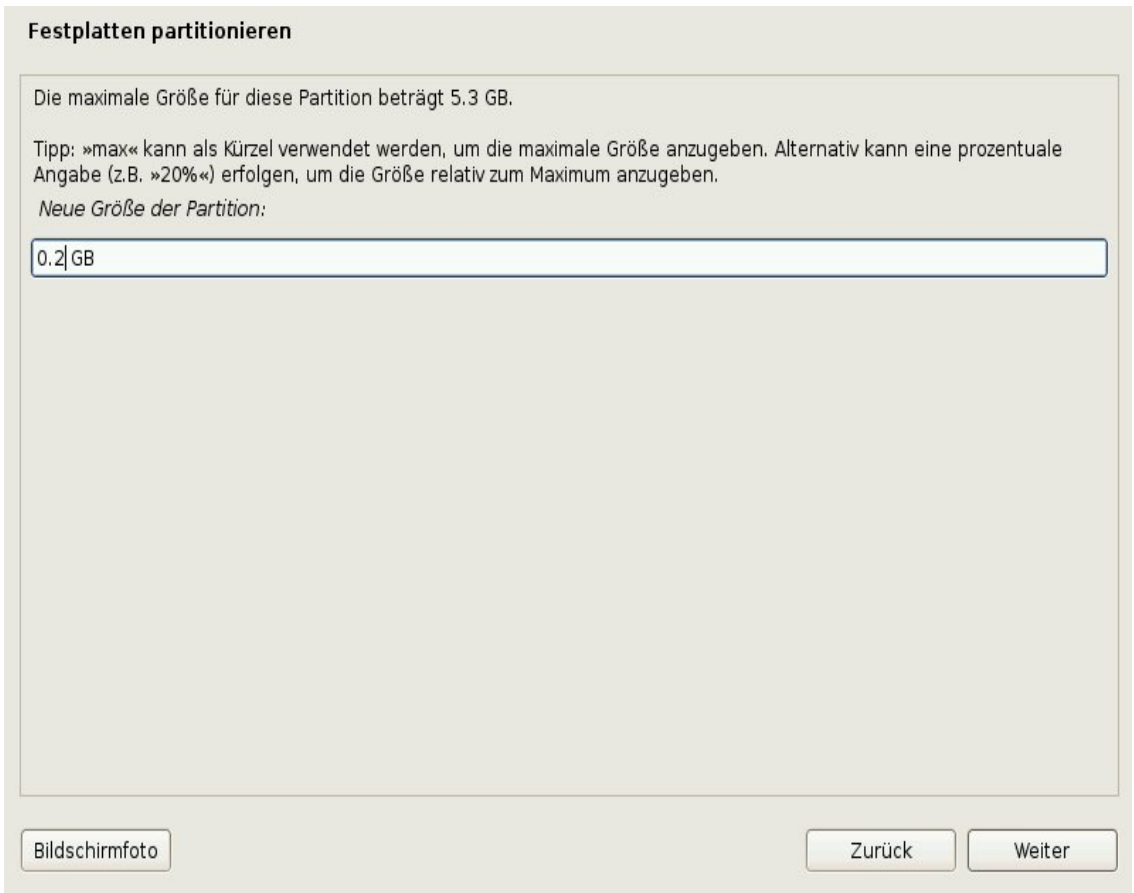
/boot -> Platte 1 -> 0,5 GB

/tmp -> Platte 1 -> 2,5 GB

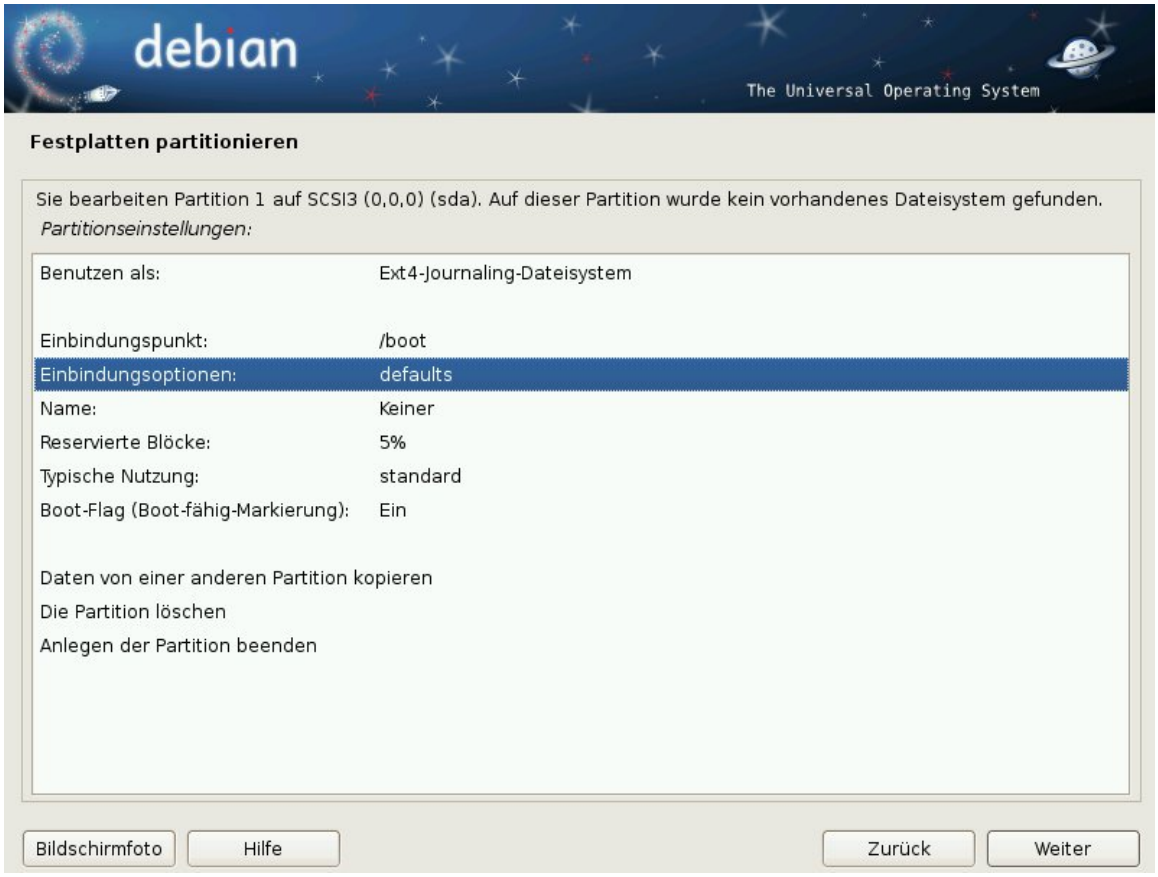
/swap -> Platte 2 -> 3,5 GB

Achtung: Auf den screenshots sind andere Größenangaben!


debian
The Universal Operating System
Festplatten partitionieren
Dies ist eine Übersicht über Ihre konfigurierten Partitionen und Einbindungspunkte. Wählen Sie eine Partition, um Änderungen vorzunehmen (Dateisystem, Einbindungspunkt, usw.), freien Speicher, um Partitionen anzulegen oder ein Gerät, um eine Partitionstabelle zu erstellen.
Geführte Partitionierung
Software-RAID konfigurieren
Logical Volume Manager konfigurieren
Verschlüsselte Datenträger konfigurieren
SCSI3 (0,0,0) (sda) - 5.3 GB ATA VBOX HARDDISK
> pri/log 5.3 GB FREIER SPEICHER
SCSI4 (0,0,0) (sdb) - 5.3 GB ATA VBOX HARDDISK
> pri/log 5.3 GB FREIER SPEICHER
Änderungen an den Partitionen rückgängig machen
Partitionierung beenden und Änderungen übernehmen
Bildschirmfoto
Hilfe
Zurück
Weiter

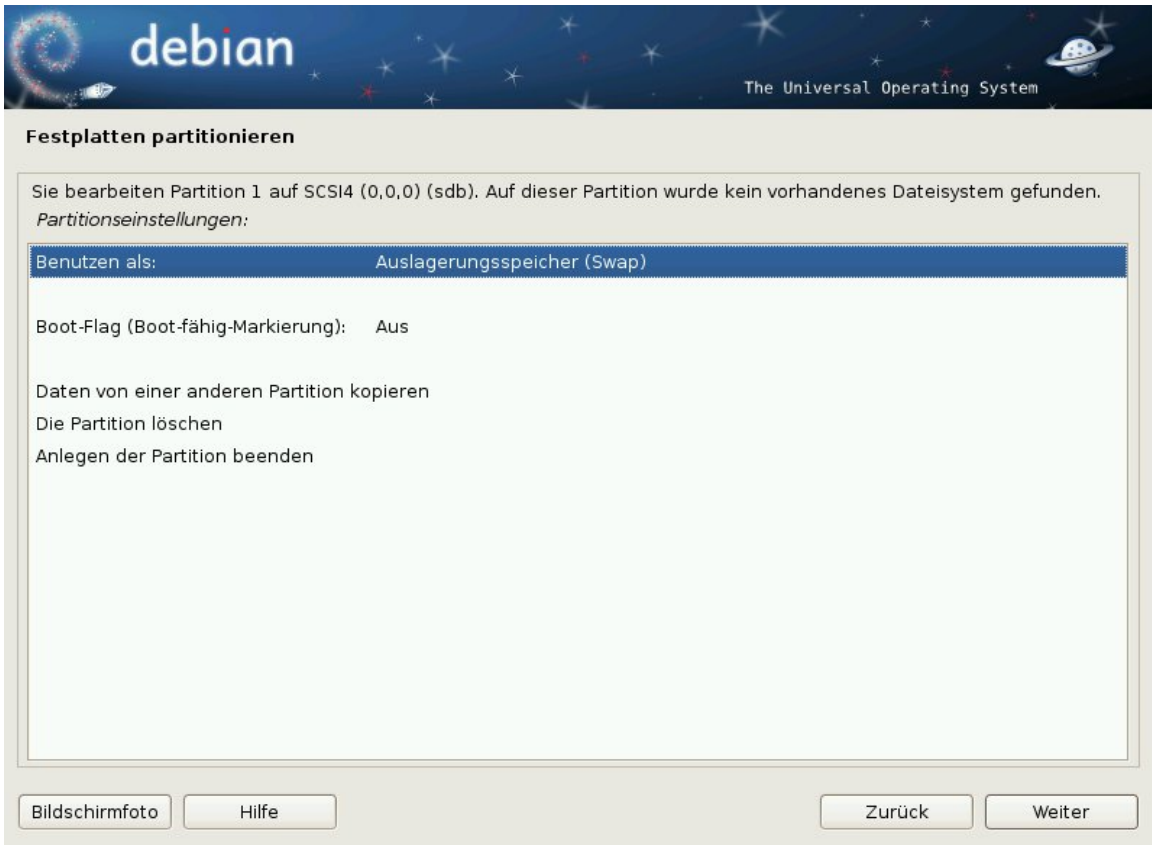

Festplatten partitionieren
Die maximale Größe für diese Partition beträgt 5.3 GB.
Tipp: »max« kann als Kürzel verwendet werden, um die maximale Größe anzugeben. Alternativ kann eine prozentuale Angabe (z.B. »20%«) erfolgen, um die Größe relativ zum Maximum anzugeben.
Neue Größe der Partition:
0.2 GB
Bildschirmfoto
Zurück
Weiter

Die Boot-Partition anlegen

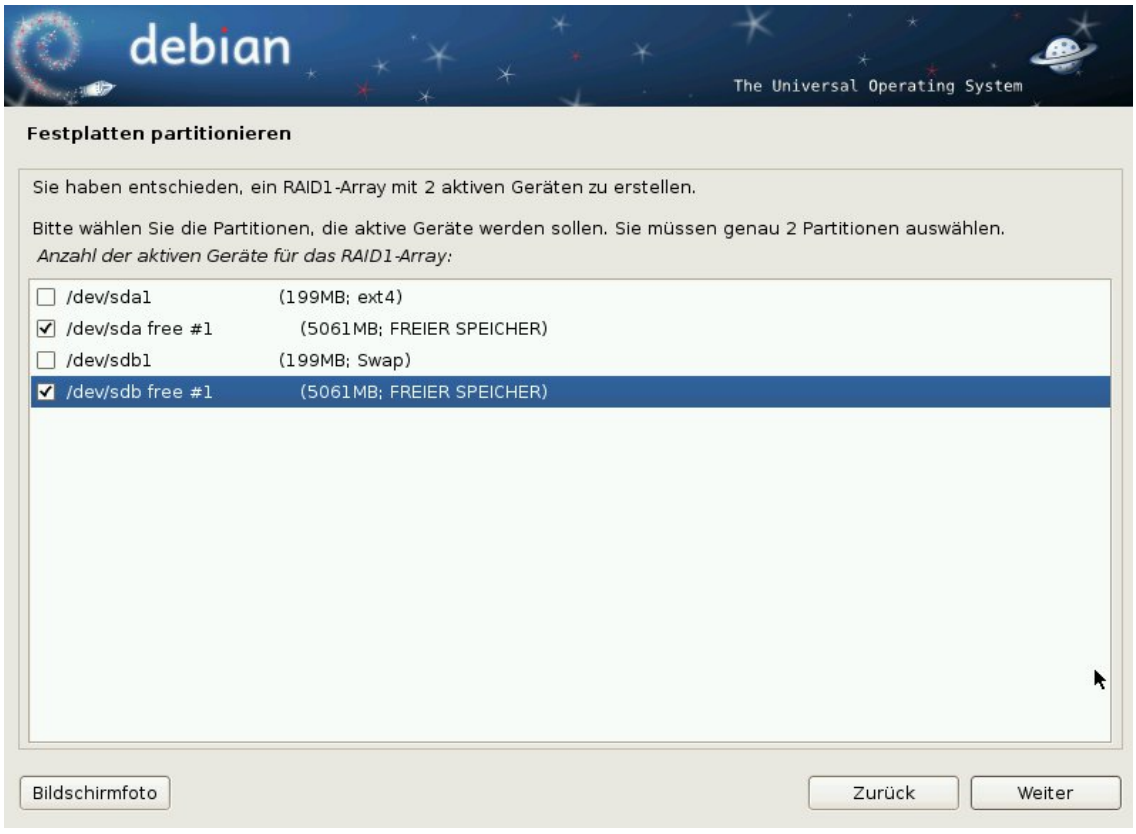


Dasselbe für /tmp wiederholen. Bei den Einbindungsoptionen für /tmp zusätzlich noexec,nosuid,nodev anklicken

Die Auslagerungsdatei swap anlegen (auf der 2. physikalischen Platte)



Wenn /boot, /swap und /tmp angelegt sind, geht es weiter mit der RAID-Erstellung:



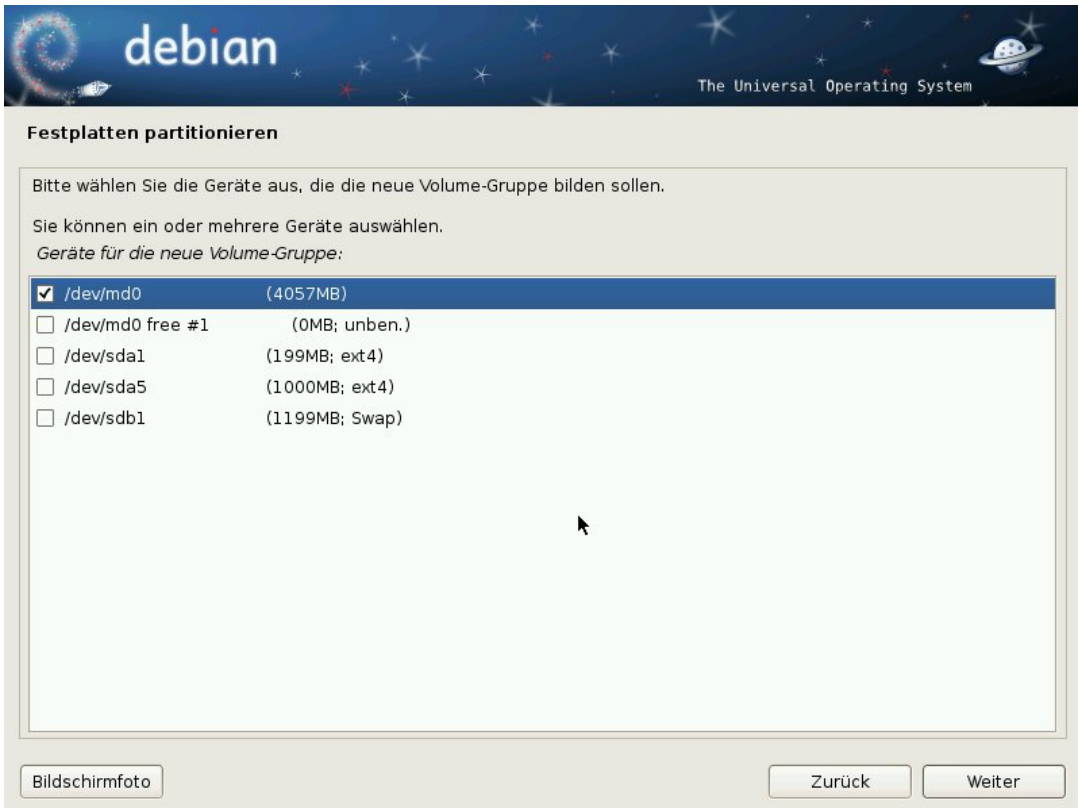
Software-raid konfigurieren -> ja -> MD-Gerät erstellen -> RAID1 -> 2 -> 0 -> dann die beiden freien Bereiche auswählen

-> ja -> Fertigstellen

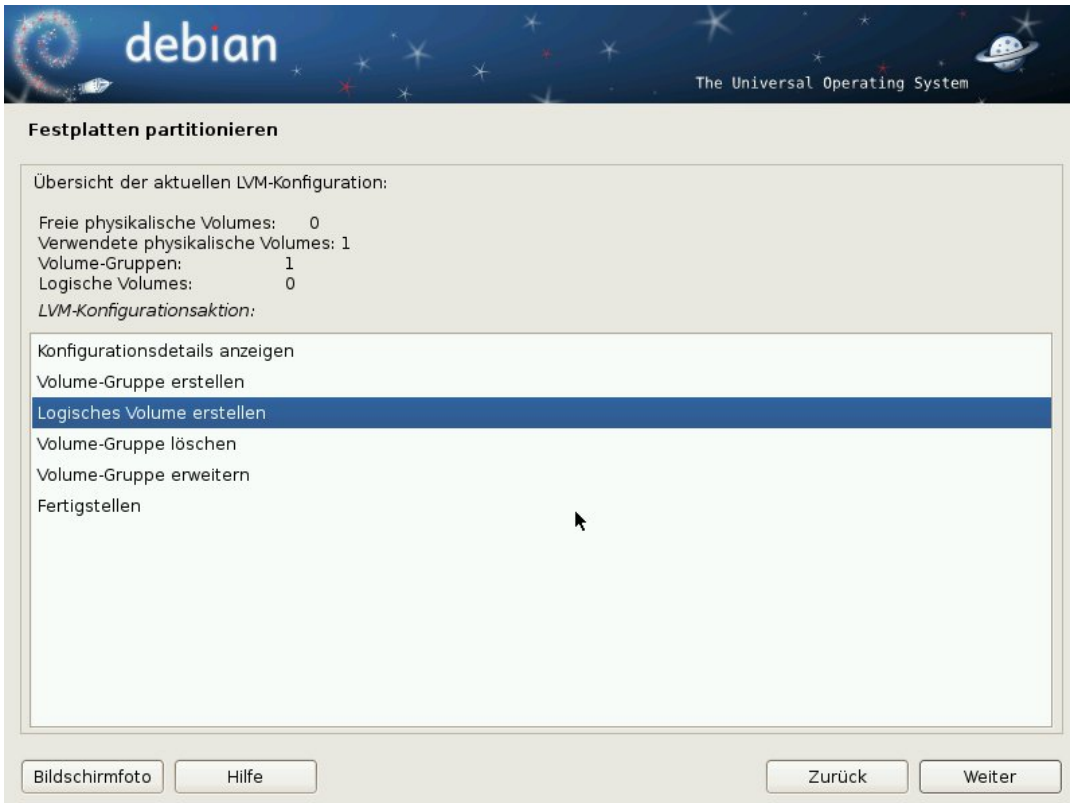
Jetzt habe ich folgende Situation:

3 Partitionen direkt auf der Festplatte und ein RAID1-Array. Nun fehlen mir aber noch die anderen Partitionen, welche auf das RAID sollen.

Dazu lege ich auf dem RAID ein LVM an und auf diesem ein logisches Volumen (bei mir vg0).



Auf vg0 erstelle ich die logischen Volumen

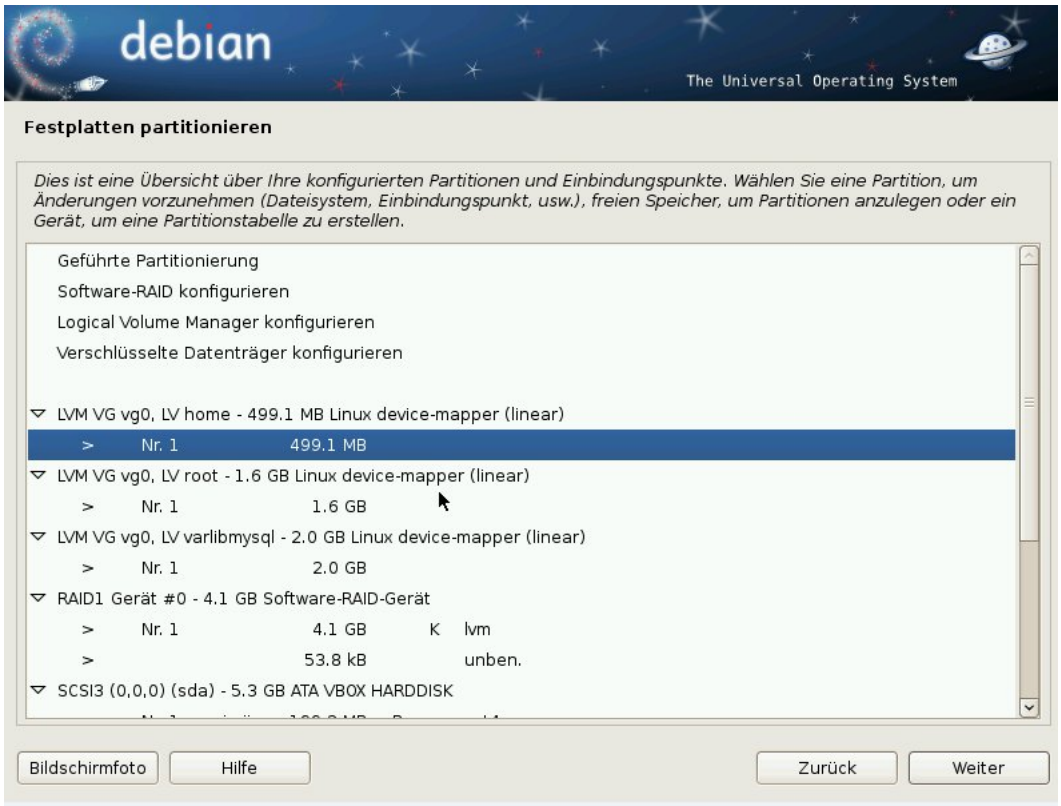


für die vorgesehen (also 2, ich hatte erst überlegt, noch einige Extrapartitionen zu erstellen, bin davon aber wieder abgerückt) Einbindungspunkte:

var -> 100 GB (hier liegt u.a. nach die Datenbank)

root -> Rest

Danach werden die eben erstellten logischen Volumen als Partition angelegt und eingehängt.



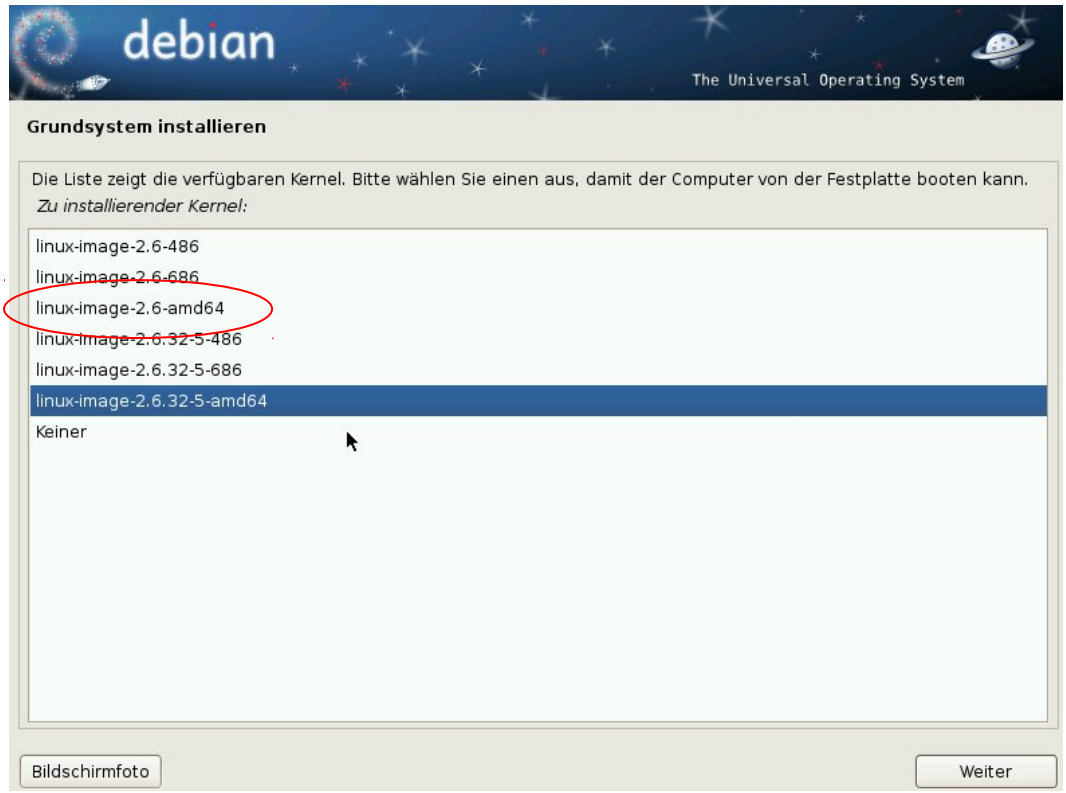
Als Einhängepunkt wird wie folgt zugeordnet:

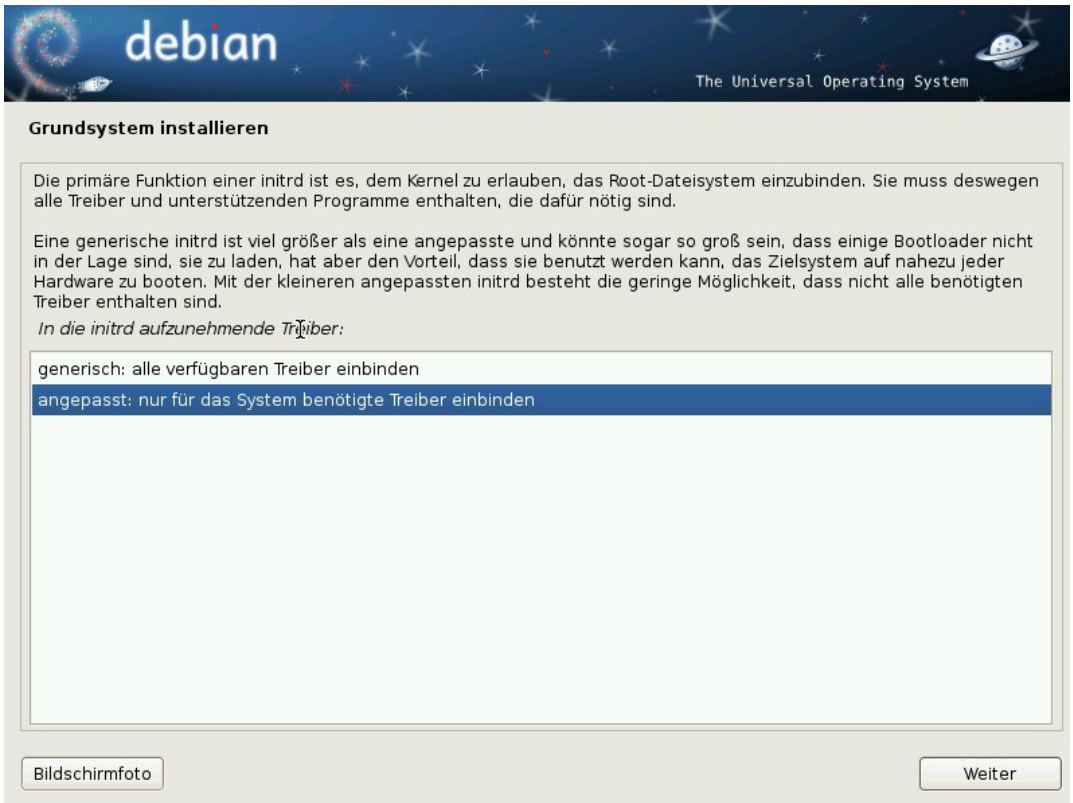
root als /

var als /var

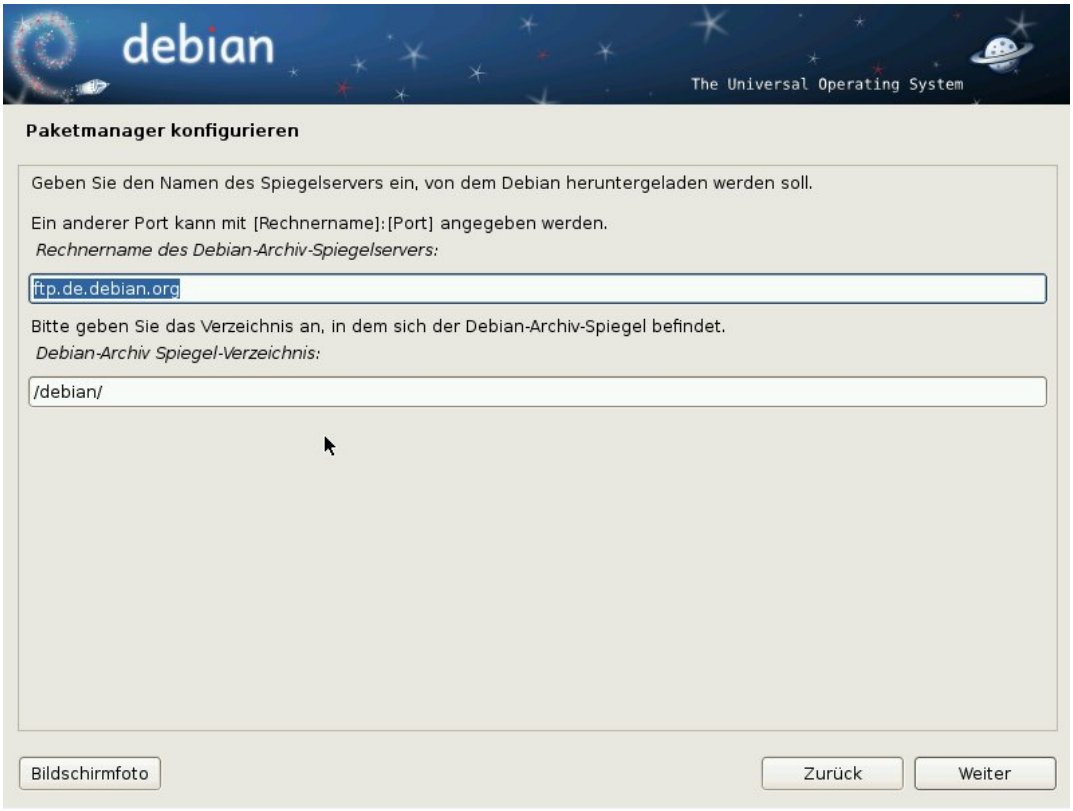
Dateisystem ist bei allen ext4.

Auf die Verschlüsselung der Laufwerke verzichte ich, da der Server sonst nicht ohne Eingriff alleine startet (beispielsweise nach Stromausfalls).


debian
The Universal Operating System
Grundsystem installieren
Die Liste zeigt die verfügbaren Kernel. Bitte wählen Sie einen aus, damit der Computer von der Festplatte booten kann.
Zu installierender Kernel:
linux-image-2.6-486
linux-image-2.6-686
linux-image-2.6-amd64
linux-image-2.6.32-5-486
linux-image-2.6.32-5-686
linux-image-2.6.32-5-amd64
Keiner
Bildschirmfoto
Weiter


debian
The Universal Operating System
Grundsystem installieren
Die primäre Funktion einer initrd ist es, dem Kernel zu erlauben, das Root-Dateisystem einzubinden. Sie muss deswegen alle Treiber und unterstützenden Programme enthalten, die dafür nötig sind.
Eine generische initrd ist viel größer als eine angepasste und könnte sogar so groß sein, dass einige Bootloader nicht in der Lage sind, sie zu laden, hat aber den Vorteil, dass sie benutzt werden kann, das Zielsystem auf nahezu jeder Hardware zu booten. Mit der kleineren angepassten initrd besteht die geringe Möglichkeit, dass nicht alle benötigten Treiber enthalten sind.
In die initrd aufzunehmende Treiber:
generisch: alle verfügbaren Treiber einbinden
angepasst: nur für das System benötigte Treiber einbinden
Bildschirmfoto
Weiter

Netzwerkspiegel -> ftp



unfreie Software -> ja -> ... ...Standardsystemsoftware und ssh Server.

Nicht SQL-Datenbank auswählen, dahinter versteckt sich nicht MySQL, sondern PostgreSQL .

Die grafische Umgebung lasse ich weg. Hier würde GNOME mit einer nahezu kompletten Büro- und Spieleumgebung eingerichtet, dass will ich nicht.

Entgegen aller Vernunft werde ich später eine kleine schlanke grafische Umgebung nachinstallieren.

6. Systemeinrichtung

6.1. Grundeinrichtung

Als erste Maßnahme im BIOS die bootprioritäten wieder ändern und das BIOS mit einem Passwort versehen.

Dann nach der Installation die Softwarequellen einrichten:

Also Anmeldung an der Konsole mittels

USERNAME + Passwort (dieses ist bei der Eingabe nicht sichtbar). Danach werden wir zum Superuser root mittels

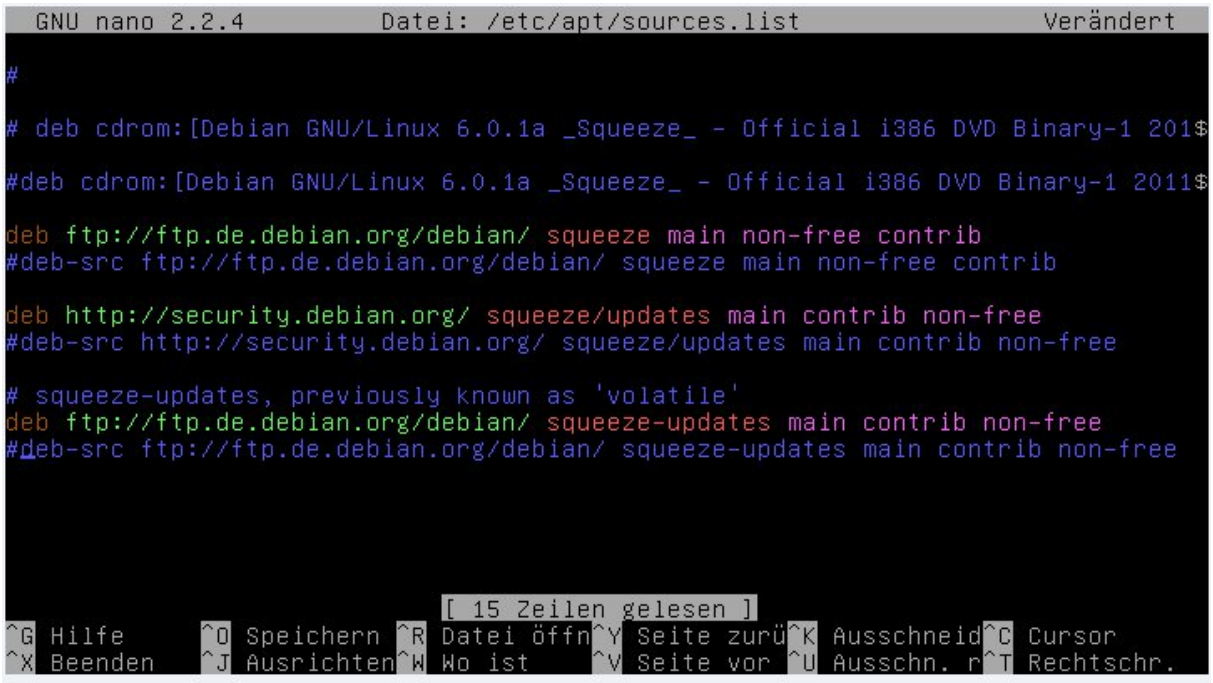
su – *(ein Leerzeichen zwischen su und -)*

rootpasswort

nano /etc/apt/sources.list

Auskommentieren der nicht benötigten Zeilen mit einer #

So soll es dann aussehen:



Mit *STRG + O* und *ENTER* speichern und *STRG + X* beenden.

Dann die Paketlisten und das System aktualisieren

apt-get update

apt-get upgrade

Weitere Software einspielen:

apt-get install xorg icewm synaptic chkconfig mc bzip2 sensord freeipmi-tools rkhunter

Ja, ich höre das Schreien!!!!

„Was hat eine grafische Oberfläche auf einem Server zu suchen? Blasphemie!“

Egal, sie wird sicherlich dem einen oder anderen user das Leben etwas leichter machen. Ausserdem wird der X-Server und icewm später nicht automatisch gestartet.

Zur Erklärung: xorg ist der grafische Server, icewm die grafische Oberfläche und synaptic ein (grafisches) Paketverwaltungsprogramm.

reboot

Anmelden als root:

root

rootpassword

grafische Oberfläche starten:

startx

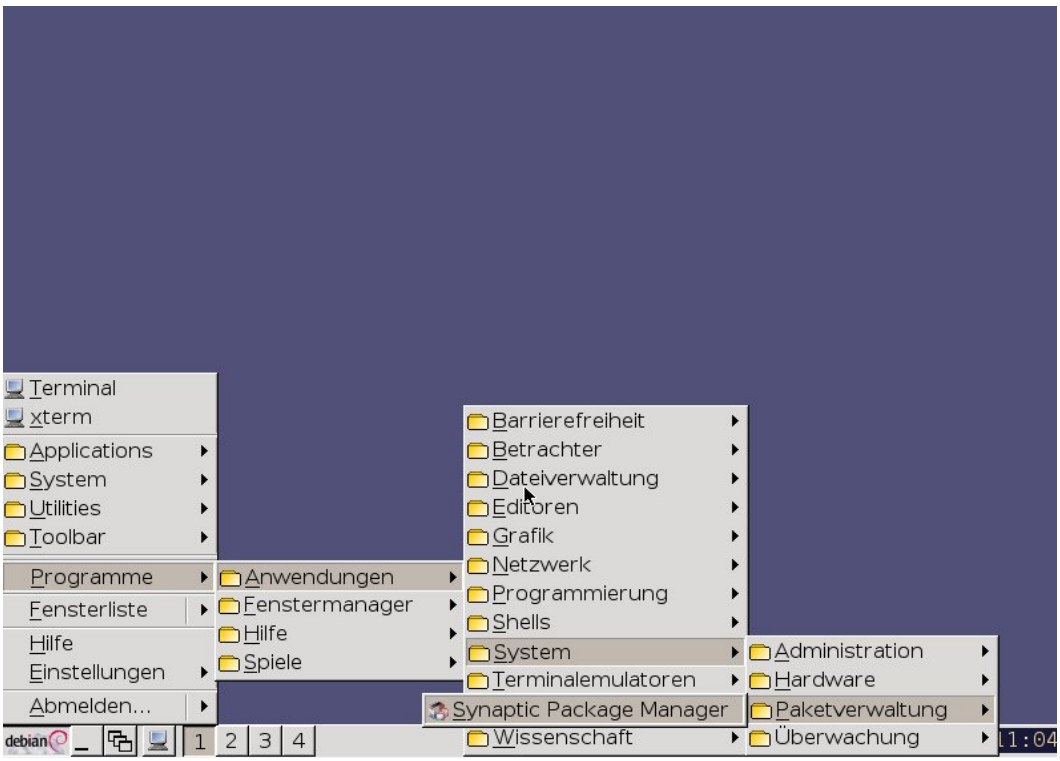
Es begrüßt uns ein wunderschöner lila Desktop!

6.2. Weitere Software entfernen, installieren und konfigurieren

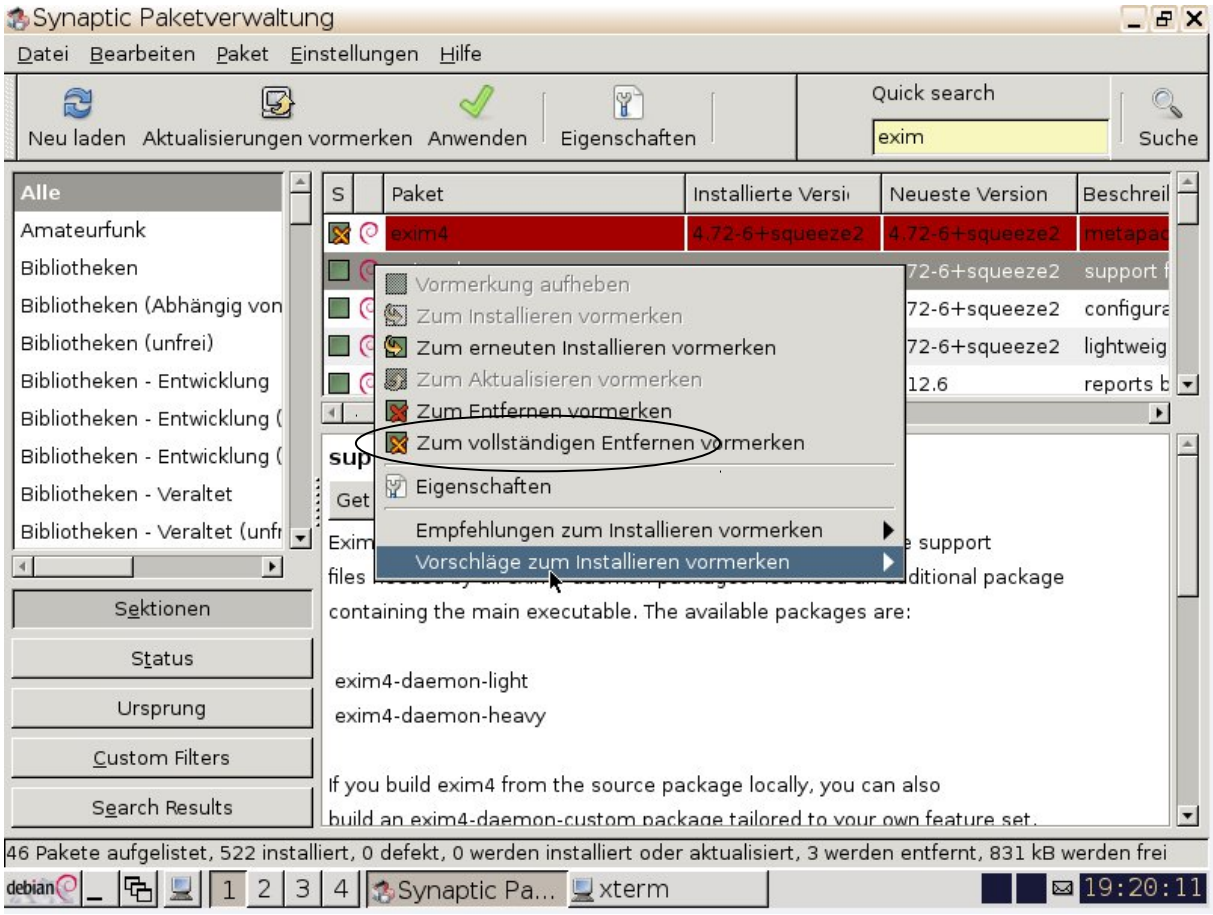
6.2.1. Software entfernen:

Als überflüssige Dienste sind noch aktiv: nfs-common, portmap und exim. Verraten hat das der Befehl *chkconfig*, welcher als root auf einer Konsole ausgeführt wurde.

Synaptic starten:



Diese Zeichenfolgen (nfs, portmap) in das Suchfenster eintippen, rechte Maustatste auf das installierte Paket (grüne Markierung bedeutet: ist installiert) und vollständig entfernen. Exim kann hier nicht entfernt werden. Dies erfolgt später im Zuge der Mailservereinrichtung.



6.2.2. Software installieren

apcupsd, gapcmon, fail2ban, mysql-admin, mysql-server (die Grundeinrichtung erfolgt während der Installation), automysqlbackup, anacron, smartmontools, leafpad, thunar

6.2.3. mc einrichten:

konsole starten, mc -> F9 -> Optionen -> Konfiguration -> mit Pfeiltasten bis “internen Editor verwenden”, mit space ankreuzen, weiter bis speichern.

6.2.4. Konfigurieren der fstab

Mit leafpad die Datei /etc/fstab öffnen und die Zeile mit dem CD-ROM mittels # auskommentieren oder löschen. Sonst werden keine USB-Laufwerke erkannt, ist ein bekannter bug.

Datei speichern und schliessen

6.2.5. Konfiguration von apcupsd:

mit leafpad die Datei /etc/apcups/apcupsd.conf öffnen und wie folgt anpassen (oder ersetzen)

```
## apcupsd.conf v1.1 ##
#
UPSNAME Meine-Server-APC
UPSCABLE usb
UPSTYPE usb
DEVICE
LOCKFILE /var/lock
ONBATTERYDELAY 6
BATTERYLEVEL 8  #  8% Restladung oder:
MINUTES 3 # 3 Minuten Restlaufzeit
TIMEOUT 0
ANNOY 300
ANNOYDELAY 60
NOLOGON disable
KILLDELAY 0
NETSERVER on
NISIP 192.168.0.2 #ggf. muss ich auch localhost rein
NISPORT 3551
EVENTSFILE /var/log/apcupsd.events
EVENTSFILEMAX 10
UPSCLASS standalone
UPSMODE disable
STATTIME 0
STATFILE /var/log/apcupsd.status
LOGSTATS off
DATATIME 0
```

Datei speichern und schliessen.

Analog dazu die /etc/default/apcupsd (letzte Zeile von no auf yes ändern)

```
# Defaults for apcupsd initscript

# Apcupsd-devel internal configuration
APCACCESS=/sbin/apcaccess
ISCONFIGURED=yes
```

apcups-dämon neustarten:

/etc/init.d/apcupsd restart

apcups testen: Dazu muss der apcups-dienst angehalten werden:

/etc/init.d/apcupsd stop

apctest

Jetzt können verschiedenste Dinge getestet und angezeigt werden.

Verhalten des Rechners bei Stromausfall simulieren:

in der oben beschriebenen /etc/apcups/apcupsd.conf das *BATTERYLEVEL* auf 98 setzen, den apcupsd neu starten und der UPS den Stromstecker klauen. Der Rechner sollte jetzt nach kurzer Zeit, begleitet vom Piepen der UPS normal runterfahren. Weiterhin sollte (wenn die Einrichtung wie weiter unten beschrieben fertig ist) in Eurem mailpostfach eine Warnmeldung auftauchen.

Nach dem Einstecken sollte der Rechner wieder hochfahren. Nicht vergessen, den Wert wieder zurück zu setzen!

Anmerkung! Ist noch zu kontrollieren, BIOS-Änderung erforderlich?

Antwort: Ja, ggf. im BIOS den AC-Power-loss auf always on setzen

6.2.6. postfix

Um Statusmeldungen und anderes per mail zugestellt zu bekommen, wird postfix installiert und so konfiguriert, dass keine externen mails empfangen werden, aber interne mails nach außen über einen beliebigen mailprovider verschickt werden können. Bei der Gelegenheit wird gleich exim deinstalliert.

Ggf. rüste ich das Empfangen von externen mails irgendwann nach, aber zum jetzigen Zeitpunkt wird es nicht benötigt und erspart das Einrichten eines Spamfilters und eines Virenscanners für den mailverkehr.

Mit einigen Anpassungen erfolgt die Konfiguration wie unter http://wiki.ubuntuusers.de/postfix beschrieben. Da einige Sachen providerspezifisch sind, hier keine detailierte Anleitung

Ergänzend zu der Ubuntu-Anleitung:

Die Datei /etc/aliases bearbeiten: #root darf keine mails bekommen, alle mails für root werden auf USERNAME umgeleitet

```
...
...
root: USERNAME
USERNAME : meineAdresse@googlemail.com
```

postalias /etc/aliases

als root ausführen, um die aliasesdb zu erstellen

6.2.7. smartmontools

in /etc/default/smartmoontools das automatische starten anschalten

6.2.8. md-admin

in /etc/mdadim/mdadmin.conf den Mailempfänger freischalten

6.2.9. mysql-server einrichten

/etc/mysql/my.cnf öffnen und die Zeile

```
bind    127.0.0.1
```

mittels # auskommentieren (Anmerkung: kontrollieren, ob max_packet_size auf einen vernünftigen Wert, also 8 oder 16 MB steht. Wenn nicht, anpassen)

mysql-admin starten, user -> root -> rechtsklick auf neuen host und entweder alles (Sicherheitsrisiko!) oder lokales Netz freigeben.

Wenn Thera-Pi danach erstmalig auf einem anderen Rechner eingerichtet wird, war's das schon mit MySQL. Ansonsten die Thera-Pi-Datenbank importieren, den Thera-Pi-Datenbankuser erstellen und ihm die Rechte an der Thera-Pi-Datenbank geben.

6.2.10. automatisches backup der Datenbank.

automysqlbackup und bzip2 wurden bereits installiert

cron-Job für die tägliche, wöchentliche (Sonnabend, kann geändert werden) und monatliche Sicherung wurde bei der Installation automatisch eingerichtet.

Backups rotieren wie folgt:

tägliche Backups-> wöchentlich

wöchentliche backups-> 5-Wochenrhytmus

monatliche backups -> nie (müssen händisch gesichert/entfernt oder sonstwas werden-> siehe postbackup

folgende Änderungen in der /etc/default/automysqlbackup

```
dbname="ptlwl"
```

so heißt meine Datenbank, welche ich sichern will

```
mailcontent="quiet"
```

Es wird nur eine mail verschickt, wenn es einen Fehler bei der Sicherung gab. Für's testen vielleicht auf „log" setzen, dann bekommt eine mail über den Verlauf des backups eine mail. Sieht dann ungefähr so aus:

```
================================================================================
AutoMySQLBackup VER 2.5
http://sourceforge.net/projects/automysqlbackup/
Backup of Database Server - ptLWL01
================================================================================
Backup Start Time Mo 15. Aug 20:50:43 CEST 2011
================================================================================
Daily Backup of Database ( ptlwl )
Rotating last weeks Backup...
Compression information for /var/lib/automysqlbackup/daily/ptlwl/ptlwl_2011-08-15_20h50m.Montag.sql.bz2
 /var/lib/automysqlbackup/daily/ptlwl/ptlwl_2011-08-15_20h50m.Montag.sql:
 2.014:1,  3.972 bits/byte, 50.35% saved, 348681282 in, 173134560 out.
--------------------------------------------------------------------------------
Backup End Mo 15. Aug 20:56:33 CEST 2011
================================================================================
Total disk space used for backup storage..
Size - Location
498M /var/lib/automysqlbackup
================================================================================
If you find AutoMySQLBackup valuable please make a donation at
http://sourceforge.net/project/project_donations.php?group_id=101066
================================================================================
```

```
MAIL_ADDR="root"
```

an diese Adresse wird obige mail geschickt, bei mir dann intern weitergeleitet an USER, welcher definiert ist als meineadresse@googlemail.com

```
comp=bzip2
```

die Datenbanksicherung wird mit bzip2 komprimiert. Bzip2 komprimiert stärker als das voreingestellte gzip, braucht dafür aber etwas länger. Da die Sicherung nachts abläuft, stört das nicht weiter.

#für später:POSTBACKUP="/etc/mysql-backup-post" hier werde ich wohl eine rsync-Syncronisation mit einem externen FTP-Server einrichten. Oder sowas in der Art.

6.2.11.cron

cron läuft standardmäßig um 6:25 morgens. Ich finde, dass ist eine ungünstige Zeit, das sollte etwas früher sein. Also die /etc/crontab öffnen die 6'en gegen eine 3 austauschen. Schon laufen die ganzen backups und Sicherheitstests um 03:25 Uhr und stören damit nicht.

6.3. System absichern

Erstmal: hier lesen!

Ich werde hier nicht versuchen, einen supersicheren Hochverfügbarkeitsserver aufzubauen.

Dies würde mir wohl auch nicht gelingen. Ich glaube auch nicht, dass ich oder meine Daten so wichtig sind, dass sich jemand die Mühe macht, die Daten zu stehlen oder zu zerstören. Einen „professionellen“ Angriff werde ich sowieso nicht abwehren können. Vielmehr geht es mir um scriptkiddies oder Gelegenheitshacker, welche sich mit den folgenden Maßnahmen ausreichend abwehren lassen sollten.

Zur Argumentation: Mein router hat eine firewall und ich bin dadurch sicher. Diese firewalls sind gut und im Allgemeinen auch sicher und für vieles auch ausreichend.

Dennoch kann nicht ausgeschlossen werden, dass:

- die routerfirmware einen Fehler oder eine Hintertür hat
- jemand in das WLan eindringt und damit hinter der firewall ist
- exploits auf anderem Wege (per mail / Stick über den Thera-Pi-Windowsrechner) in das Netzwerk kommen

6.3.1. sshd

root wird der Zugriff verboten, leere Passwörter werden verboten.

/etc/ssh/sshd_config bearbeiten und folgendes ändern/prüfen:

```
...
...
PermitRootLogin no
PermitEmptyPasswords no
```

http://wiki.ubuntuusers.de/SSH

6.3.2. fail2ban

dient gegen brute-force-Angriffe, indem nach einer bestimmten Anzahl von ungültigen (Anmelde)Versuchen die angreifende IP für eine gewisse Zeit geblockt wird.

Vorerst wird nur sshd abgesichert, da nix anderes (mysql kann nicht mit fail2ban gesichert werden) läuft:

http://www.root-on-fire.com/2011/06/29/howto-linux-server-mit-fail2ban-absichern/

6.3.3. rkhunter

Überprüft den Rechner regelmäßig auf rootkits

rkhunter –-update

rkhunter –propupd

rkhunter -c

eine Datei unterhalb von /etc mit dem Namen rkhunter.config.local erstellen und wie folgt füllen:

```
ALLOWHIDDENDIR=/dev/.udev
ALLOWHIDDENDIR=/dev/.initramfs
ALLOWHIDDENDIR=/dev/.mdadm
RTKT_FILE_WHITELIST="/etc/init.d/.depend.boot /etc/init.d/hdparm"
SCRIPTWHITELIST="/sbin/chkconfig"
MAIL-ON-WARNING=“root“
language=de
```

Es werden jetzt bei Durchlauf von rkhunter erst die Einstellungen aus der vorgegebenen rkhunter.config eingelesen und danach die spezifischen Einstellungen aus der rkhunter.config.local verwendet.

bei nächsten Aufruf von

rkhunter -c

dürften keine Warnungen mehr kommen

http://wiki.ubuntuusers.de/rkhunter

6.3.4. firewall einrichten

Ich werde nicht direkt in den iptables rumschreiben, sondern installiere das grafische Programm „firestarter“. Dies hilft mir, meine Regeln anzulegen.

Die Grobeinrichtung erfolgt mit einem Assistenten beim ersten Programmstart. Danach wechseln auf Richtlinie -> Richtlinie für eingehend und mit rechter Maustaste in das Feld „erlaube Dienst“ -> neue Regel

Hier dann aus den vorbelegten die benötigten (ssh, imap, ftp, smtp) auswählen und zusätzliche Regeln für die USV (port 3551) und MySQL (3306) erstellen. Fertig.

7. Was fehlt noch bzw. demnächst auf dieser Bühne!

- die Einrichtung von lmsensors. Noch leichte Probleme mit dem Lüfter des HP
- irgendwo hab ich noch meine Festplatten konfiguriert (Standby nach 2 Stunden). Muss nochmal raussuchen, wo das war.
- I-netanschluss bestellen
  - dynamische IP Einrichten
  - VPN-Zugang einrichten
  - den FTP-Speicherplatz nutzbar machen
  - automysqlbackup so einrichten, dass das backup direkt auf einen externen FTP-Server syncronisiert wird
  - externe Rechner auf die MySQL-Datenbank aufschalten.
- Webalyser einrichten zum Überprüfen des Serverzustandes mittels grafisch aufbereiteter logs

Fragen, Anregungen und Kritiken ausdrücklich erwünscht!

Michael [letzter3]