

## Partitionieren von Festplatten unter LINUX

---

oder

Wie und warum der Blödsinn?

In der letzten Zeit hatten wir an den BreLUG-Abenden oder im Maillister oft das Thema des Partitionieren der Festplatten unter Linux.

Dabei hab ich mich immer wieder gefragt: Verdammt noch mal, was haben die denn bloß? So schwierig ist das doch gar nicht. Dann fiel mir in den Linux-Newsgroups im Usenet auf, daß auch dort dieses Thema immer stärker auftrat. Speziell, seit die kommerziellen Distributionen immer größer und unübersichtlicher werden, seitdem immer mehr Anfänger auf der Linux-Bühne auftauchen und Halb- und Vollautomatismen die Partitioniererei in den Sand setzen, schallt die Frage ins Rund: Wie partitioniere ich meine Platte(n)?

Mein Unverständnis diesem Problem gegenüber brachte mich dazu, mal darüber nachzudenken, wieso das gerade in letzter Zeit so schlimm ist. Und ich glaube, ich habe einen der Hauptgründe gefunden. So unglaublich das klingt, aber ich denke, daß die Firma MicroSoft da dran schuld ist. Nein, ich trete hier nicht wieder einen Religionskrieg los. Das ist mein voller Ernst.

Wie war es denn früher? Die ersten Generationen von Linux-Ein- und Umsteigern waren Leute, die erstens mit der Commandline und zweitens mit Handarbeit ihr MS-DOS oder Win3 installiert haben oder installieren \*mußten\*. Damals war eine Festplatte relativ teuer, und man bekam sie in aller Regel in einem nicht betriebsbereitem Zustand in die Hand. Zuerst mußte mit speziellen Programmen (Ontrack oder Spedstor, erinnert sich noch jemand?) die Platte lowlevel-formatiert werden. Dann wurde der allseits gehaßte fdisk angeworfen und die Aufteilung (Partitionierung) angelegt und als letzter Schritt mittels format die Partitionen dem MS-DOS bedienbar gemacht. Erst viel später gab es das eine oder andere Installationsprogramm, daß diese Schritte automatisierte, vor allem, als der Waschmittel äh.... ATA/IDE-Bus sich durchsetzte. Mit MS-DOS 5 hatte der Anwender plötzlich keine Probleme mehr mit dem Thema und MS-DOS 6 stellte nicht mal mehr irgendwelche Fragen. Der Begriff 'Partitionieren' geisterte nur noch bei den Spezialisten herum. Der Anwender kam damit nicht mehr in Berührung, sprich er lernte dies erst gar nicht mehr. Win95/98/NT verdummten den Anwender noch stärker. Der allergrößte Teil der heutigen Linux-Einsteiger kommt aus der Win95/98/NT-Welt und hat natürlich keine blasse Ahnung von Betriebssystem-Strukturen und File-System-Aufteilungen. Die Betriebssystem-Struktur wird von MicroSoft geschickt versteckt und eine logische File-System-Aufteilung existiert erst gar nicht. Logisch ist des Rätsels Lösung: Woher soll es der Linux-Ein- oder Umsteiger denn wissen?

Unix ist ein Multi-User-System, das nach Bereichen für Systemdienste, Administrationsdienste, Grundeinstellungen, allgemeine Programme, Anwender und anderes streng unterscheidet. All dies gibt es in der Win95/98/NT-Welt bis zum heutigen Tage nicht. Gute Administratoren geben sich viel Mühe damit, aber es kann nur Flickwerk bleiben. Das fängt schon damit an, daß das gesamte System auf einer Partition, und was noch schlimmer ist, die ganze Platte umfassenden Partitionen installiert wird. Damit ist das Kind bereits rettungslos in den Brunnen gefallen und ertrunken.

Liebe Leute, kapiert und akzeptiert eines: UNIX ist in gar keiner Weise mit dem vergleichbar, was man bisher unter MicroSoft bedient hat. Wer sich privat mit UNIX beschäftigt, wird zwangsweise Systemadministrator. Wer nicht bereit ist, die Zeit und Mühe zu investieren, sich in das Filesystem und die Struktur von UNIX einzuarbeiten, kann jetzt seinen Sch....-Partition-Magic starten und bei Micro\$oft bleiben.

Noch interessiert, weiter zu lesen?

Gut, dann stelle ich folgende Anforderungen an Euch:

1. Ihr besitzt Englischkenntnisse die in etwa der Mittleren Reife entsprechen. Ohne diese Kenntnisse erleidet Ihr mit 100%iger Sicherheit Schiffbruch im Umgang mit UNIX.
2. Ihr seid bereit, jedwede Dokumentation zu lesen, welche Euch zum Thema angeboten wird.
3. Ihr drängt jede Erinnerung an Partition-Magic und MickeyWeich aus Eurem Gedächtnis.
4. Ihr installiert Euer UNIX unter Umständen einhundert

mal, bis Ihr es gefressen habt. Und das ohne zu jammern.

5. Angeblich muß man 8 Stunden pro Tag schlafen. Diese 8 Stunden kann man auch mit UNIX-Installation verbringen. Rotgeränderte Augen werden nur akzeptiert, wenn gleichzeitig das Wort LINUX auf der Netzhaut eingebrannt ist :-)

So. Immer noch interessiert?

Then let's go on!

Laßt uns etwas über das UNIX/LINUX-Filesystem reden. Ohne Verständnis für diese Systemstruktur kommt auch kein Verständnis für das Partitionieren einer Festplatte auf (das Gejammer aus der hinteren Reihe habe ich jetzt überhört).

Das Folgende bezieht sich auf den  
Filesystem Hierarchy Standard -- Version 2.0  
(FHS2.0) vom Oktober 1997. Entstanden ist dies aus der  
Linux Filesystem Structure -- Release 1.2  
vom März 1995.

Gute Linux-Distributionen liefern den FHS unter /usr/doc/fhs mit. Dies sollte von jedem einmal gelesen werden. Am neuesten Standard 'Linux Base System' wird gearbeitet, allerdings geht es da extrem schleppend voran, da sich einige Mitglieder dieses Gremiums nicht mehr einigen können. Es handelt sich hierbei um die Distributoren bekannter großer Linux-Distributionen. Ich will das Thema nicht ausweiten, da es schon ausreichend Religionskriege gibt. Soviel sei lediglich gesagt: Die kommerziellen Interessen kollidieren hier mit dem Willen, freie Software etwas zu standardisieren.

Der FHS gibt die Richtlinien für die Datei- und Verzeichnisstruktur eines Unix/Linux-Filesystems vor.

Sehen wir uns zuerst einmal das Hauptverzeichnis an:

```
/bin
/boot
/dev
/etc
/home
/lib
/lost+found
/mnt
/opt
/proc
/root
/sbin
/tmp
/usr
/var
```

und umreißen kurz, was und wofür denn die einzelnen hier dargestellten Unterverzeichnisse sind.

Um die folgenden Beschreibungen etwas übersichtlich zu gestalten, werden die einzelnen Unterverzeichnisse durcheine Reihe von Sternchen (\*\*\*\*..) getrennt. Es folgt der Name des Unterverzeichnisses in Form des kompletten Pfades. Dahinter in runden Klammern wird vermerkt, ob es sich gegebenenfalls um eine eigene Partition handelt (P) (wobei der Name in diesem Fall auch der mountpoint ist), oder zu welcher Partition das Verzeichnis gehört und welche Datenmenge zu erwarten ist.

```
*****
/bin (Root-Partition 5 MB)
```

beinhaltet die essenziellsten Befehle, die der Systemadmin (auch im Single-User-Mode) und auch die Anwender brauchen. Weitere Unterverzeichnisse sollten nicht unter /bin existieren. Hier eine Übersicht:

```
{cat, chgrp, chmod, chown, cp, date, dd, df, dmesg, echo, ed, false, kill, ln, login, ls, mkdir, mknod, more, mount, mv, ps, pwd, rm, rmdir, sed, setserial, sh, stty, su, sync, true, umount, uname}
```

Wer den einen oder anderen Befehl nicht kennt, sollte sich mit 'man Befehl' schlau machen. Was \*ich\* gerne mache: Am Sonntagmorgen bewaffnet mit einer Tasse Kaffee in das Verzeichnis /usr/man gehen und man-pages lesen. Das kann sehr informativ sein.

Hier kommen ca. 5MB zusammen.

\*\*\*\*\*  
/boot (P 20 MB)

beinhaltet die statischen Dateien des Bootloaders. Dazu gehört bei den meisten Distributionen auch der Kernel selbst. Für /boot wird immer eine eigene Partition mit ca. 20MB angelegt. Diese Partition muß in aller Regel innerhalb der ersten 1024 Zylinder der Festplatte liegen. Das Bios des Rechners ist an dieser Vorgabe schuld.

\*\*\*\*\*  
/dev (Root-Partition 0.1 MB)

beinhaltet die Einträge für die Geräte-Dateien. /dev wird der root-partition zugeschlagen.

\*\*\*\*\*  
/etc (Root-Partition 2MB)

beinhaltet die Konfigurationsdateien des Rechners, vor dem Ihr sitzt. Auch /etc wird grundsätzlich der root-partition zugeschlagen, da in /etc auch grundlegende Konfigurationsdateien liegen, die direkt nach dem booten des Kernels benötigt werden.

\*\*\*\*\*  
/home (P ?MB)

beinhaltet die Anwender-Heim-Verzeichnisse (home-directories). Auf einem privatem Linux-Rechner wird man selten mehr als 2-3 home-directories finden. Nur ist dadurch nichts über die benötigte Datenmenge ausgesagt. Es gibt Leute wie mich, welche Daten zu bestimmten Gebieten sammeln. Da können 2 Gbyte schnell zu wenig sein. Andere schreiben nur Texte, da reichen 100MB Jahre. Für /home wird auf jeden Fall eine eigene Partition angelegt. Also hier erst mal hinsetzen und darüber nachdenken, was ich denn mit dem Rechner mache und was ich wohl an Daten erzeuge. Ich rate hier dringend, sich für dafür Zeit zu nehmen. Auch mal eine Nacht darüber schlafen.

\*\*\*\*\*  
/lib (Root-Partition 10MB)

beinhaltet die essenziellsten shared-libraries (also Bibliotheken), welche zum booten und zur Lauffähigkeit des root-Filesystems notwendig sind. Zusätzlich befinden sich in /lib auch die Kernel-Module. /lib wird immer der Root-Partition zugeschlagen.

\*\*\*\*\*  
/mnt

ist nur ein mountpoint für zu montierende Geräte (devices). Dieser mountpoint benötigt keine zuberücksichtigende Datenmenge. Bei diesem mountpoint werden übrigens von den verschiedenen Distributionen verschiedenen Philosophien verfolgt. Die einen setzen /mnt/cdrom, /mnt/floppy und /mnt/hd, die anderen trennen nach /mnt und /cdrom und /floppy. Welche hier die bessere Lösung ist, kann in ewig langen Streits im Usenet verfolgt werden. Für unser Thema ist es ohne Belang.

\*\*\*\*\*  
/opt (P 300MB)

beinhaltet zusätzlich installierte Anwendersoftware, welche mit den Distributionen geliefert wird. Bekannte Größen sind hier die Desktops KDE und Gnome, aber auch Netscape und anderes findet sich hier. Aber auch hier sind sich die Distributoren nicht einig. Manche installieren Gnome auch nach /usr. Man kann sich also nicht sicher sein. 200Mbyte kommen hier aber leicht zusammen, sicherheitshalber sollte man aber auf 300MB gehen, man weiß ja nicht, was die nähere Zukunft bringt.

\*\*\*\*\*  
/proc

ist nur ein Fenster zum Kernel. Darin hat nur der Spezialist was zu suchen. Wenn jemand nicht weiß, was man in /proc machen kann, läßt er dringend die Finger davon. Die notwendige Datenmenge geht gegen Null.

\*\*\*\*\*  
/root (Root-Partition 20 MB)

-----

ist das home-directory für den root (nicht zu verwechseln mit dem root-verzeichnis, welches das Hauptverzeichnis der Festplatte ist!). Da man als root nicht im System arbeiten soll, sondern als normaler User, hält sich der notwendige Platz für /root auch in Grenzen. Aber es kommt natürlich vor, daß der root mal dies und das als root machen will oder muß. So scheinen mir 20MB völlig ausreichend. Es wird wohl eher viel weniger wirklich notwendig sein. Daher hier auch wieder der Zuschlag zur Root-Partition.

\*\*\*\*\*  
/sbin(Root-Partition 5MB)

beinhaltet zusätzlich zu /bin zum Booten notwendige Binärdateien. Auch nur für den root ausführbare Dateien finden sich hier. Grob gesagt dient /sbin der Systemadministration. Der notwendige Platz ist gering und bewegt sich bei ca. 5MByte. Weitere Unterverzeichnisse sollten nicht in /sbin existieren. /sbin liegt immer in der Root-Partition.

\*\*\*\*\*  
/tmp (P 30MB)

hält temporäre Dateien, welche entweder vom System oder anderen Programmen während des Betriebes erzeugt werden. Es kann dabei durchaus einiges auflaufen, sodaß 30MB nicht zu klein sind. Auch 50MB sind nicht unbedingt zu groß gewählt. Auf jeden Fall ist für /tmp eine eigene Partition anzulegen. So wird verhindert, daß, sollte das System einmal durchdrehen, die Root-Partition vollläuft und einen mit 'Device out of space' überrascht. Denn sowas ist einfach nur widerlich. Ein Trost: Unix/Linux läßt eine Notreserve auf der Root-Partition frei, damit sich wenigstens noch der root einloggen und die Schweinerei aufräumen kann.

\*\*\*\*\*  
/usr (P 1000MB)

ist der zweite Hauptteil des Filesystems. Hier ist der größte Teil der Software, Bibliotheken, das Entwicklungssystem, Kernelquellen, das X Window System und anderes installiert. Eine genauere Aufschlüsselung spar ich mir hier, dafür lest Ihr die FHS durch! Hier kommen locker 800-1000MB (also 1 GByte) zusammen. Sollte man den Wahnsinn besitzen, mal eine Suse- oder RedHat-Distribution \*komplett\* zu installieren, reicht selbst das nicht mehr. Da sind dann sicherheitshalber schon 2 GByte notwendig. Dies ist aber nur der theoretische Fall und kommt ansich nicht vor. (Ok, von einem Fall hab ich mal gehört, der ist dann aber auch prompt in ein 'out of space' gerannt und hat von vorne angefangen) Für /usr ist in jedem Fall eine eigene Partition anzulegen!

\*\*\*\*\*  
/var (Root-Partition 50MB)

Variablen Daten. Prozeß-Accounting, Logfiles, Lockfiles, mail, Druckerspools und mehr findet sich hier. Das Package-Management der großen Distributionen hält hier auch seine Datenbanken vor. So können tatsächlich leicht 30-50MB oder sogar mehr zusammen kommen. Bei einer privaten Linux-Kiste stabilisiert sich das aber sehr schnell und daher wäre eine eigene Partition nicht herun. Auch hier lest Ihr wegen der genauen Aufschlüsselung mal den FHS durch. Mit /var/log und /var/spool kommt Ihr früher oder später sowieso unweigerlich in Berührung.

\*\*\*\*\*  
\* Merke: /bin, /sbin, /etc, /dev und /lib bekommen  
\* nie eine eigene Partition!  
\*\*\*\*\*

Damit hätten wir es im Großen und Ganzen schon. Halt, da wäre noch eine Sache: /usr/local

Wer gerne viel Programme aus dem INTERNET in Form von Quellcode zieht, selbst compiliert und installiert, kommt um eine eigene Partition für /usr/local nicht herum. Denn diese Programme werden grundsätzlich nach /usr/local installiert. Die Hierarchie dieses Verzeichnisses lest Ihr vor Benutzung ebenfalls im FHS nach. Über die notwendige Größe dieser Partition läßt sich diskutieren. Der eine hat da 400MB in der Benutzung, der andere braucht nur 50MB. Wenn man sich sicher ist, /usr/local extrem wenig oder gar nicht zu brauchen, schlägt es der /usr partition zu. Aber auch dann würde ich sicherheitshalber mit 20-30MB rechnen. Angelegt wird diese Verzeichnishierarchie in jedem Fall bei der Installation.

\*\*\*\*\*

Nun machen wir eine kleine Übersicht:

Partitionen	Inhalt	Größe	Gesamt
-------------	--------	-------	--------

-----

```

/ (Root) /bin    5 MB
         /dev   0.1 MB
         /etc    2 MB
         /lib   10 MB
         /root  20 MB
         /sbin   5 MB    mit Reserve
         /var   50 MB    ca. 100 MB

/boot    /boot  20 MB        20 MB
/opt     /opt  300 MB       300 MB
/tmp     /tmp  30 MB        30 MB
/usr     /usr  1000 MB     1000 MB

```

-----  
ergibt erstmal ca 1.5 GB

```

Hinzu kommen die unklaren
/home      und??? 270 MB
/usr/local.  ??? 100 MB

```

Hier der ultimative Rat: Eine 2 GByte-Platte ist genau der richtige Einstieg in ein Linux. Die übrigen 500MB lassen sich sehr schön auf /home mit 270MB und /usr/local mit 100MB aufteilen.

Oooooops, da fehlen doch 130 MB!?!?

Genau, die brauchen wir nämlich für die Swap-Partition. Die ist mit 128MB genau richtig bemessen. Mehr braucht nur jemand, der einen fetten Server mit 1Gbyte-RAM oder so fährt.

So ergibt sich eine 2Gbyte-Platte mit

```

/boot
/
swap
/opt
/tmp
/usr
/home
/usr/local.

```

Dies sind dann 8 Partitionen. Diese legt man dann folgendermaßen an: /boot, / und swap als primäre Partitionen, dann eine erweiterte Partition mit den folgenden logischen Partitionen für /opt, /tmp, /usr, /home und /usr/local.

Ein paar Grundlagen: Auf einer Platte lassen sich maximal 4 primäre Partitionen anlegen. Wer mehr braucht, legt drei primäre und eine erweiterte Partition an. Ein SCSI-System kann maximal 15 Partitionen, ein ATA/IDE-System 63 Partitionen enthalten (vielleicht liefert mir mal irgendwann einer die Begründung dafür?). Die maximale Partitionsgröße beträgt derzeit ca. 2.2TB, also 2.2 Terabyte. Allerdings kann eine Datei auf einer 32bit-Maschine maximal 2Gbyte sein. (Gibts dafür eigentlich jetzt einen Patch?)

Beschäftigt Euch mit dem fdisk (das Gejammer aus dem Hintergrund habe ich jetzt wieder nicht gehört). Ihr arbeitet mit UNIX/Linux. Da wird der fdisk verwendet und sonst nichts. Auch der cfdisk, der seit einiger Zeit bei den Distributionen mitgeliefert wird, ist mit Vorsicht zu genießen.

Der Disk-Druid oder wie diese anderen Tools alle heißen, sind für DAUs und Weicheier, die sowieso besser bei Gates bleiben.

Der fdisk hat eine derart hohe Funktionalität, daß es jedem MickeyWeich-Looser nur die Tränen in die Augen treibt. Und die 'sogenannte' Unbedienbarkeit kann ich beim besten Willen nicht nachvollziehen. Wenn man sich merkt, daß man mit dem 'w'-Kommando erst den ultimativen Schreibvorgang einleitet und alles andere nur Voreinstellungen sind, hat man schon gewonnen.

Noch eine Sache, die immer wieder gefragt wird: Wie kann man Partitionen im Linux vergrößern oder verkleinern? Antwort: Überhaupt nicht. In einem laufenden UNIX-System geht sowas bis heute nicht. Und ich nehme an, dies wird auch nie gehen. (Angeblich soll es tatsächlich ein Tool unter Linux geben, was das könnensoll, aber die Berichte erzählen hauptsächlich von Problemen). Der nächste, der mir hier wieder mit Partition-Magic und Windows kommt, sage ich: Linux != Windows. Vergeßt es. Und wenn der Partition-Magic von DOSE oder Windows aus Linux-Partitionen manipulieren kann, dann you are at your own.

Folgende man-pages sind zu lesen:

fdisk,  
fstab,  
mount,  
df,  
mke2fs  
und dazu noch  
/usr/doc/Linux-mini-HOWTOs/Partition,  
/usr/doc/Linux-mini-HOWTOs/Large-Disk,  
/usr/doc/Linux-HOWTOs/Bootdisk-HOWTO,  
/usr/doc/Linux-HOWTOs/Multi-Disk-HOWTO  
und die FHS.

Am besten ausdrucken und in einem Aktenordner bereit halten.

Noch ein Thema, welches immer wieder gefragt wird: Was hat es mit den 'inodes' auf sich.

Beim Erzeugen des Filesystems wird manchmal nach der Inode-Dichte (richtiger wäre i-node) gefragt. Legt man ein Filesystem von Hand an (mk2efs o. ä.), kann man auch die Inode-Dichte angeben. Nun, ich versuch das mal populär zu erklären. Die Inode-Dichte bestimmt die, man könnte sagen, am meisten genutzte Dateigröße, man könnte auch sagen durchschnittliche Dateigröße, welche auf dem jeweiligen Filesystem zum Einsatz kommt. Bei der per default gewählten Inode-Dichte von 4KB (4096byte) setzt das Filesystem voraus, daß alle Dateien ca. 4KByte groß sind. Wenn ich nun in dieses Filesystem nur Dateien von ca. 2KB fülle, kann ich nur die Hälfte der Platte mit Dateien füllen. Auf der Platte wäre zwar noch Platz, aber ich habe alle Inodes verbraucht. Jede Datei (auch Verzeichnis) benötigt eine Inode. Nehme ich durchschnittlich aber ca. 8KB große Dateien, ist die Platte zwar komplett auszunutzen, aber ich habe noch Inodes frei, was auch nicht gut ist. Die Inodeeinträge benötigen nämlich auch Platz, und der wäre nun nutzlos. Bei einem Mischmasch an Dateien auf einem privaten Linux-Rechner hat sich eine Inode-Dichte von 4KB bewährt. Hat man allerdings vor, ein Filesystem hauptsächlich für Datenbanken zu verwenden, wo große Dateien die Hauptsache sind, kann man bis auf 16KB Dichte gehen. Umgekehrt sollten für ein Filesystem, welches einem Newsserver dient, nur 2KB verwendet werden. Hier ist ja mit vielen kleinen Dateien zu rechnen.

Die nun logischerweise aufkommende Diskussion: "... ach ist das alles umständlich, kann man denn nicht nur 3 Partitionen (/boot, swap und /) anlegen?" kann ich kurz anschnitten. Natürlich geht das. Warum auch nicht. Die Linux-Kiste wird aber beträchtliche Geschwindigkeitseinbußen (oder in neudeutsch performance-problems) haben. Denn auch Linux verwaltet kleine Partitionen viel schneller als Monsterpartitionen. Beim Updaten oder Upgraden eines Systems ist es sehr vorteilhaft und beruhigend, wenn man zum Bleistift die /usr/local-Partition abmontieren kann. Der lieben Handarbeit kann so nichts passieren. Ebenso kann es vorteilhaft sein, bestimmte Partitionen mit verschiedenen Inode-Dichten (siehe weiter oben) zu fahren. Es kann auch durchaus ekelhaft sein, wenn der Druckerspoolbereich vollgemüllt wird und dann, wenn /var keine eigene Partition ist, das root-Verzeichnis voll läuft. Es gibt sicher noch den einen oder anderen guten Grund, durchdacht zu partitionieren.

Nun ein kleiner Absatz zu den Namenskonventionen von Festplatten. Befindet sich eine Waschmittel äh... ATA/IDE-Festplatte im System, so heißt diese hda, genauer gesagt /dev/hda. Denn für den Anwender ist auch die Platte nicht mehr als ein Eintrag im Unterverzeichnis /dev. Merkt Euch das und laßt es in Fleisch und Blut übergehen. Werden auf /dev/hda nun Partitionen angelegt, so heißen diese in logisch aufsteigender Reihenfolge hda1, hda2, hda3 usw. hda1-4 sind für die ersten 4 primären Partitionen bzw. ersten 3 primären und die erweiterte Partition vorgesehen. hda5 ist dann die erste logische Partition. Befindet sich aber eine richtige Platte äh... eine SCSI-Platte im System, heißt diese /dev/sda. Und logischerweise heißen die Partitionen sda1, sda2, sda3 usw. Dies ist leicht zu merken. Die erste Platte in einem System erkennt man am 'a'. Sollte nun eine zweite ATA/IDE-Platte eingebaut werden, erhält diese den Namen /dev/hdb. Und die Partitionen heißen wieder logischerweise hdb1, hdb2, hdb3 usw. Analoges gilt selbstverständlich für eine zweite SCSI-Platte mit /dev/sdb und sdb1, sdb2, sdb3 usw. Also: Platten werden alphabetisch, Partitionen numerisch hochgezählt. Die Bezeichnungen sind übrigens leider in anderen UNIXen etwas anders. Obiges gilt nur für Linux.

So, das soll es erst mal gewesen sein. Für Schreibfehler bin ich zuständig. Für Hinweise, die die obige Beschreibung erweitern oder verbessern, bin ich durchaus dankbar. Meckerei geht aber nach /dev/null, was übrigens trotz beliebiger Datenmengen keine eigene Partition braucht ;-)

Cheerio  
Peter <[hertrich@int.uni-karlsruhe.de](mailto:hertrich@int.uni-karlsruhe.de)>