

Linux

Dokumentation

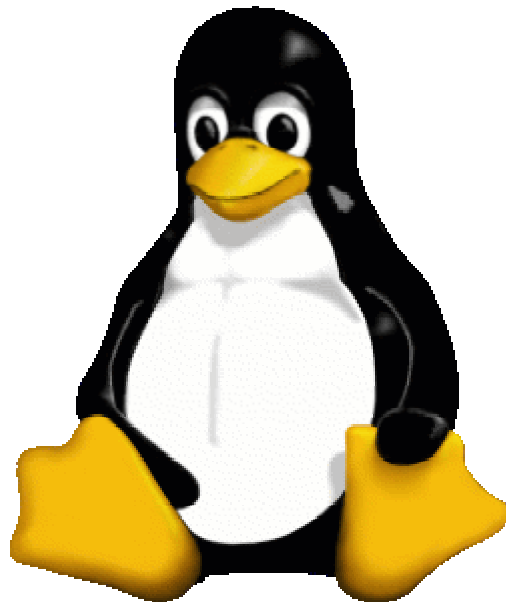
Dateisysteme und Verzeichnisstruktur unter Linux

von

Schubert Matthias, Mattil Peter

Ein Team der FSAut99

Fachlehrer Herr Scheib



Inhaltsverzeichnis

1. Dateisysteme unter Linux

1.1 Die Partition.....
1.2 Bootvorgang / MBR.....
1.3 Dateisysteme allgemein.....
1.4 Linux Dateisysteme.....

2. Verzeichnisstruktur unter Linux

2.1 Dateisystem vs. Verzeichnisbaum.....
2.2 Der Filesystem – Hierarchie – Standard (FHS).....
2.3 Der Linux Verzeichnisbaum.....
2.4 Arbeiten mit Verzeichnissen und Dateien.....

3. Literatur- und Quellenverzeichnis

1. Dateisysteme unter Linux

1.1 Die Partition

Das Dateisystem stellt unter jedem Betriebssystem die Grundlage zur Speicherung von Daten und Dateien dar. Damit ein Betriebssystem auf einem Datenträger wie zum Beispiel einer Festplatte installiert werden kann, muss diese erst einmal dafür vorbereitet werden.

Eine Festplatte kann in verschiedene Partitionen, unterteilt werden. Unter dem Begriff Partitionieren versteht man das Aufteilen der Festplatte in physikalische Abschnitte (reservieren). Partitionen werden in primäre und erweiterte, als auch logische Partitionen unterschieden. Es können bis zu vier primäre Partitionen auf einer Festplatte eingerichtet werden. Alternativ auch drei primäre, eine erweiterte und dann die darin enthaltenen logischen Partitionen. Erweiterte Partitionen dienen nur als Container für die logischen Partitionen. Sinn und Zweck der erweiterten und deren logischen Partitionen besteht darin die Grenze von vier maximalen primären Partitionen zu überwinden. Das Partitionieren stellt die Grundlage zur Nutzung mehrerer Betriebssysteme auf ein und derselben Festplatte dar. Das Aufteilen einer Festplatte bietet neben der Nutzung mehrerer Betriebssysteme die Möglichkeit einer individuellen Einteilung der Festplatte. Es können Partitionen eingerichtet werden, welche z.B. von zwei unterschiedlichen Betriebssystemen genutzt werden. Nach der Aufteilung der Festplatte in eine oder mehrere Partitionen wird durch das Formatieren der entsprechenden Partition das entsprechende Dateisystem erzeugt. Erst nach dieser logischen Formatierung spricht man von einem Datenträger oder Laufwerk. Zu beachten ist, dass durch das Formatieren der ausgewählten Partition alle auf der Partition enthaltenen Daten gelöscht werden, sofern diese zuvor schon ein Dateisystem enthalten haben sollte.

Betriebssysteme benötigen ein entsprechendes Dateisystem. Einige Betriebssysteme können auch auf Partitionen mit unterschiedlichen Dateisystemen zugreifen. Genauere Informationen erhält man bezüglich des unterstützten Dateisystems bei den entsprechenden Herstellern der Betriebssysteme.

1.2 Bootvorgang / MBR

Damit mehr als ein Betriebssystem auf einer Festplatte genutzt werden kann, muss diese in Partitionen aufgeteilt werden. Im Nachfolgenden wird kurz eine mögliche Vorgehensweise beim Booten (laden) eines bzw. mehrerer Betriebssysteme beschrieben.

Die BIOS (Basic Input Output System)-Bootroutine führt das Master-Bootprogramm (bzw. Bootloader) aus, welches Bestandteil des MBR (Master-Boot-Record) ist. Der MBR ist 512Byte groß und befindet sich im ersten Sektor der physikalischen Festplatte. Neben dem Master-Bootprogramm enthält der MBR eine Partitionstabelle in der alle primären Partitionen (max. 4 Partitionen) der Festplatte enthalten sind. Anhand dieser Partitionstabelle erkennt das Master-Bootprogramm welche Primärpartition aktiv ist. Bei nur einer Partition wird das Betriebssystem von dieser Partition gebootet. Enthält die Festplatte mehrere bootfähige Partitionen mit Betriebssystemen, wird über einen Bootdatensatz, welcher im ersten Sektor der jeweiligen (ggf. primären) Partition gespeichert ist das Betriebssystem über ein Bootprogramm gestartet.

Der Bootloader von Linux **LILO (Linux Loader)** kann z.B. im Bootsektor der Linux Partition untergebracht werden und von der im MBR enthaltenen Laderoutine aufgerufen werden.

Betriebssystem	gebootet von :	Primäre Partition	Logische Partition
MS Windows		✓	
Linux		✓	✓

Tabelle 1.2.1

Anhand der *Tabelle 1.2.1* ist erkennbar, dass Linux flexibler bezüglich der Installation auf der Festplatte gegenüber Microsoft - Betriebssystemen ist.

Wenn Linux neben Microsoft Betriebssystemen auf einer Festplatte installiert wird, sollte zuerst das Microsoft - Betriebssystem installiert werden. Da Microsoft – Installationsroutinen meist keine bestehenden, fremde Betriebssysteme berücksichtigen und möglicherweise den MBR überschreiben.

Es ist darauf zu achten, dass noch genug Platz auf der Festplatte für Linux zur Verfügung steht. Da Microsoft - Betriebssysteme sich in der Regel nur über primäre Partitionen booten lassen, sollte man Linux in einer logischen Partition installieren. Somit besteht noch die Möglichkeit mehrere zusätzliche Microsoft – Betriebssysteme auf den noch verfügbaren primären Partitionen zu installieren.

1.3 Dateisysteme allgemein

Das Dateisystem bestimmt die Art der Verwaltung (Organisation) und den Zugriff auf Dateien eines Datenträgers (z.B. Festplatte, CD-ROM usw.).

FAT 16 (File Allocation Table) Dateizuordnungstabelle bzw. VFAT (Virtual File Allocation Table)

Wurde von Microsoft als Grundlage für ihre Betriebssysteme entwickelt.
Kann bis zu 2 Gigabyte (GByte) verwalten.
Wird von DOS, Windows 95/98/Me, Windows NT/2000/XP benutzt.

FAT 32

Wird von Windows 98/Me/2000/XP verwendet und ermöglicht die Verwaltung von Partitionen von bis zu 2 Terabyte (TByte).

Hinweis :

FAT 16 und FAT 32 bzw. VFAT werden umgangssprachlich auch als FAT-Dateisystem bezeichnet.

NTFS (New Technology File System)

Windows NT/2000/XP

HPFS (High Performance File System) Hochleistungsdateisystem

Dateisystem von OS/2, das bedeutend schneller als das FAT- Dateisystem auf die Festplatte zugreift.

HFS (Hierarchical File System)

Dateisystem von Macintosh

Linux kann auf alle o.a. Dateisysteme zugreifen, in welchem Rahmen dies möglich ist zeigt die untenstehende *Tabelle 1.3.1*.

	Linux lesen	Linux schreiben
FAT	✓	✓
NTFS	✓	
HPFS	✓	✓
HFS	✓	✓

Tabelle 1.3.1

Umgekehrt ist der Zugriff auf Linux – Dateisysteme von Microsoft – Betriebssystemen nur mit spezieller Zusatzsoftware, die nachträglich erworben werden muss, möglich.

1.4 Linux Dateisysteme

Mit dem logischen Formatieren einer Partition wird auf dieser das entsprechende Dateisystem erzeugt. Erst nach diesem Formatieren steht die Partition dem Anwender zur Verfügung. In Windows erkennt man dies durch die Vergabe von Laufwerksbuchstaben, wie z.B C, D, F für die Partitionen, auf die nun über diese Laufwerksbuchstaben zugegriffen werden kann. Linux hingegen verschont den Benutzer mit solchen Hardwareangelegenheiten. Der primäre Unterschied zu Windows besteht darin dass Linux auf Laufwerksbuchstaben verzichtet und das Dateisystem der Partition in eine hierarchische Verzeichnisstruktur (Dateisystem) einbindet. Der Vorteil besteht in der Transparenz für den Benutzer.

Als Dateisystem versteht man einerseits die Gesamtheit aller unter Linux verwendeten Dateisysteme, aber andererseits auch das einzelne Dateisystem z.B. das einer Festplatte, CD-ROM oder einer Diskette. Ein Dateisystem stellt eine Verwaltungsstruktur auf der Festplatte dar. Die Gesamtheit der Dateisysteme wird unter Linux durch einen einzigen Verzeichnisbaum dargestellt. Diese Darstellung ist standardisiert siehe Kapitel 2.2. Dadurch entsteht eine hierarchisch klar strukturierte Darstellung des Dateisystems. In diesen Verzeichnisbaum, der von der Systempartition (root, Wurzel) ausgeht, welche ebenfalls eine Partition mit einem Linux – Dateisystem darstellt, werden die entsprechenden Dateisysteme eingebunden. Die Partitionen einer Festplatte müssen nicht das gleiche Dateisystem besitzen, Linux bietet verschiedene Dateisysteme mit entsprechenden Eigenschaften an.

Dateisysteme unter Linux :

- ext2
- ext3
- ReiserFS
- XFS
- JFS

CD-ROM / DVD-Dateisysteme unter Linux :

- iso9660
- udf

Des Weiteren gibt es noch Netzwerkdateisysteme, virtuelle Dateisysteme, Unix-Dateisysteme und die unter Kapitel 1.3 dargestellten Dateisysteme.

Im Nachfolgenden sind einige Merkmale der unter Linux verwendeten Dateisysteme aufgeführt. Es wurde hier besonders Wert auf die Darstellung einfach verständlicher Eigenschaften gelegt.

☑ **ext 2 (extended filesystem, Version 2)**

ext2 ist ein historisch gewachsenes Standard-Dateisystem unter Linux.

Merkmale/Eigenschaften :

- Maximale Partitionsgröße beträgt 2 Terabyte (TByte).
- Lange Dateinamen, maximale Länge 255 Byte.
- Maximale Dateigröße z.B. 16 Gigabyte (GByte), 256 GByte oder 2TByte (abhängig von der Blockgröße).
- Stabiles Dateisystem.
- Zugriff im Notfall von außen, wenn Linux sich nicht mehr starten läßt, ist einfacher z.B. über Notfalldiskette als bei anderen Dateisystemen.
- Unterstützt kein Journaling

☑ **ext 3 (extended filesystem, Version 3)**

Nachfolger des oben beschriebenen ext2 Dateisystems.

Merkmale/Eigenschaften :

- Kompatibel zu dem Dateisystem ext2
- Umwandlung von ext2 in ext3 möglich
- Dateisystem relativ neu und noch nicht so ausgereift wie ext2
- Unterstützt Journaling

☑ **ReiserFS (Namensgebung vom Initiator Hans Reiser)**

Merkmale/Eigenschaften :

- Noch nicht so ausgereift wie ext2, aber länger im Einsatz wie ext3
- Höhere Geschwindigkeit bei kleinen Dateien im Vergleich zu ext2
- Inkompatibel zu ext2
- Unterstützt Journaling

☑ **XFS**

Dateisystem von SGI für IRIX, ist aber mittlerweile für Linux verfügbar.

Merkmale/Eigenschaften :

- Ist ein 64-Bit Dateisystem
- Verwaltung großer Dateien im Terabyte – Bereich (9 Millionen Terabyte)
- Maximale Dateisystemgröße liegt bei 18 Millionen Terabyte
- Unterstützt Journaling

☑ JFS (Journaled File System)

Ist ein von IBM entwickeltes Journaling Dateisystem, ursprünglich für UNIX ist aber mittlerweile auch unter Linux verfügbar.

Wird hauptsächlich als Dateisystem in der Großrechnerwelt eingesetzt.

Merkmale/Eigenschaften :

- Schnelles Dateisystem
- Maximale Dateigröße bis 4 Petabytes (Pbytes)
- Unterstützt Journaling

Wie sich aus den Merkmalen der unterschiedlichen Dateisysteme erkennen läßt, wird Journaling von allen oben genannten Linux – Dateisystemen außer von ext2 unterstützt. Das Dateisystem ext2 protokolliert die Dateizugriffe nicht. Durch das fehlende Protokollieren der Dateizugriffe muss nach einem Systemabsturz, beim Neustart des Rechners das gesamte Filesystem (Dateisystem) der Festplatte überprüft werden. Es werden hierbei alle Dateien auf Inkonsistenz geprüft. Bei großen Festplatten kann dies unter Umständen Stunden dauern. Anders sieht das bei Dateisystemen mit Journaling – Funktion aus. In einem Journal (bzw. Protokoll) wird immer zu jeder Zeit festgehalten, welche Dateien im Moment bearbeitet und auf die Festplatte geschrieben werden sollen. Nach einem Systemabsturz wird bei einem Neustart dieses Journal abgearbeitet und anhand der Einträge wird festgestellt, welche Dateien des Dateisystems inkonsistent sein könnten. Lediglich die betroffenen Dateien müssen überprüft und möglicherweise korrigiert werden. Man kann sich vorstellen dass bei einem gezielten Prüfen der betroffenen Dateien der Dateisystem – Check bedeutend schneller vorangeht, als wenn das gesamte Dateisystem überprüft werden muss. Der Neustart nach einem Systemabsturz wird durch das Journaling im Gegensatz zu dem Vorgehen wie bei ext2 und dem FAT – Dateisystem bedeutend schneller durchgeführt. Die Journaling – Funktion ist effizienter.

2. Verzeichnisstruktur unter Linux

2.1 Dateisystem vs. Verzeichnisbaum

Zur Abgrenzung zwischen dem physikalischen Dateisystem und dem hierarchischen Verzeichnis- und Dateisystem unter Linux sei noch einmal folgende Veranschaulichung erlaubt:

Stellen wir uns unsere Festplatte einmal als ein Bücherregal vor, genauer gesagt ein Regal ohne Fachböden, also im Prinzip ein Kasten. Dieser Kasten ist ein Stück Hardware. Nun bringen wir die Regalbretter an. Diese repräsentieren das Dateisystem, z.B. ReiserFS, das zur Organisation der Daten auf den Partitionen dient.

Stellen wir nun Ordner (Verzeichnisse) und Mappen (Dateien) in unser Regal so haben wir analog das ausgeführt, was Linux bei der Installation tut, nämlich ein bestimmtes Ordnungs- bzw. Verzeichnissystem auf die Festplatte zu speichern. Diesen Verzeichnisbaum wollen wir uns nun im Folgenden ansehen.

2.2 Der Filesystem-Hierarchie-Standard (FHS)

Linux speichert Dateien im Gegensatz zu MS-DOS und Windows in einem großen Verzeichnisbaum, der mit dem Wurzelverzeichnis / (root) beginnt und sich schnell sehr breit verzweigt.

Ein Betriebssystem, das für Desktopsysteme ebenso verwendet werden kann wie für komplexe Netzwerkserver, benötigt einen gewissen Standard für die Hierarchie und Struktur des Verzeichnisbaums.

Hier kommt nun der Filesystem-Hierarchie-Standard ins Spiel.

Der File-System-Standard ist das Projekt einer Gruppe von Linux-Systemadministratoren, -Entwicklern und -Benutzern, die ihre Erfahrungen und Vorstellungen ausführlich im Internet ausgetauscht und diskutiert haben. Das Ergebnis ist von Daniel Quinlan im File-System-Standard zusammengefasst worden, dessen Version 1.2 im März 1995 fertig gestellt wurde. Die Arbeit an der Standardisierung des Dateisystems geht weiter und man versucht das Layout der Verzeichnisstruktur immer mehr einheitlich anzunähern.

Mehrere zentrale Prinzipien ziehen sich durch den Standard:

Um der zunehmenden Bedeutung von Linux als verteiltes Betriebssystem auf lokalen TCP/IP-Netzwerken Rechnung zu tragen, muss ein maximaler Anteil des Dateisystems für die gemeinsame Nutzung (über NFS) vorbereitet sein. Eine Voraussetzung dafür ist, dass dieser Teil des Dateisystems beim normalen Betrieb nicht verändert wird.

Dieses Prinzip führt zu einer Unterscheidung der Daten in zwei Richtungen:

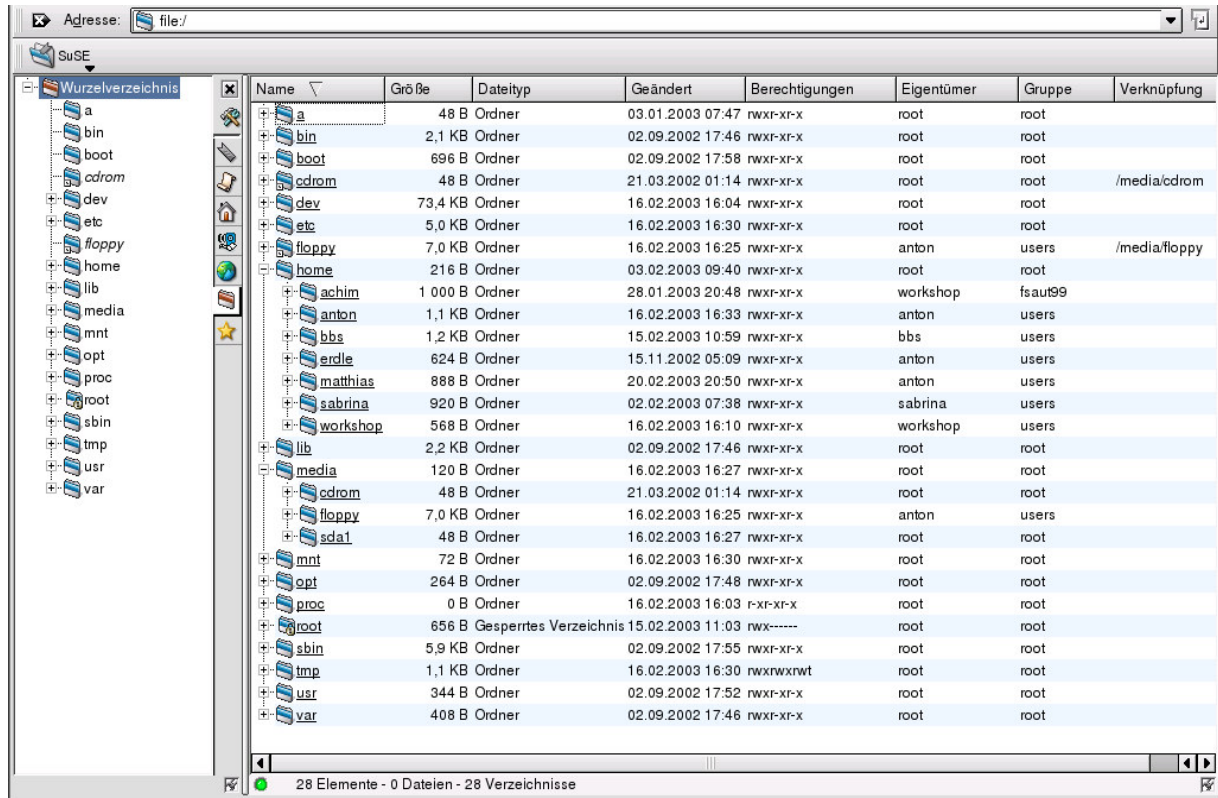
- verteilte Daten / lokale Daten
- variable Daten / statische Daten

Die Unterteilung des Dateisystems sollte nach den bei Unix üblichen funktionalen Kategorien stattfinden. Das bedeutet, dass jeweils Binärdateien, Konfigurationsdateien, Hilfstexte und sonstige Daten in verschiedenen Verzeichnissen untergebracht werden.

Die beiden Prinzipien oben führen insbesondere zur Forderung nach einer klaren und übersichtlichen Strukturierung der Konfigurationsdateien. In dieser Struktur muss sich auch die Unterscheidung zwischen netzweit verteilten und (rechner-) lokalen Daten sowie die Unterscheidung zwischen der von einer Distribution eingespielten und der zusätzlich (netz-) lokal hinzugefügten Software widerspiegeln.

2.3 Der Linux Verzeichnisbaum

Bei der Installation von SuSE Linux werden unter dem Wurzelverzeichnis / (root) mehrere Verzeichnisse und ca. 100.000 Dateien angelegt. Eine grafische Darstellung sehen wir in der folgenden Übersicht des Konquerors.



Im Folgenden möchten wir die einzelnen Verzeichnisse und deren Aufgabe etwas näher beleuchten:

/ (root)

Das Wurzelverzeichnis steht ganz oben in der Hierarchie.

/bin

Hier befinden sich wichtige Programme für Anwender, die immer verfügbar sein müssen, z. B. die Shells. Sie werden für die Aufgaben der Systemverwaltung gebraucht, können aber auch von den anderen Systembenutzern verwendet werden.

/boot

Hier befinden sich der LILO Bootloader mit den zum Hochfahren des Systems unbedingt erforderlichen Dateien. In der Hauptsache ist das der Kernel, die Datei mit dem Namen vmlinuz .

/dev

Dieses Verzeichnis enthält nur Spezialdateien, so genannte Gerätedateien. Jedes Gerät, z.B. eine Festplatte, ein Drucker oder Scanner, ein Laufwerk, usw. wird hier durch eine Datei repräsentiert. Diese stellen eine einfach zu nutzende Schnittstelle zur Hardware dar. Beispiel Festplatten und ihre Partitionen: /dev/hda ist die erste EIDE-, /dev/sda die erste SCSI-Festplatte im System. Höhere Buchstaben (hdb, hdc) stellen weitere Festplatten dar, Zahlen am Ende (sda1, sda2) sind die Partitionen der Festplatten.

/etc

Hier sind viele der Konfigurationsdateien untergebracht, die die Einstellungen verschiedener Programme, Hardware, Dienste oder auch grundlegende Systeminformationen enthalten. Beispiel /etc/services:

Hier können die einzelnen Ports durch das Setzen eines # vor die entsprechende Zeile gesperrt werden.

Überhaupt wird damit ein weiterer Vorteil des Linux – Systems sichtbar. Alle Konfigurations- dateien sind einfache Textdateien, die mit jedem ASCII – Editor bearbeitet werden können.

Somit kann jeder, z.B. über den Befehl cat die Dateien lesen. Ob er sie auch ändern darf hängt von seinen User – Rechten ab.

/home

Dieses Verzeichnis ist das Heimatverzeichnis der Benutzer des Systems. Hier hat der Benutzer alle Rechte und kann nach belieben weitere Verzeichnisse und Dateien anlegen.

Die Existenz solcher individuellen Bereiche ist für ein Mehrbenutzersystem eine selbstverständliche Notwendigkeit, denn die Trennung von System- und Benutzerdaten erhöht die Betriebssicherheit und Stabilität eines Systems.

/lib

Hier befinden sich die wichtigsten Funktionsbibliotheken des Systems. Eigentlich gibt es nur eine Grundregel im Umgang mit diesem Verzeichnis: Finger weg!

/lost+found

Das Filecheck – Programm legt hier Daten ab, die einem defekten Dateisystem nicht mehr zugeordnet werden können.

/media

Dieses Verzeichnis enthält Links zu den angeschlossenen Geräten und Laufwerken.

/mnt

Das Verzeichnis ist leer und steht standardmäßig als mountpoint zur Verfügung.

/proc

/proc ist eigentlich kein normales Verzeichnis, sondern stellt eine Schnittstelle zum Kernel dar. Jedes laufende Programm wird hier in einem Unterverzeichnis geführt, dessen Dateien viele Informationen z.B. über den aktuellen Programmstatus enthalten. Über den Befehl *ps* können diese laufenden Prozesse aufgelistet werden.

/root

Dies ist das Heimatverzeichnis des Systemverwalters root. Es liegt traditionell im Wurzelverzeichnis, damit root auch dann auf seine Dateien (beispielsweise Diagnoseprogramme) zugreifen kann, wenn durch einen Fehler der Zugriff auf andere Partitionen nicht mehr möglich ist.

/sbin

Ähnlich wie /bin enthält auch /sbin wichtige Programme. Diese sind jedoch hauptsächlich für den Systemverwalter gedacht, da sie Funktionen erfüllen, auf die ein normaler Benutzer keinen Zugriff hat, d.h. sie können nur mit root – Rechten gestartet werden.

/tmp

Dieses Verzeichnis kann von jedem Benutzer und jedem Programm als temporäre Ablage für Dateien verwendet werden. Der Inhalt dieses Verzeichnisses wird beim nächsten Booten gelöscht.

/usr

Die umfangreichste Verzeichnisstruktur des Systems. Hier liegt der größte Teil der installierten Software. Die Programmdateien sind meist in /usr/bin, die Spiele in /usr/games. In Netzwerken, an die viele gleichartige Systeme angeschlossen sind, wird dieses Verzeichnis häufig auf einem zentralen Server gespeichert, und alle anderen Computer greifen über das Netzwerk darauf zu. Dadurch müssen die einzelnen Programme auf den einzelnen Stationen nicht jedes Mal vollständig installiert werden. Auch der GNU – C – Compiler ist hier zu finden.

/var

Unter diesem Verzeichnis werden hauptsächlich Dateien gespeichert, die sich ständig verändern. Der Name /var steht für variabel, also veränderlich. Hier befinden sich beispielsweise das Spool – Verzeichnis (/var/spool), Protokolldateien (/var/log), Systemanmeldungsdateien (/var/adm) und vieles mehr.

/opt

(optionale Software) Kommerzielle Software oder sehr große Programme, die nicht unmittelbar zum System gehören, wie etwa *KDE*, *Netscape*, *Mozilla* usw. finden hier ihren Platz.

2.4 Arbeiten mit Verzeichnissen und Dateien

Schauen wir uns im Folgenden an wie man unter Linux auf der CLI (vgl. Kommandozeile) mit Verzeichnissen und Dateien arbeitet.

Die wichtigsten Befehle lauten:

`pwd`

Steht für „print working directory“ und gibt das Verzeichnis aus, in dem wir uns befinden.

`cd Verzeichnisname`

„Change directory“; wechselt in das angegebene Zielverzeichnis.

`cd ..`

Wechselt in das nächst übergeordnete Verzeichnis.

`ls`

Bedeutet „list“ und listet den Inhalt des aktuellen Verzeichnisses auf. Der Befehl kennt eine ganze Reihe von Optionen, z.B. `ls -l`. Damit wird der Verzeichnisisinhalt mit allen Attributen angezeigt.

`mkdir Verzeichnisname`

„Make directory“; legt ein neues Verzeichnis mit dem angegebenen Namen an.

`rmdir Verzeichnisname`

„Remove directory“; entfernt das angegebene Verzeichnis, vorausgesetzt es ist leer.

`rmdir -r Verzeichnisname`

Löscht das Verzeichnis mit allen darin enthaltenen Verzeichnissen und Dateien.

`cat Dateiname`

Bedeutet „catalog“ und zeigt den Inhalt der genannten Datei an

`touch Dateiname`

Mit dem Befehl `touch` kann man schnell eine leere Datei anlegen, z.B. für Testzwecke.

`rm Dateiname`

Löscht die angegebene Datei ohne Nachfrage.

`cp Quelldatei Zielfdatei`

Die Datei wird kopiert, wobei die Zielfdatei ein Dateiname oder ein Verzeichnis sein kann.

`mv Dateiname1 Dateiname2`

Bedeutet „move“; ist *Dateiname2* ein Verzeichnis, dann wird die Datei *Dateiname1* in dieses Verzeichnis verschoben. Falls *Dateiname2* eine Datei ist, wird *Dateiname1* in *Dateiname 2* umbenannt.

`ln -s Dateiname1 Dateiname2`

ln steht für „link“ und -s für „symbolic“; mit diesem Befehl wird ein symbolischer Link von *Dateiname1* erzeugt, der partitions- und datenträgerübergreifend funktioniert. Unter Windows nennt man solche Gebilde Verknüpfungen. Ohne das Attribut -s entsteht ein sogenannter Hardlink der nur auf der aktuellen Partition liegen darf.

Diese Liste erhebt bei weitem keinen Anspruch auf Vollständigkeit. Mehr Informationen zu diesem Thema erhalten Sie in der einschlägigen Literatur und im Internet (siehe Literatur- und Quellenverzeichnis)

3. Literatur- und Quellenverzeichnis

SuSE Linux 8.0 Systemverwalter

Herdt – Verlag

Linux Installation, Konfiguration, Anwendung
Kernel 2.4 SuSE 8.1 6.Auflage

ADDISON-WESLEY

www.selflinux.de

www.linux-ag.de