

Ausarbeitung für den Schwerpunkt

"Linux"

Vorwort

Im Rahmen unserer Projektarbeit haben wir uns unter anderem für die Ausarbeitung von Schwerpunkten entschieden, die bei der Umfrage eine hohe Resonanz hatten. Eine unserer Frage lautete „Welche Betriebssysteme sollten schulischer Lerninhalt sein?“. Wie erwartet standen die Betriebssysteme Windows und NT von Microsoft an erster Stelle der Rangliste, gefolgt von Unix und Linux. Da Kenntnisse in Windows von einem Datenverarbeitungs-Techniker erwartet werden können und NT schulischer Lerninhalt ist, haben wir uns für Linux als separates Themengebiet entschieden.

Linux erfreut sich seit Jahren zunehmender Beliebtheit und wird mit der Unterstützung von großen Firmen wie SUN, HP, Intel, Oracle, SAP, AOL und Corel voran getrieben, indem sie ihre Software kostenlos nach Linux exportieren, Treiber für Geräte zu Verfügung stellen oder die großen Distributoren finanziell unterstützen. Distributoren sind Firmen die mit dem Kernel (Linux Betriebssystem), Software und Installationshilfen mit Handbüchern zur Verfügung stellen bzw. verkaufen. SuSE ist z.B. in Deutschland Marktführer und RedHat in den USA, um nur einige von vielen Distributoren zu nennen, die es gibt.

Die Zahl der Linux Anwender stieg seit der Entstehung 1991 unaufhaltsam, von einigen auf über 10 Millionen Anwender bis heute. Sollte die rasante Zunahme der Anwender so weiter gehen, wird in wenigen Jahren auf fast jedem Computer Linux installiert sein.

In dem nachfolgenden Ausarbeitung wird die Entstehung, Philosophie und die Eigenschaften bzw. Stärken von Linux erläutert ohne in die Tiefe der Materie einzugehen. Es soll für jeden Anfänger leicht verständlich sein.

Inhaltsverzeichnis

- Vorwort
- 1. Linux History
- 2. Linux Philosophie
 - 2.1 Linux Logo
- 3. Was ist Linux ?
- 4. Aufbau des Kernels
 - 4.1 Module
 - 4.1.1 Der Daemoenprocess kerneld
 - 4.2 Speicherverwaltung
 - 4.3 Zugriff auf die Peripheriegeräte (Gerätetreiber)
 - 4.3.1 Treibertypen
 - 4.3.2 Zeichenorientierte Treiber
 - 4.3.3 Blockorientierte Treiber
 - 4.3.4 Pseudogerätetreiber
 - 4.4 Programm- und Prozessverwaltung
 - 4.5 Das Netzwerk
- 5. Das Dateisystem
 - 5.1 Die wichtigste Merkmale des Linux - Filesystems ext2fs
 - 5.2 Andere Dateisysteme
 - 5.2.1 Von Linux unterstützte Dateisysteme
 - 5.3 Virtuelles Dateisystem
 - 5.4 Zugriffsrechte verwalten
 - 5.5 Verzeichnisstruktur
- 6. Der Kommandozeilen-Interpreter (Shell)
- 7. Das X-Window
 - 7.1 Window Mananger
- 8. Zum Schluss noch einige Fähigkeiten von Linux

- 8.1 Personal Unix Workstations
- 8.2 X Terminal Clients
- 8.3 X Application Server
- 8.4 Unix Entwickler Plattform
- 8.5 Kommerzielle Entwicklungen
- 8.6 Network Server
- 8.7 Internet Server
- 8.8 Terminal Server, Fax Server, Modem Server
- 9. Literaturverzeichnis

1. Linux History

1969 Am Anfang von Linux steht Multics, ein von Bell Laboratories geschaffenes "Betriebssystem", das noch mit Lochkarten arbeitet. Unix ist eine Weiterentwicklung unter Mitarbeit von Ken Thompson und Dennis Ritchie. Unix, der Name kommt aus **Uni** von University und **X** von MULTICS (sprich Multix) zustande.

1984 Durch Richard Stallman wird die Free Software Foundation (FSF) gegründet. Diese initiiert ein Projekt namens GNU (**GNU** ist **Not Unix**), welches das Ziel verfolgt, ein freies UNIX System zu schaffen.

1987 AT&T verbietet die Verwendung des Unix Quellcodes an den Universitäten. Während dieser Zeit ist Andrew (Andy) Tanenbaum Professor an der Freien Universität von Amsterdam und entwickelt Minix.

1991 Linus Torvalds hat als Student für Computerwissenschaften an der Universität in Helsinki (in Finnland) Minix installiert und das Buch von Andy Tanenbaum gelesen. Er beginnt im Sinne der FSF einen Unix ähnlichen Kernel zu programmieren mit der Idee, "ein besseres Minix als Minix" zu bauen. 1992 stellt er die Version 0.02 mit dem Namen Linux der Internetgemeinde zur Verfügung.

1994 Die Linux Version 1.0 wird freigegeben. Das „Linux Fieber“ beginnt...

1998 Der Kreis der Entwickler und Anwender ist kräftig angeschwollen. KDE 1.0 (**K-Desktop Environment**, eine Grafische Benutzeroberfläche) wird veröffentlicht. Die Zuwachsraten von Linux übertreffen die von Windows NT deutlich.

1999 Der lange erwartete stabile Linux Kernel Version 2.2 wird freigegeben. Namhafte Firmen wie IBM kündigen ihre Unterstützung für Linux an. Die kanadische Softwareschmiede Corel kündigt eine eigene Linux Distribution an und geht zu diesem Zweck eine Partnerschaft mit Debian und KDE ein.

2. Linux Philosophie

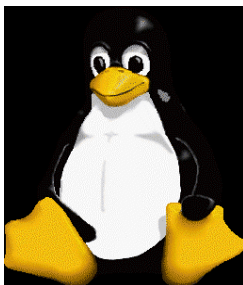
Linux ist im Gegensatz zu den meisten anderen Betriebssystemen kein kommerzielles Produkt, sondern ein von auf der ganzen Welt verstreuten Entwicklern geschaffenes System. Diese Gruppe hat die Ressourcen des Internets für ihr Projekt genutzt um Linux zur Verfügung zu stellen.

Linus Torwalds, der „Vater“ von Linux, hat das Minix - Betriebssystem zu einem voll - leistungsfähigen Unix - Clone ausgebaut. Das Projekt ist noch weit davon entfernt abgeschlossen zu werden. Linux wird ständig aktualisiert und erweitert, und zwar von Hunderten von Entwicklern auf der gesamten Welt.

Während am Anfang nur die GNU (**GNU is Not UNIX**)-Pakete der Free Software Foundation zu haben waren, so werden heute bereits kommerzielle Applikationen und Software, unter der GPL (**General Public License**) verbreitet.

Praktisch alle Bestandteile des Betriebssystemkerns (im folgenden Kernel genannt) und auch die mitgelieferte Zusatzsoftware (sofern nicht kommerziell) unterliegen der GPL. So kann sich jedermann mit den entsprechenden Kenntnissen über Linux und C, den Quellcode besorgen und ihn nach seinem Gutdünken verändern und wieder anderen zur Verfügung stellen.

2.1 Linux Logo



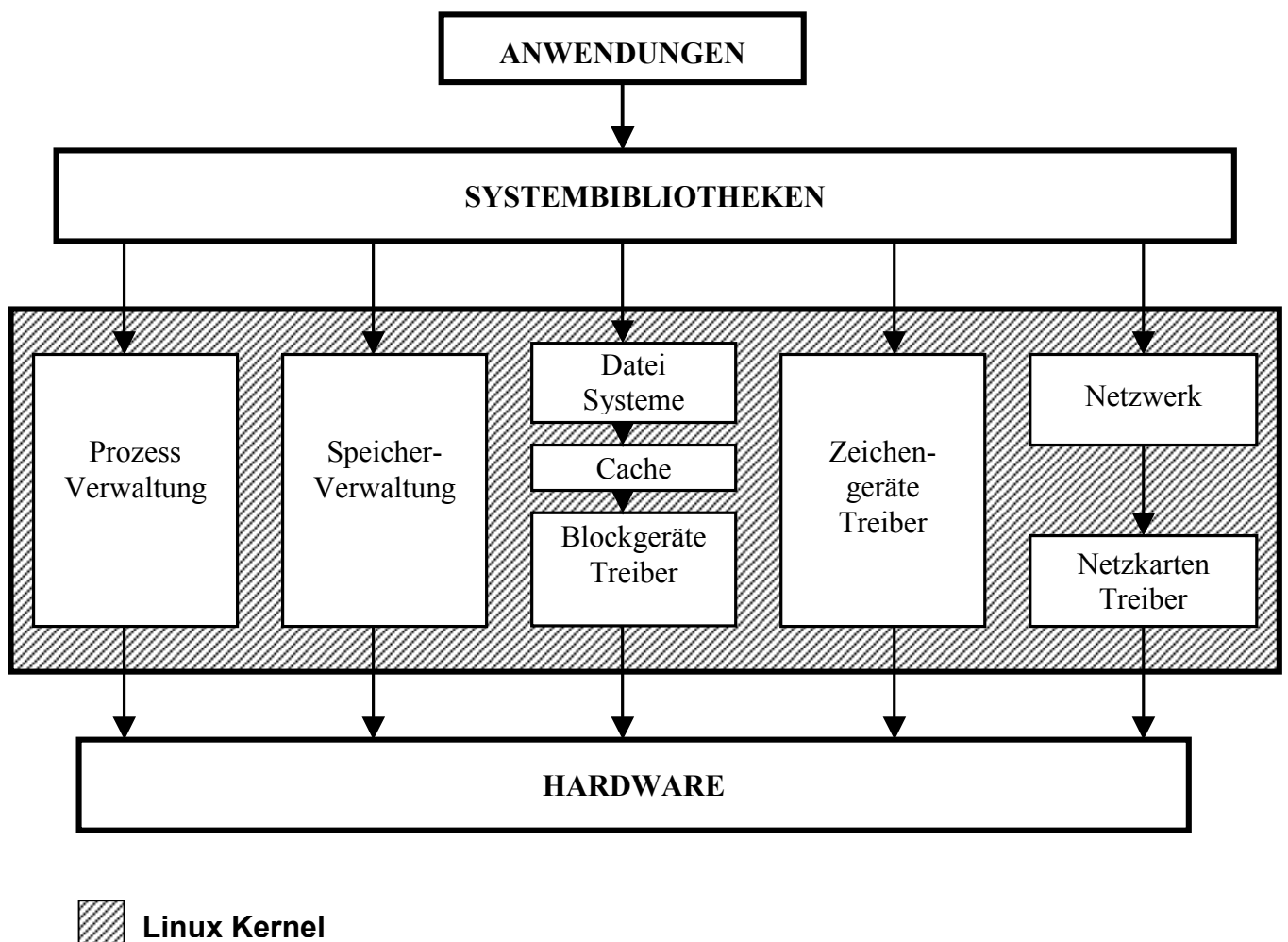
Das Maskottchen der Linux-Gemeinde, der Pinguin Tux, wurde 1996 von Larry Ewing mit dem Grafikprogramm The GIMP (GNU Image Manipulation Programm) illustriert.

3. Was ist LINUX ?

- der Begriff „LINUX“ ist eigentlich nur der Kern eines Betriebssystems, also der Kernel
- ein modernes, frei verfügbares Unix ähnliches OpenSource (Quellcode ist frei verfügbar) Betriebssystem
- ursprünglich für Intel 80386 Prozessoren oder höher entwickelt
- heute ist es für fast alle Plattformen erhältlich
- zuverlässiges Betriebssystem auf Internet-Servern (höhere Stabilität)
- ist je nach Hardware Architektur, ein vollwertiges 32/64Bit Betriebssystem
- deckt den vollen Umfang eines modernen Betriebssystem ab:
 - echtes Multitasking (mehrere Programme und Prozesse gleichzeitig laufen lassen und verwalten)
 - dynamische Laufzeitbibliotheken
 - fortgeschrittenes Speicherverwaltung
 - Virtual Memory (ausgelagerter Arbeitsspeicher auf die Festplatte)
 - Netzwerkunterstützung
 - Multiuser-System (Mehrere Benutzer teilen sich eine Anwendung, auf virtuellen Konsolen.)
 - System Kern optimierbar
 - SMP - fähig (Syncronized Multi Processing, ist für den Mehrprozessorsysteme (Mulitprocessing) eine spezielle Kernel Option.)
 - graphische Oberflächen (KDE, X11)
 - Kompatibilität zu anderen Dateisystemen
 - Programmierbare Shell (Kommandozeilen-Interpreter, wie z.B. DOS)

4. Aufbau des Kernels

Linux ist eigentlich nur der Kern, der für die Hardware zuständig ist. Er koordiniert und verteilt begehrte Ressourcen wie CPU-Zeit und Speicherverteilung, spricht über Treiber mit den verschiedenen Geräten, vom Chipsatz des Boards bis zur Soundkarte und stellt grundlegende Netzwerkprotokolle wie TCP/IP bereit. Durch kompilieren eines eigenen Kernel kann der Linux User dafür sorgen, dass lediglich die benötigten Gerätetreiber, Dateisysteme und Kernel Funktionen eingebunden werden. So entsteht ein Kernel, der optimal an die Hardware angepasst ist.



4.1 Module

Ein weiterer wichtiger Bestandteil der Linux - Architektur sind die Module. Sie ermöglichen eine größere Flexibilität des Kernels. Eigentlich ist Linux ein sogenanntes monolithisches System. Der Kernel ist ein einziges große Programm, in dem alle Komponenten Zugriff auf die internen Datenstrukturen haben. Alternativ könnte man ein Micro-Kernel-System verwenden. Hierbei enthält der Kernel nur die allernotwendigsten Komponenten. Alle anderen Funktionen werden später über zusätzliche Prozesse realisiert. Eine monolithische Struktur macht den Kernel allerdings schwer erweiterbar. So muss nach Einbau neuer Hardware der Kernel umgestellt und neu kompiliert werden.

Linux löst diese Problem heute durch sogenannte Module. Sie stellen eine Mittelweg zwischen monolithischer Struktur und Micro-Kernel dar. Module können ähnlich wie Windows9x- oder DOS-Treiber zur Laufzeit geladen und wieder entfernt werden.

4.1.1 Der Daemoenprocess kerneld

Der Daemoenprocess (Ein Daemon ist ein Systemprozess, der agiert ohne, dass ihn ein Benutzer dazu aufgefordert hat.) kerneld übernimmt das automatische Laden oder Entfernen von Modulen. Dazu informiert ihn der Kernel, wenn ihm ein bestimmtes benötigtes Modul fehlt. Der Daemon stellt das Modul dann zur Verfügung. Dieses Nachladen nach Bedarf wird demand loading genannt. Der Daemoenprocess läuft als ganz normaler Systemprozess mit Rechten eines Administrators. Nach einer bestimmten Zeit entfernt der Daemon das Modul auch wieder. Nur manuell geladene Module werden nicht entfernt.

4.2 Speicherverwaltung

Der physikalische Speicher wird durch Virtualisierung erweitert. Dabei werden wie in Windows Programme und Programmteile, die aktuell nicht benötigt werden, in einen Auslagerungsbereich der Harddisk (Swap-Partition in Linux, Auslagerungsdatei in Windows9x).

4.3 Zugriff auf die Peripheriegeräte (Gerätetreiber)

Gerätedateien stellen die Schnittstelle zwischen Gerätetreibern und Hardware da. Über die Gerätedateien können wiederum Anwendungen mit ihnen kommunizieren. Sie sind Teil des Kernels oder können als Modul geladen werden. Der Linux-Kernel unterstützt eine Vielzahl von Gerätetreibern für Festplatten, Streamer, CD-ROMs usw.

4.3.1 Treibertypen

Linux unterscheidet 3 verschiedene Kategorien von Treibern.

- Zeichenorientierte Treiber
- Blockorientierte Treiber
- Pseudo-Geräte-Treiber

4.3.2 Zeichenorientierte Treiber

Sie werden von Geräten benutzt, die einen Zeichenweise Zugriff aufweisen. Der Kernel puffert die Ausgabe nicht, wie z.B. bei Tastatur, Maus, serielle Leitungen, Modems und parallel angeschlossenen Drucker.

4.3.3 Blockorientierte Treiber

Einige Geräte erlauben den Zugriff in größeren Datenblöcken. Für diese blockorientierten Geräte existierte ein Buffer Cache zwischen Geräte und Dateisystem. Um an diesen Buffer Cache nicht gebunden zu sein, kann in kleineren Einheiten als die des physikalischen Geräts oder des Buffer Cache zugegriffen werden, wie z.B. bei Festplatten, Disketten, CD-ROM-Laufwerke.

4.3.4 Pseudogetätetreiber

Pseudo- oder Softwaretreiber kontrollieren eigentlich keine Hardware. Sie benutzen aber dieselben Schnittstellen wie die anderen Treibertypen.

4.4 Programm- und Prozessverwaltung

Ein Prozess ist ein in der Ausführung befindliches Programm mit seiner Ausführungsumgebung. Linux trennt einen Prozess streng von dem eigentlichen Programm und der Umgebung, in welcher der Prozess abläuft. Die Prozesse werden hierarchisch durch eindeutige Nummer (Prozessnummer, PID) und die Nummer des Elternprozesses (auch Erzeugerprozess) geordnet. Alle Prozesse werden in einer zentralen Prozesstabelle vom Kernel verwaltet. Damit jeder Prozess das gleiche Recht auf die Rechenzeit der CPU hat, muss dies durch ein Verfahren gewährleistet und kontrolliert werden. Ein Teil des Kernels, **Scheduler** genannt, hat genau diese Aufgabe. Er durchläuft hierzu ständig die Prozesstabelle. Schedule wird immer zu Beginn und zum Ende eines Systemaufrufs „angesprungen“. Die strenge „Regie“ eines Betriebssystems über die einzelnen Prozesse nennt man **preemptives Multitasking**.

Wichtige Einträge in der Prozesstabelle sind:

- Prozessidentifikation (PID)
- Prozessprioritäten (Zeit und Reihenfolge für die jeweilige Prozessorverarbeitung des Prozesses)
- Accounting - Information (virtueller Speicher)
- Kontrollterminals (Einträge 0, 1, und 2. Nur bei Daemons kann kein Terminal eingetragen sein).

4.5 Das Netzwerk

Linux ist vom Hause aus für Netzwerkverbindungen konzipiert.

Über die Netzwerkkarte bzw. Module (Treiber) werden WAN (**Wide Area Network**) Verbindungen durch PPP-Protokolle (**P**oint to **P**oint **P**rotocol) hergestellt. Mittels PPP ist es möglich, ins Internet eine Verbindung herzustellen, wobei das Internet Protokoll TCP/IP benutzt wird. Die Fähigkeiten zum Umgang mit PPP und TCP/IP, bringt der Kernel schon mit sich.

5. Das Dateisystem

Linux benutzt das ext2fs-Dateisystem (**E**xtended **F**ile **S**ystem Version 2), zusätzlich werden andere Filesysteme Unterstützung (MS DOS, VFAT, ISO9660). Nicht zu vergessen sind Kenntnisse über die Besitzverhältnisse und Zugriffsrechte, die zu den wesentlichsten Merkmalen von Linux Dateisystemen zählen.

5.1 Die wichtigsten Merkmale des Linux - Filesystems ext2fs

- Dateinamen können bis zu 255 Zeichen lang sein.
- Groß- und Kleinschreibung werden unterschieden.
- Alle Zeichen sind erlaubt (aber auf Sonderzeichen sollte man besser verzichten). Es ist sogar möglich, einen aus Leerzeichen bestehenden Dateinamen zu bilden.
- Die Dateilänge ist auf 2 GB beschränkt.
- Eine Dateinamenserweiterung wie unter DOS existiert in Linux nicht; der Punkt ist ein ganz normales Zeichen. Dennoch verwenden Archivierungs- und Komprimierungsprogramme wie tar und gzip derartige Suffixe .tar, .gz.

5.2 Andere Dateisysteme

Linux ist ein Multitalent und versteht die Sprachen vieler Dateisysteme. Voraussetzung für den Zugriff auf diese ist natürlich ein entsprechend konfigurierter Kernel. Um alle unterstützten Dateisysteme unter einem Hut zu vereinigen, ist eine einheitliche Schnittstelle erforderlich. Diese wird vom **virtuellen Dateisystem** bereitgestellt.

5.2.1 Von Linux unterstützte Dateisysteme

Ext	Vorgänger des ext2
ext2	Linux
iso9660	CD-ROM
hpfs	OS-2 (nur lesend)
minix	Minix (kaum noch verbreitet)
msdos	DOS
nfs	Network File System
ntfs	WindowsNT
SMB	(Server Message Block) NT, Windows für Workgroups
Swap	Swap-Partitionen oder -Dateien
SystemV	Verschiedene Unixe
Proc	Prozessverwaltung
Umsdos	DOS-Dateisystem unter Linux verwenden
Vfat	Windows95
...	...

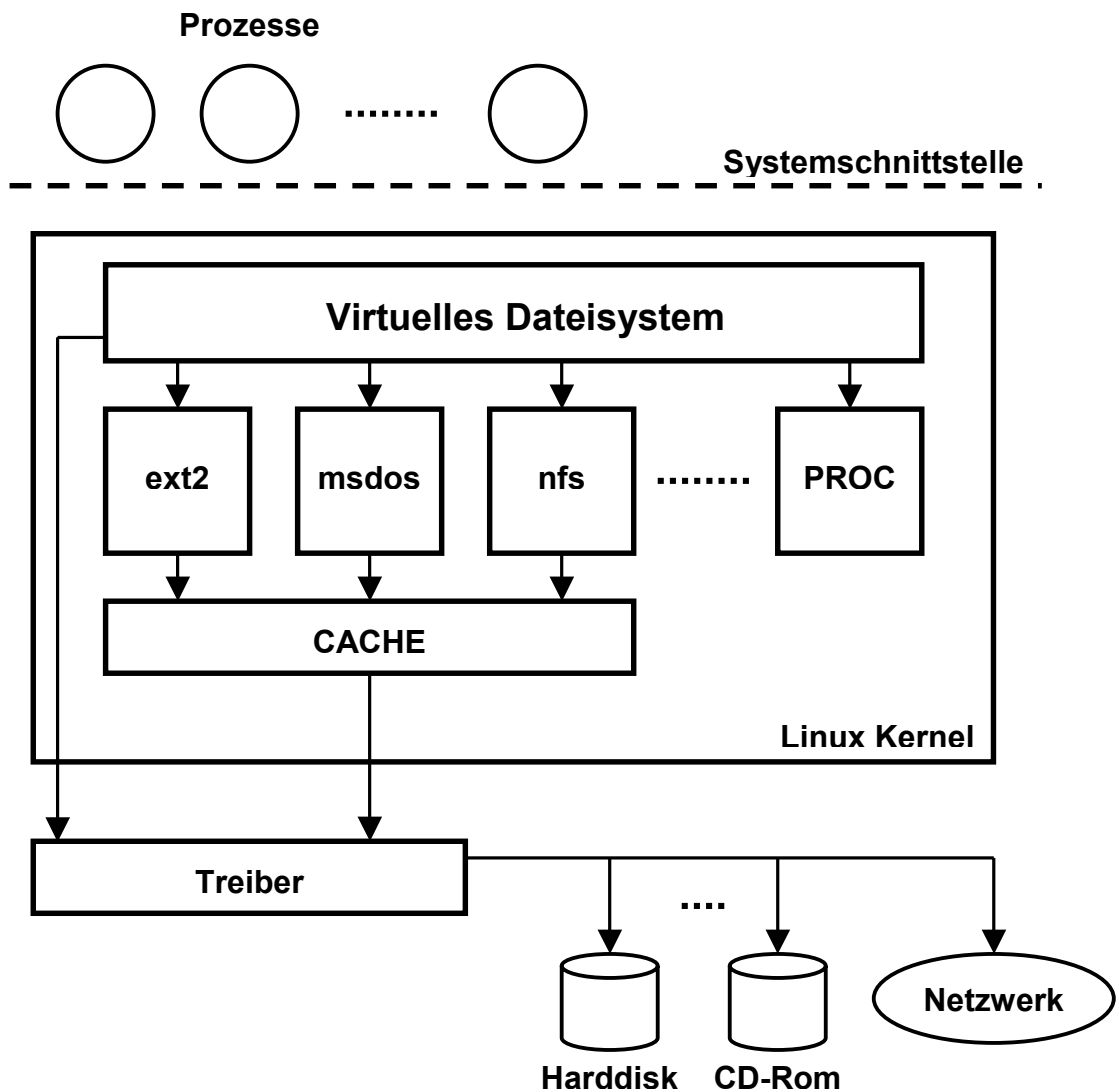
5.3 Virtuelles Dateisystem

Unter Linux gibt es, im Gegensatz zu den Betriebssystemen von Micro\$oft, keine einzelnen Laufwerke. Alle Laufwerke werden als frei zu wählende Unterverzeichnisse des Wurzelverzeichnisses ROOT angezeigt.

Für einen Prozess im System spielt es keine Rolle, auf welchem Medium sich die zu bearbeitende Datei befindet, ob im Linux - Dateisystem (ext2), ob auf einer MS-DOS - formatierten Diskette oder auf einem anderen Rechner irgendwo im Netzwerk...

Der Prozess weiß nichts über Details der Implementierung; er kennt nur die Schnittstelle des Virtuellen File Systems (VFS) und greift über dessen Funktionen auf die Dateien zu.

Das Virtuelle Dateisystem vereint alle unterstützten Dateisysteme und setzt deren interne Strukturen in eine klar definierte Struktur um, die letztlich den Prozessen zur Arbeit mit den Daten dient. Die Abbildung veranschaulicht dies



5.4 Zugriffsrechte verwalten

Linux ist ein Multiuser- und Mehrprozess System, mehrere Benutzer können zur gleichen Zeit mehrere Prozesse bzw. Programme ausführen. Alle Programme, die ein Benutzer startet, laufen unter seiner Benutzerkennung. Alle Dateien, die von einem User angelegt werden, unter dessen Benutzerkennung gespeichert. Benutzerkennung besteht aus dem Benutzernamen, mit dem sich der User anmeldet und einer User-ID, mit der das Betriebssystem arbeitet.

Rechte auf einem Mehrbenutzersystem heisst, dass immer auch ein Rechtekonzept für Prozesse, Dateien und Verzeichnisse existieren, muss damit User A nicht auf die Dateien von User B zugreifen kann. Linux implementiert dazu das Standardkonzept aller Unix Betriebssysteme, jede Datei / Verzeichnis ist genau einem User und einer Gruppe zugeordnet. Jeder User kann Mitglied mehrerer Gruppen sein. Jede Datei kennt Berechtigungen für den **User**, die **Gruppe** und allen **Anderen**. Für diese drei Gruppen können jeweils Schreib-, Lese- und Ausführungsrechte vergeben werden. Anhand dieser Berechtigungen ist somit festgelegt, wer eine Datei lesen, schreiben oder ausführen darf.

5.5 Verzeichnisstruktur

Das Verzeichnisstruktur von Linux ist organisiert nach dem sogenannten FHS Standard (engl. **F**ilesystem – **H**ierarchy - **S**tandard).

Folgende Verzeichnisse gibt es auf jedem Linux-System, unabhängig davon, welche Distribution verwendet wird:

/bin	Wichtige Dienstprogramme für alle Anwender
/boot	Enthält bei den meisten Distributionen den Kernel sowie wichtige Informationen für den/die Ladeprogramme.
/dev	Verzeichnis mit allen Gerätedateien, über die die Hardware angesprochen wird.
/etc	Ordner mit Dateien für die Konfiguration des Systems
/home	Ordner mit den Heimatverzeichnissen der Benutzer
/lib	Verzeichnis für gemeinsam genutzte Bibliotheken (sog. Shared Libraries)
/opt	Zusatzprogramme, die kein fester Bestandteil von Linux sind, werden in diesem Ordner gespeichert
/mnt	Verzeichnis für temporär einzuhängende Dateisysteme (z.B. Diskettenlaufwerk, CD)
/proc	Verzeichnis mit Informationen über laufende Prozesse und die Hardware (sog. Pseudo-Dateisystem)
/root	Heimatverzeichnis des Administrators
/sbin	Systemprogramme, die nur vom Administrator gebraucht werden (sog. System Binaries)
/tmp	Verzeichnis für temporäre Dateien
/usr/X11R6	Stammverzeichnis für alle Daten des X-Window-Systems (grafische Oberfläche)
/usr/bin	Das Verzeichnis enthält weitere Programme
/usr/doc	Dokumentation von Linux
/usr/include	Header-Dateien für C/C++
/usr/lib	Bibliotheken

/usr/local	Dateien und Programme, die nur auf dem aktuellen Rechner benötigt werden
/usr/man	Das Online-Handbuch
/usr/sbin	Das Verzeichnis enthält weitere Systemprogramme
/usr/src	Der Quelltext für den Betriebssystemkern und eventuell weiteren Programmpaketen
/var	Spezielles Verzeichnis, in dem Linux variable, sich schnell oder häufig ändernde Daten unterbringt (z.B. Drucker-Warteschlange)

6. Der Kommandozeilen-Interpreter (Shell)

Die „bash“ ist standardmäßig das erste Programm, das von Linux nach dem erfolgreichen Einloggen gestartet wird. Die Shell allgemein ermöglicht eine bequeme Eingabe und Ausführung von Kommandos - daher auch die Bezeichnung eines Kommandozeileninterpreters. Es ist eine Schnittstelle zwischen dem Anwender und dem Betriebssystem. Selbst bei grafischer Bedieneroberfläche wird man an einem Shell-Terminal nicht vorbei kommen. Zusätzlich stellen die Shells unter Linux eine Programmiersprache zur Verfügung, um so genannte Shell-Skripte erstellen zu können. Shell-Programme bzw. Shell-Skripts sind im wesentlichen Kommandofolgen, zusammengefasst in einer Datei. Ihre zentrale Bedeutung für Linux kommt unter anderem dadurch zum Ausdruck, dass ein großer Teil der Systeminitialisierung durch Abarbeiten der Shell-Skripts erreicht wird. Auf der nachfolgender Seite sind einige Standart Kommandos von bash-Shell, zur Dateiverwaltung aufgeführt:

Dateien / Verzeichnis listen (**Quelle**)

ls [-parameter] [-l Muster] [Verzeichnis] [Datei]

Parameter	Erläuterung
-a	all, listet auch Dateien auf, die mit „.“ beginnen
-x	Ausgabe in horizontal sortierten Spalten
-i	zeigt zusätzlich die Inodenummer zu jeder Datei
-R	recursive, alle Unterverzeichnisse rekursive
-l Muster	ignoriert alle Dateien, die dem Muster entsprechen
-l	long, zeigt auch Recht, Typ, Hardlinks, Besitzer, Groups, Größe in Byte und Veränderungsdaten

Verzeichnis erstellen

mkdir [-p] [-m Modus] Verzeichnis

Parameter	Erläuterung
-m Modus	setzt die Rechte von Verzeichnis auf Modus, der von chmod angegeben wird
-p	parents, legt nicht existierende Elternverzeichnis mit an

Verzeichnis löschen

rmdir [-p] [-m Modus] Verzeichnis

Parameter	Erläuterung
-p	Verzeichnis rekursiv löschen

Dateien kopieren

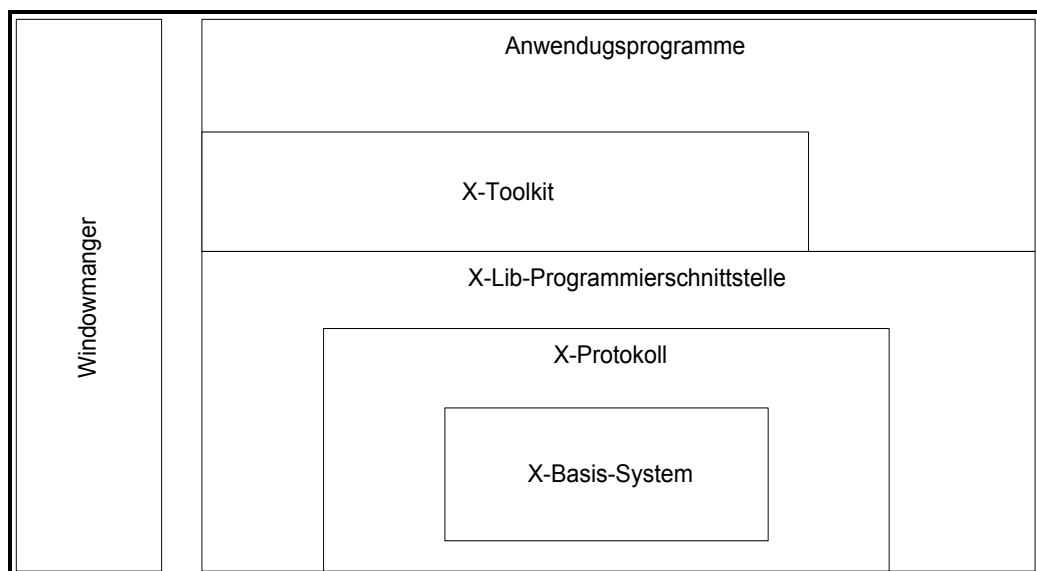
cp [-parameter] [-V {numbered,existing,simple}] Quelle Ziel

Parameter	Erläuterung
-b	backup, sichert die Dateien vor dem überschreiben
-P	path, kopiert die Dateien mit dem Verzeichnis relativ zum Ziel
-r	kopiert die Dateien mit dem zugehörigen Verzeichnis mit
-v	verbose, ausführliche Meldungen
-R	recursive, rekursives Kopieren (alle Unterverzeichnisse mitkopieren)
-V	version-controll, frühere Dateiversionen werden in einem Backup gehalten

7. Das X-Window

Das X-Window-System wurde von DEC (Digital Equipment Corporation) und dem MIT (Massachusetts Institute of Technology) entwickelt und stellt heute einen Standard für grafische Bedieneroberflächen für Unix dar. Die offizielle Bezeichnung bei Linux/Unix Systemen ist **X11-Window**. Die Arbeitsweise von X-Window folgt einer sogenannten Client/Server-Architektur in der Weise, dass der **X-Server** die lokalen (- also der eigene Rechner -) Ein-/Ausgabekomponenten steuert und damit für einen oder mehrere X-Anwendungen (**X-Clients**) eine Ein-/Ausgabeschnittstelle unterhält. X-Clients können auf dem eigenen Rechner oder aber auch auf jedem anderen Rechner im Netz laufen - selbst unter einem anderem Betriebssystem. Informationsaustausch findet zwischen dem X-Server und dem X-Client über das **X-Protokoll** statt.

Das untenstehende Schichtenmodell zeigt den Aufbau eines **X-Window Systems**.



X-Window-Layermodell

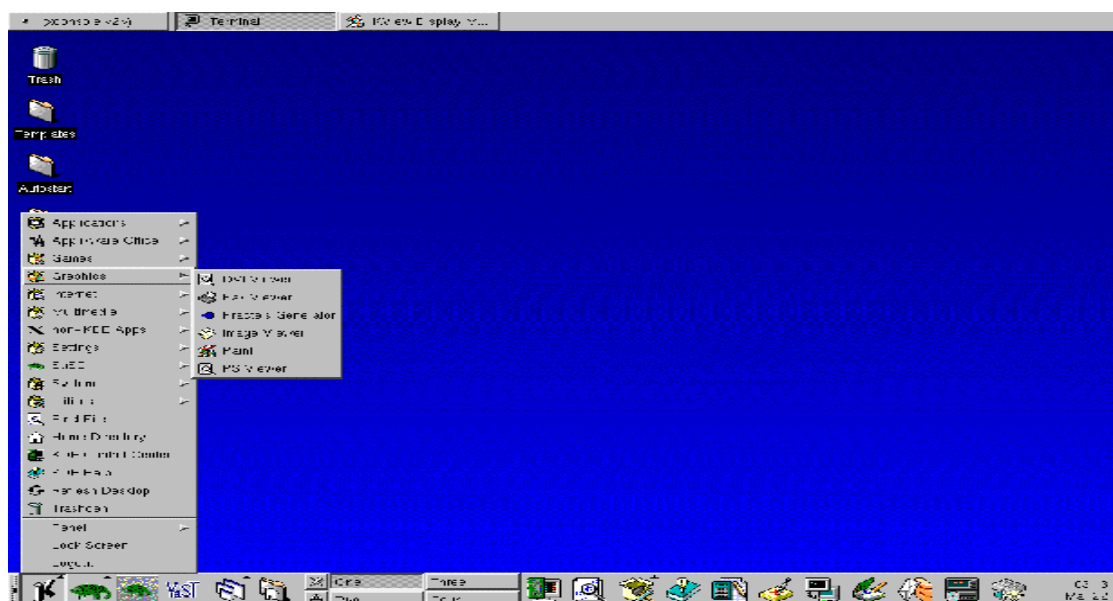
7.1 Window Manager

Aussehen und Handhabung der Fenster unter X-Window bestimmt ein Window Manager, der selbst nur ein X-Client ist und theoretisch sogar auf einem anderen Rechner laufen kann. Window Manager legen die Funktionsweise und Optik von Fensterrahmen, Buttons, Icons und Menus fest. Der X-Server sorgt lediglich für die Darstellung von grafischen Elementen bzw. die Übermittlung von Ereignissen.

Unter Linux stehen dem Anwender eine ganze Reihe unterschiedlichster Window Manager für X-Window zur Auswahl. Einer der führenden Window Manager ist der KDE (**K-Desktop Environment**).

Damit wird Linux auch für den normalen Anwender mit Windows, in der Bedienung vergleichbar einfach.

Das KDE bringt alle Features einer modernen Oberfläche mit sich. Dies umfasst Funktionen wie moderne Desktopgestaltung, Dateimanager, Werkzeuge für die tägliche Arbeit, wie z.B. zum Auffinden von Dateien oder leicht bedienbare Editoren.



8. Zum Schluss noch einige Fähigkeiten von Linux

8.1 Personal Unix Workstations

Ob zu Hause oder am Arbeitsplatz, die Kombination aus Linux und einem auf Intel basierenden PC ergeben eine effiziente Unix Maschine. Linux ist ideal für Organisationen, die sich keine Sun oder HP Anlagen für jeden Techniker leisten können und denen X Terminals einfach zu unintelligent sind. Linux kann auch für Hybridanlagen verwendet werden, auf denen auch DOS und Windows Benutzer Zugriff auf ihre Daten haben und verschiedene Programme ausführen können.

8.2 X Terminal Clients

Linux' ausgezeichnete X11 Unterstützung macht es möglich, Anwendungen auf einem anderen Application Server (das kann von einer einfachen zweiten Linux Maschine bis zu einem Supercomputer mit vielen Prozessoren reichen) laufen zu lassen und es auf der lokalen Linux Anlage auf den Bildschirm zu bringen. Linux hat gegenüber den traditionellen X Terminals den Vorteil, dass es die komplette Bandbreite an Multimedia Einrichtungen wie Sound, Animationen und komplexe Grafikanwendungen unterstützt. Falls man Linux also einmal für mehr als eine X Terminal Funktion benötigt, können alle Schritte schnell und einfach ausgeführt werden. Also warum mehr bezahlen, wenn schnelle, zuverlässige und billige Hardware erhältlich ist?

8.3 X Application Server

Jeder Computer, der sich als X Terminal verwenden lässt, wird von der weiten Palette an X Anwendungen für Linux profitieren. Es macht genauso viel Sinn, Linux dazu zu verwenden, um RISC Workstations Zugriff auf Applikationen zu ermöglichen wie umgekehrt. Linux macht es Firmen aber auch möglich, PCs der 286 und 386 Reihe weiter zu verwenden: Sie können als einfache X Terminals fungieren und als DOS X Server betrieben werden. Das bedeutet, dass Linux auch eine günstige Wahl für den Betrieb eines billigen X Windows Netzwerkes ist.

8.4 Unix Entwickler Plattform

Linux besitzt alle Möglichkeiten professionelle Unix Entwicklung zu unterstützen. Die meisten gebräuchlichen Sprachen sind erhältlich, und viele weniger verbreitete ebenfalls. Ein Cross Compilation Environment ermöglicht es, Binärcodes für viele andere Linux Plattformen zu generieren. GNU C++, GNU Fortran 77, ADA, Pascal, Modula 2 und 3, TCL/Tk, Scheme und SmallTalk/X sind gratis erhältlich. Auch Interpreter für viele Sprachen werden im Sinne einer Zeitersparnis bei der Entwicklungsarbeit bereitgestellt. All das, gemeinsam mit flexiblen Script Shells, sprachsensitiven Editoren, Source Code Control Paketen und Dokumentationshilfen summiert sich zu einer Umgebung, die für den Programmierer vollständig konfigurierbar ist. Linux ist auch ein ideales Betriebssystem für Studenten. Es ermöglicht die vollständige Kontrolle über ein komplexes System, ohne andere Benutzer auf gemeinsamen Einrichtungen zu stören.

8.5 Kommerzielle Entwicklungen

Neben den anderen Entwicklersystemen existiert eine kommerzielle Implementierung von CA/Clipper (auf objektorientiertem Compiler basierendes erweitertes dBase und Fox), also können Clipper Entwickler ihre Software mit wenig oder gar keinen Änderungen auf Linux portieren. Daraus resultiert die gleiche Funktionalität, obgleich mit verbesserter Performance - 20 bis 200 Mal schneller als die DOS Version.

8.6 Network Server

Bildungseinrichtungen und kommerzielle Organisationen sind von Linux als Firmenserver außerordentlich begeistert. Für gemeinsame Speicherplatz- und Druckerbenutzung kann Linux so konfiguriert werden, daß es NFS, AppleTalk und NetBIOS Protokolle verwendet. Seine kostengünstige Anschaffung und die gute Performance machen Linux sehr attraktiv.

LAN Bridging für die Einrichtung von WANs ist auch eine gute Anwendungsmöglichkeit für Linux. Wird Linux auf RISC und Symmetric Mutli-Processing (SMP) Hardware verwendet, bringt es Flexibilität und einfache Handhabung in professionelle Umgebungen.

8.7 Internet Server

Linux hat seinen Ursprung im Internet und bietet daher überragende Internet Services. World Wide Web, Usenet News, Email, FTP und vieles andere werden mit Linux standardmäßig unterstützt. All das kann so konfiguriert werden, daß es entweder internen Benutzern eines Netzwerks den Zugang ermöglicht oder Informationen für das ganze Internet zugänglich macht. Kombiniert mit Einwählmodems (Benutzung von Mehrfachseriellverbindungen) wird Linux zu einem perfekten Internet Zugangsknoten. Ein Gutteil der kommerziellen Internetanbieter hat sich, ob seiner Zuverlässigkeit und Performance in diesem Bereich für Linux entschieden.

8.8 Terminal Server, Fax Server, Modem Server

Serielle Geräte und Telefone werden durch Linux einwandfrei unterstützt. Teure, speziell angefertigte Terminal Server (wie Annex) bieten üblicherweise SLIP, PPP, Direktverbindung, Dial-up Appletalk und The Internet Adapter Unterstützung. Linux verfügt ebenfalls über all diese Funktionen und beinhaltet zusätzlich Unterstützung für konfigurierbare Sicherheitsmaßnahmen, Authentifikation und Ereignisprotokollierung. Internetprovider haben die Möglichkeit, über 200 Modems mittels einem einzigen einfachen System zu verbinden, um eine verlässliche und Aufrechterhaltbare Einwählleitung zu sichern.

9. Literaturverzeichnis

Hetze, Sebastian „Linux Handbuch“ Müller-Hetze Verlag, Berlin 1997

Hantelmann, Fred „Linux für Durchstarter“ Springer Verlag, Heidelberg 1999

“c’t – PC Magazin“, Heft 12 – 1999

<http://linux-download.virtualave.net/linux-geschichte.htm>

<http://www.linuxinfo.de/de/forum/listboard.php3>

<http://www.webtraxx.de>