

EINFÜHRUNG
IN DAS
BETRIEBSSYSTEM LINUX

insbesondere
Mac Os X

INHALTSVERZEICHNIS

1. Vorwort	3
1.1. WAS IST LINUX?	3
1.2. WAS KANN LINUX?	3
1.3. GESCHICHTE	4
2. GNU-Tools	5
2.1. ERLÄUTERUNG UND VORWORT ZUR (UNIX-)SHELL	5
2.2. DIE WICHTIGSTEN GNU-KOMMANDOS UND TASTATURBEFEHLE	5
2.2.1. <i>Online-Hilfe</i>	5
2.2.2. <i>Dateihandling</i>	5
2.2.3. <i>Verzeichnisnavigation</i>	6
2.2.4. <i>Andere Befehle (Benutzerverwaltung, Prozesse)</i>	6
2.3. ÜBERSICHT ÜBER VERSCHIEDENE LINUX SHELLS	6
2.4. DIE WICHTIGSTEN EIGENSCHAFTEN DER BOURNE-SHELL (SH)	7
2.4.1 <i>Metazeichen</i>	7
2.4.2 <i>Vordefinierte Shellvariablen</i>	7
2.4.3 <i>Übersicht der Befehle</i>	8
3. Dateisystem	9
3.1. VERGLEICH DER POPULÄRSTEN DATEISYSTEME	9
3.2. VERZEICHNISSTRUKTUR	9
4. Vergleich Linux / Mac OS X	11
4.1 MAC OS X	11
4.1.1 <i>Merkmale von Mac OS X</i>	11
4.2 Linux	11
4.2.1 <i>Merkmale von Linux</i>	11
4.3 Tabellarischer Vergleich (nur Unterschiede)	11
5. Anhang	12
5.1. LINKS	12
5.2. BUCHEMPFEHLUNGEN	12

1. Vorwort

1.1. Was ist Linux?

Linux ist erwiesenermaßen eines der schnellsten und stabilsten Betriebssysteme der Welt, zudem es auch noch kostenlos ist. Ein Betriebssystem ist die Software, die es dem User ermöglicht mit der Hardware zu interagieren, wodurch das Arbeiten mit einem Rechner erst ermöglicht wird.

Linux kann man entweder komplett aus dem Internet runterladen oder als Distribution auf CDs erwerben. Eine Distribution ist eine Sammlung von Programmen und Linux-Stücken aus dem Internet. Jeder kann seine eigene Distribution machen, es ist jedoch einfacher einfach eine bekannte Distribution zu erwerben.

Folgende sind die bekanntesten Distributoren:

Caldera	http://www.caldera.com/	Aktuelle Version: 2.3
Debian	http://www.debian.org/	Aktuelle Version: 2.2r3
Redhat	http://www.redhat.com/	Aktuelle Version: 7
Slackware	http://www.slackware.com/	Aktuelle Version: 7.1
SuSE	http://www.suse.de/	Aktuelle Version: 7.1

1.2. Was kann Linux?

- **Echtes Multitasking**
Verschiedene Programme laufen zur gleichen Zeit nebeneinander ab.
- **Multi User System**
Mehrere Benutzer können zur gleichen Zeit auf der gleichen Maschine Programme ausführen.
- **Multi-Processing**
Symmetric Multi-Processing für mehrere Prozessoren.
- **Geschwindigkeit**
Linux verfügt wahrscheinlich über den schnellsten 100Mbit-Ethernet TCP/IP Code der Welt.
- **Sicherheit**
Prozesse können isoliert und bei einem Programmabsturz beendet werden, Linux läuft weiter...
- **Betriebskomfort**
Um eine neue Software zu installieren muss Linux nicht heruntergefahren werden.
- **Dateisysteme**
Über 20 Dateisysteme werden unterstützt, darunter auch Amiga, MAC OS, FAT32 und NTFS.
- **Vielfältigkeit**
Bis zu 64 virtuelle Konsolen sind möglich.
- **Clustering**
Cluster von beliebiger Größe.

- **Netzwerkstärke**
Zahlreiche klassische und moderne Protokolle (Ethertalk, Appletalk, TCP/IP,...) und Netzwerkeigenschaften (Masquerading, Tunneling, Forwarding,...) werden unterstützt.
- **Vorteilhaftes Dateisystem**
Case-sensitive Dateinamen, Unterstützung von 256 Zeichen, Geräte wie CDROM-Laufwerke oder Festplatten können im Dateiverzeichnis angesprochen werden.

1.3. Geschichte

Meilensteine von „Linux“:

1969-70	Linux's Anfang wurde mit dem Betriebssystem „Multics“ geschaffen. Bei Multics handelte es sich um ein System das noch mit Lochkarten arbeitete, es wurde von AT&T und Western Electric weiterentwickelt. 1970 gab B. Kernighan dem System dem Namen Unix. Die Merkmale von Unix waren folgende: <ul style="list-style-type: none"> • Eine Hierarchische Anordnung der Dateisysteme • Die Ausgabe von Daten an die Peripherie(-geräte) und Dateien sollte gleich sein • Prozesse und Programme sollten miteinander kommunizieren • Multitasking • Persönliche Shell's • Der Compiler sollte für viele Programmiersprachen tauglich sein • Portabilität
1971	Unix wurde vom Assembler-Code in die gerade neu entwickelte Programmiersprache C protiert. Dadurch wurde es möglich Unix einfach auf die bestehende Hardware anzupassen.
1975	AT&T darf aufgrund von vertraglicher Bindungen mit der US-Regierung nicht vermarktet werden, daher stellt AT&T den Source-Code für wenig Geld den Universitäten zur Verfügung. Unix wird von den Studenten sehr schnell als flexible Umgebung angenommen und es entstehen weitere Unix-Versionen, wobei die Version 7 sehr erfolgreich wird. Verschiedene Firmen entwickeln ihre eigenen Unix-Derivate.
1984	Durch einen Gerichtsbeschluss wird AT&T erlaubt Unix zu vermarkten. AT&T verkauft sein Unix 5. Auch Sun trägt mit SunOS stark zur Ausbreitung von Unix bei.
GNU!	Free Software Foundation (FSF) wird gegründet und initiiert ein Projekt names GNU (GNU is not Unix) mit dem Ziel ein freies Unix zu schaffen.
1987	AT&T verbietet die Verwendung des Source-Code von Unix an Universitäten. Minix wird daraufhin entwickelt, es bietet die gleiche Funktionalität wie Unix 7 mit dem Unterschied, dass es komplett ohne AT&T Code auskommt. Unter news:comp.os.minix entsteht im Usenet eine Mimix-Gemeinde die den Mimix-Kernel weiterentwickelt und Anwendungen schreibt.
1991	Linus Torvalds hat als Student der Computerwissenschaften an der Universität Helsinki Mimix installiert. Er beginnt im Sinne der FSF einen Unix-Ähnlichen Kernel zu bauen um „ein besseres Minix als Minix“ zu bauen. 1992 stellt Linus die Version 0.02 der Betriebssystemes Linux im Internet frei zur Verfügung.
1994	Linux 1.0 wird freigegeben.
1998	KDE 1.0 wird freigegeben.
1999	Der Kernel 2.2 wird freigegeben. IBM kündigt seine Unterstützung für Linux an.
2001	Der Kernel 2.4 wird im Januar freigegeben!

2. GNU-Tools

2.1. Erläuterung und Vorwort zur (Unix-)Shell

Wie die Eingabeaufforderung unter Windows gibt es unter Linux eine Möglichkeit Befehle direkt an das Betriebssystem zu geben.

Ein Befehl, der meinst englisch ist (z.B. copy), wird mit der Tastatur eingegeben und dann vom System ausgeführt. Der Vorteil gegenüber der Steuerung mit der Maus in einer grafischen Umgebung besteht u.a. darin, dass man sehr viel schneller Befehle an das System übergeben kann. Durch die sog. Shell ist man auch in der Lage Bereiche des Betriebssystems zu erreichen, die mit der Maus entweder gar nicht oder nur sehr viel schwieriger erreichbar wären.

Da Linux in erster Linie nur das Herz - der Kernel - ist und gar keine grafische Oberfläche besitzt, ist im Prinzip die Shell das eigentliche Werkzeug um Linux nutzen zu können und muss daher auch entsprechend beachtet werden, auch in Zeiten wo Linux immer mehr durch grafische Oberflächen kontrollierbar wird.

Auch in MacOS X ist es möglich alternativ zur Steuerung mit der Maus die Shell zu benutzen um bestimmte Arbeiten zu erleichtern.

Einfache praktische Beispiele um die Leistungsfähigkeit der Shell zu demonstrieren:

1. In einem Ordner möchte man alle Dateien eines bestimmten Typs filtern und umbenennen.
Für diese Operation benötigt man einen simplen Ausdruck, mit der Maus würde man bei einer großen Dateimenge stunden brauchen (dieses Beispiel ist auch unter DOS/Windows realisierbar).
2. Man möchte aus einer gecacheten Webseite mit mehr als 200 html-Seiten, nach bestimmtem Worten suchen und dann die genauen Zeilenangaben der gefundenen Stellen in eine Textdatei speichern um später gezielt auf der Webseite die Abschnitte zu lesen, die die entsprechenden Worte enthalten.
Auch dies ist mit einem (nicht ganz so simplen) Ausdruck unter Linux realisierbar, jedoch nicht ohne weitere Programme unter DOS, Windows oder Apple.

Anmerkung:

Diese Form der Steuerung ist unter MacOS 9 über Umwegen und mit Einschränkungen mit AppleScript zu realisieren und unter Windows mit den Scriptsprachen VBS und JS.

2.2. Die wichtigsten GNU-Kommandos und Tastaturbefehle

Wechseln der Virtuellen Konsole: [ALT] + F1, F2, F3 ...

Wort-Vervollständigung: [EINGABE] + [TAB]

2.2.1. Online-Hilfe

Im Prinzip kann man zu jedem GNU-Kommando eine Online-Hilfe abfragen indem man „**man [befehl]**“ eingibt, alternativ kann „**whatis [befehl]**“. „**whatis**“ greift wie „**man**“ auf die Online-Hilfe zurück gibt jedoch nur eine kurze Info zum Befehl aus.

2.2.2. Dateihandling

Anmerkung: Verzeichnisse sind auch Dateien! Wenn also von Dateien die rede ist, lassen sich die Befehle ebenso auf Verzeichnisse anwenden.

Kennzeichnung der Befehle und Parameter:

- Befehle sind **fett**
- Parameter sind durch eckige Klammern ([]) gekennzeichnet
- unbedingt benötigte Parameter sind **[fett]**
- nicht unbedingt benötigte Parameter sind **[fett kursiv]**,

Auflisten von Dateinamen im aktuellen Verzeichnis:	ls [optionen] [dateien] Beispiele: ls -l ls -il
Löschen von Dateien	rm [optionen] [dateien] rmdir [optionen] [verzeichnis]
Kopieren von Dateien:	cp [optionen] [ursprung 1.. 2..] [ziel]
Bewegen (Umbenennen) von Dateien:	mv [optionen] [ursprung 1.. 2..] [ziel]
Dateien Linken (Alias Namen geben):	ln [optionen] [ursprung 1.. 2..] [ziel]
Suchen von Dateien:	find [optionen] [verzeichnis] [bedingungen]
Suchen in Dateien:	grep [optionen] [ausdruck] [dateien] egrep : extended grep fgrep : fast grep (nur strings)
Finden der genauen Adresse (Pfad) eines Befehls	whereis [befehl]
Ändern der Rechte einer Datei:	chmod [perm.] [datei]
Ändern der Gruppenzugehörigkeit einer Datei	chgrp [gruppe] [datei]
Ändern von Besitzer und Gruppe einer Datei	chown [besitzer:gruppe] [datei]
Anlegen einer leeren Datei oder eine bestehende Datei auf die aktuelle Zeit setzen	touch [datei]

2.2.3. Verzeichnisnavigation

Anzeigen des aktuellen Verzeichnis:	pwd
Wechseln zum Verzeichnis:	cd [verzeichnis]
Verzeichnis anlegen:	mkdir [optionen] [verzeichnis]
Verzeichnis löschen:	rmdir [optionen] [verzeichnis]
Verzeichnisse vergleichen:	diff [optionen] [verzeichnis1] [verzeichnis2]

2.2.4. Andere Befehle (Benutzerverwaltung, Prozesse)

Mounten von Dateisystemen	mount [optionen] [dateisystem]
Unmounten von Dateisystemen	umount [dateisystem]
Freien Speicherplatz erfragen	df
Eigenes Paßwort ändern	passwd [login-name]
Prozessverwaltung	ps [optionen]
Prozess beenden	kill [signalnr] [pids]
Formatieren von Disketten/Dateien	/bin/format [optionen] [gerät] fdformat [-n] [gerät]
X Window starten	startx

2.3. Übersicht über verschiedene Linux Shells

	Bourne-Shell	Korn-Shell	C-Shell	Bourne-Again-S.	TC-Shell
	Sh	ksh	Csh	Bash	tcsh
Alias-Mechanismus		X	X	X	X
Arithmetische Operatoren		X	X	X	X

Arrays		X	X		X
Ausführung von .logout			X		X
Autoload-Funktionen		X			
CDPATH	X	X	X	X	X
Dateinamenexpandierung		X	X	X	X
Directorystack (dirs, pushd)			X	X	X
Ein, / Ausgabeumlenkung	X	X	X	X	X
Exportieren von Aliasen		X			
Formatierte Variablen		X			
Funktionen	X	X		X	
Funktionslokale Variablen		X		X	
History-Mechanismus		X	X	X	X
Integer-Arithmetik		X	X	X	X
Jobkontrolle		X	X	X	X
Kommando select		X			
Kommando-Substitution	X	X	X	X	X
Kommando-Tracking	X	X	X	X	X
Kommandozeilen-Editor		X		X	X
Koprozesse		X			
Login-Dateien	X	X	X	X	X
Metazeichen [!..]	X	X		X	[^..]
Noclobber, ignoreof		X	X	X	X
RC-Dateien (runtime commands)		X	X	X	X
Readonly-Variablen	X	X		X	X
Signalbehandlung	X	X	X	X	X
Signalnamen		X	X	X	X
String-Operatoren		X		X	
Tilde-Expandierung		X	X	x	X
Tippfehler-Korrektur					X
Variabler Prompt-String		X	X	X	X
Wort-Vervollständigung	X	X	X	X	X
Zufallszahlen (random)		X		X	

2.4. Die wichtigsten Eigenschaften der Bourne-Shell (sh)

Im folgenden werden wir uns nur mit der Einfachsten der Linux-Shells befassen, der Bourne Shell. Selbst diese Shell bietet weitaus mehr Möglichkeiten als die Eingabeaufforderung unter Windows.

Mit diesen Zeichen und Befehlen lassen sich Makros erstellen die einfach Befehle abarbeiten, d.h. man hat selbst ohne eine Programmiersprache zu benutzen die Möglichkeit sich wiederholende Aufgaben zu automatisieren.

2.4.1 Metazeichen

[befehl] >[datei]	Output vom Befehl in Datei umlenken (Datei überschreiben)
[befehl] >> [datei]	Output vom Befehl in Datei umlenken (am Ende der Datei hinzufügen)
[befehl] <[datei]	Input vom Befehl in Datei umlenken (Datei überschreiben)
*	Platzhalter für irgendein Zeichen oder kein Zeichen
?	Platzhalter für ein beliebiges Zeichen
[befehl1];[befehl2]	Nach Beendigung vom Befehl 1 wird Befehl 2 ausgeführt
[befehl1]&	Befehl wird im Hintergrund ausgeführt
#	Kommentar
[befehl1]&&[befehl2]	Befehl 2 wird nur ausgeführt wenn Befehl 1 erfolgreich war
[befehl1] [befehl2]	Befehl 2 wird nur ausgeführt wenn Befehl 1 nicht erfolgreich war

2.4.2 Vordefinierte Shellvariablen

CDPATH Suchverzeichnis für „cd“

HOME	Home-Verzeichnis des Benutzers
IFS	Wort-Trennzeichen, Voreinstellung: Leer-, Tab- und Neuezeile-Zeichen
LOGNAME	Login-Name des Benutzers
MAIL	Pfad zur Mailbox-Datei
MAILCHECK	Zeitintervall für Mail-Prüfung
MAILPATH	Pfadnamen zu Mailbox-Dateien
PATH	Suchverzeichnisse für Programme
PS1	1. Prompt
PS2	2. Prompt
SHACCT	Datei für Abrechnungsinfo
SHELL	Pfadname der Shell
TERM	Terminal des Benutzers
TZ	Zeitzone

2.4.3 Übersicht der Befehle

Folgendes sind die Befehle die Ihnen in der Bourne-Shell zur Shell-Programmierung zu Verfügung stehen. Andere Shells bieten weitaus mehr Befehle an:

- if-Anweisung
- case-Anweisung
- while-Schleife
- until-Schleife
- for-Schleife

Zusätzliche Keywords

- break
- continue
- fi, esac, done

Es können außerdem eigene Funktionen deklariert werden.

3. Dateisystem

3.1. Vergleich der populärsten Dateisysteme

Linux-Dateisysteme:

	Minix FS	Ext FS	Ext2 FS
max. Größe der Partition	64 MB	2 GB	4 TB
max. Größe einer Datei	64 MB	2 GB	2 GB
max. Länge einer Datei	16/30 c	255 c	255 c

Windows-Dateisysteme:

	Fat16 FS	Vfat FS	Fat32 FS	NTFS
max. Größe der Partition	2 GB	2 GB	>2 GB	>2 GB
max. Größe einer Datei	2 GB	2 GB		
max. Länge einer Datei	8+3 c	255 c	255 c	255 c

Macintosh-Dateisysteme:

	HFS	HFS+
max. Größe der Partition	2 GB	1 TB
max. Größe einer Datei	2 GB	2 GB
max. Länge einer Datei	31 c	255 c

3.2. Verzeichnisstruktur (Unix)

/	Wurzel- (root-) Verzeichnis. Dies ist die oberste Verzeichnisebene. Es enthält in Unterverzeichnissen alle gemounteten Volumes und alle Dateien des Dateisystems.
/bin/	Hier befinden sich meinst die ausführbaren Systemdateien oder kleinere Programme und Tools der Distribution.
/boot/	Boot-Partition oder Verzeichnis mit dem Boot-Sektor und dem kompilierten Kernel „ vmlinuz “. u.a. auch durch lilo (siehe /etc/lilo.conf) benötigt.
/dev/	Hier befinden sich sämtliche Devices (die Hardware) die angesprochen werden können. Beispiel: - /dev/hda1 steht für die erste Partition auf der ersten physikalischen Festplatte (primary / master). Diese kann durch mount /dev/hda1 gemountet werden.
/etc/	In diesem Verzeichnis befinden sich die Konfigurationsdateien zu den jeweiligen Programmen und zum System. Beispiele: - Devices die gemountet werden sollen stehen in fstab - Services wie FTP , TIME , HTTP können in inetd.conf , der Konfigurationsdatei von inetd , verfügbar gemacht werden. - In profile.* werden globale Einstellungen vorgenommen, z.B. Proxy-Server oder Änderung der vordefinierten Shellvariablen (siehe 2.4.2). - passwd und shadow sind für Benutzernamen und deren Passwörter zuständig.

/home/	Home Verzeichnisse für Benutzer.
/mnt/	„Mountpoint“. Hier werden zusätzliche Devices gemountet.
/opt/	Zusätzliche Anwendungen, wie z.B. Netscape oder Star Office
/root/	Home Verzeichnis des Systemadministrators (root).
/sbin/	zuständig für Shells und Scripting. Entsprechende Programme und Konfigurationsscripts sind hier zu finden.
/sbin/init.d/	Systemscripte
/sbin/init.d/rc?.d/	Die Runlevels (Aufruf eines Runlevels mit init [runlevel nr]), Konfiguration: /etc/inittab Runlevel 0: Halt Runlevel 1: Einzelbenutzermodus Runlevel 2: Mehrbenutzermodus, ohne Netzwerk Runlevel 3: Wie 2 nur mit Netzwerk Runlevel 4: Unbenutzt, bzw. Distributionsabhängig Runlevel 5: Wie 3 nur mit Xwindows Runlevel 6: Reboot Die Belegung der Runlevels kann von Distribution zu Distribution unterschiedlich sein, siehe auch http://linux.rice.edu/help/tips-services.html .
/tmp/	Temporäres Arbeitsverzeichnis. (Achtung mod ist meist 777)
/usr/	Anwendungen
/var/	Logdateien, HTTPD-Stammverzeichnis (bei Debian /var/www/) und andere Informationen zu den installierten Programmen

3.3. Verzeichnisstruktur (Mac OS X)

4. Vergleich Linux / Mac OS X

4.1 Mac OS X

4.1.1 Merkmale von Mac OS X

SYSTEMAUFBAU:

- **Grundstruktur:** entspricht dem früheren Next-OS, und hat dadurch seine Stärken beim Speicherschutz, preemptivem Multitasking und symmetrischem Multiprocessing.
- **Mac OS X-Kern:** Darwin, bestehend aus dem Kernel Mach 3.0 und das Betriebssystem Free-BSD.
- Mehrbenutzersystem wie Unix

Der Mac OS X-Kern ist wie Linux frei verfügbar und darf durch Vergabe von Open-Source-Lizenzen weiterentwickelt werden.

WINDOWING UND GRAFIKKOMPONENTEN (AQUA)

- **2D:** Adobes Portable Document Format (PDF) , Quartz (Grafikbibliothek)
- **3D:** OpenGL
- **Multimatia:** Quicktime-Technologie

KOMPATIBILITÄT:

Es werden vier Arten von Anwendungen unterstützt

- Carbon-Anwendungen („echte“ Mac OS X-Anwendungen, die auch lauffähig unter Mac OS 9 sind)
- Java-Anwendungen
- Cocoa-Anwendungen (abstammend von NextStep)
- Klassische Mac OS 9-Anwendungen (Emuliert durch Mac OS 9.1)

Zusätzlich werden noch Unix-Anwendungen unterstützt, die mittels des GNU-C Compiler übersetzt werden.

4.2 Linux

4.2.1 Merkmale von Linux

→ siehe 1.2

4.3 Tabellarischer Vergleich (nur Unterschiede)

	Mac OS X	Linux/Unix
Kernel	Nicht ganz so up to date wie der von Linux	Immer up to date, jedoch wird meist nie der neuste Kernel in sicherheitsrelevanten Systemen eingesetzt (Serverbetrieb)
Systemaufbau	Free BSD-Stucktur „Unix-Konform“	Distributionsabhängig, im Prinzip aber wie Mac OS X
Grafische Oberfläche	Nur Aqua	Beliebige Oberfläche, X wird meist

		eingesetzt
Kompatibilität	Mac OS X, Mac OS 9, NextStep, Java, Unix	Unix, Java

5. Anhang

5.1. Links

- Deutsches HOWTO-Projekt - <http://www.tu-harburg.de/~semb2204/dlhp/>
- Lehrtext: UNIX-Einführung - <http://www.zi.unizh.ch/publications/unix/>
- Linux Dokumentationen u.a. howtos - <http://www.linuxdoc.org/>

5.2. Buchempfehlungen

- **Linux. Installation, Konfiguration, Anwendung**

Michael Kofler
ISBN: 3827316588

Kurzbeschreibung

Den *Kofler* in seiner fünften Auflage zu rezensieren, ist eine leichte Aufgabe. Das Werk *Linux - Installation, Konfiguration, Anwendung* gilt nicht umsonst als das Standardbuch für Linux-Einsteiger. Auf inzwischen fast 1.100 Seiten wird so ziemlich alles angesprochen, was der Privatanwender für den erfolgreichen Start benötigt. Und mit der aktuellen Auflage geht Michael Kofler auch noch darüber hinaus.

Quelle: amazon.de

- **Linux – Unix Kurzreferenz**

Helmut Herold
ISBN: 3827315360

Kurzbeschreibung

Diese Kurzreferenz enthält alle Kommandos in übersichtlicher Form zum Nachschlagen für die tägliche Arbeit mit Linux/Unix. Die 2. Auflage wurde aktualisiert und erweitert. Sie ist die Zusammenfassung aller Befehle der Bücher der Reihe "Linux/Unix und seine Werkzeuge".

Quelle: amazon.de