



# GRUNDLAGEN VON LINUX

# INHALTE

1. Geschichte
2. Kernel
3. Verzeichnisstruktur
4. Shell Befehle
5. Debian Softwarepakete

# AM ENDE DIESES MODULS ...

... haben Sie Ubuntu Linux in einer VM eingerichtet

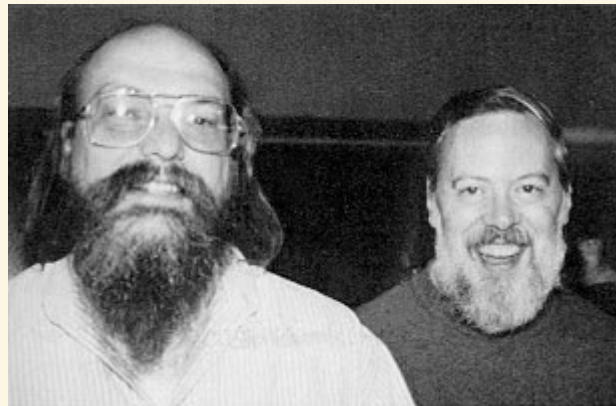
... verstehen Sie die Linux Verzeichnisstruktur

... kennen Sie die wichtigsten Linux Shell Befehle

... können Sie Linux Softwarepakete installieren

# GESCHICHTE

Das Unix-Betriebssystem wurde von Ken Thompson und Dennis Ritchie (Firma AT&T Bell Labs) im Jahr 1969 entwickelt und 1970 erstmals veröffentlicht.



# BSD

Im Jahr 1977 veröffentlichte die Berkeley Universität ein UNIX-ähnliches System, Berkeley Software Distribution (BSD). Da BSD zu dieser Zeit AT&T Coding enthielt war es allerdings nur mit gültiger UNIX Lizenz nutzbar.



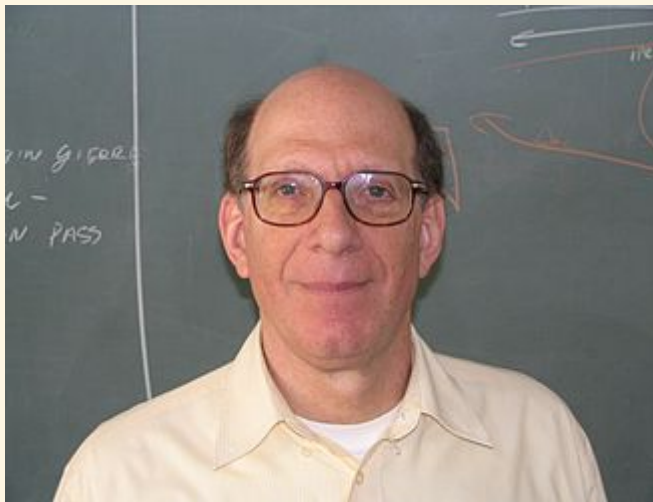
# GNU

1983 startete Richard Stallman das GNU-Projekt, um ein freies UNIX-ähnliches Betriebssystem zu erstellen. Dabei wurde unter anderem eine Compiler Toolchain, ein Codeeditor und eine Shell entwickelt. Allerdings fehlte es dem Projekt lange an einem vollständigen Betriebssystemkern (Kernel).



# MINIX

1987 veröffentlichte Andrew S. Tanenbaum MINIX, ein Unix-ähnliches System für den akademischen Gebrauch. Während der Quellcode für das System verfügbar war, wurde eine Modifizierung nicht gestattet.





# 25.08.91

Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things). [..]

Linus (torvalds@kruuna.helsinki.fi)



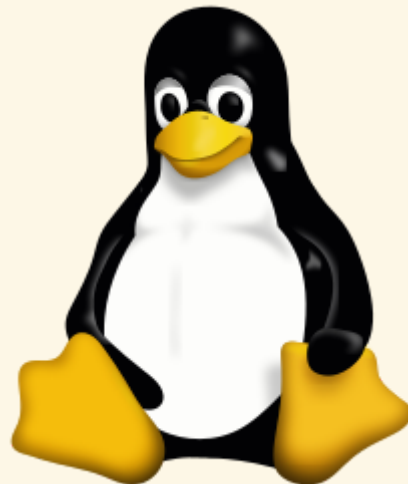
# LINUS BENEDICT TORVALDS

- Geboren: 29.12.1969 (Alter 48)
- Geburtsort: Helsinki, Finnland
- Chefentwickler des Linux Kernel
- Erfinder von GIT



# LINUX

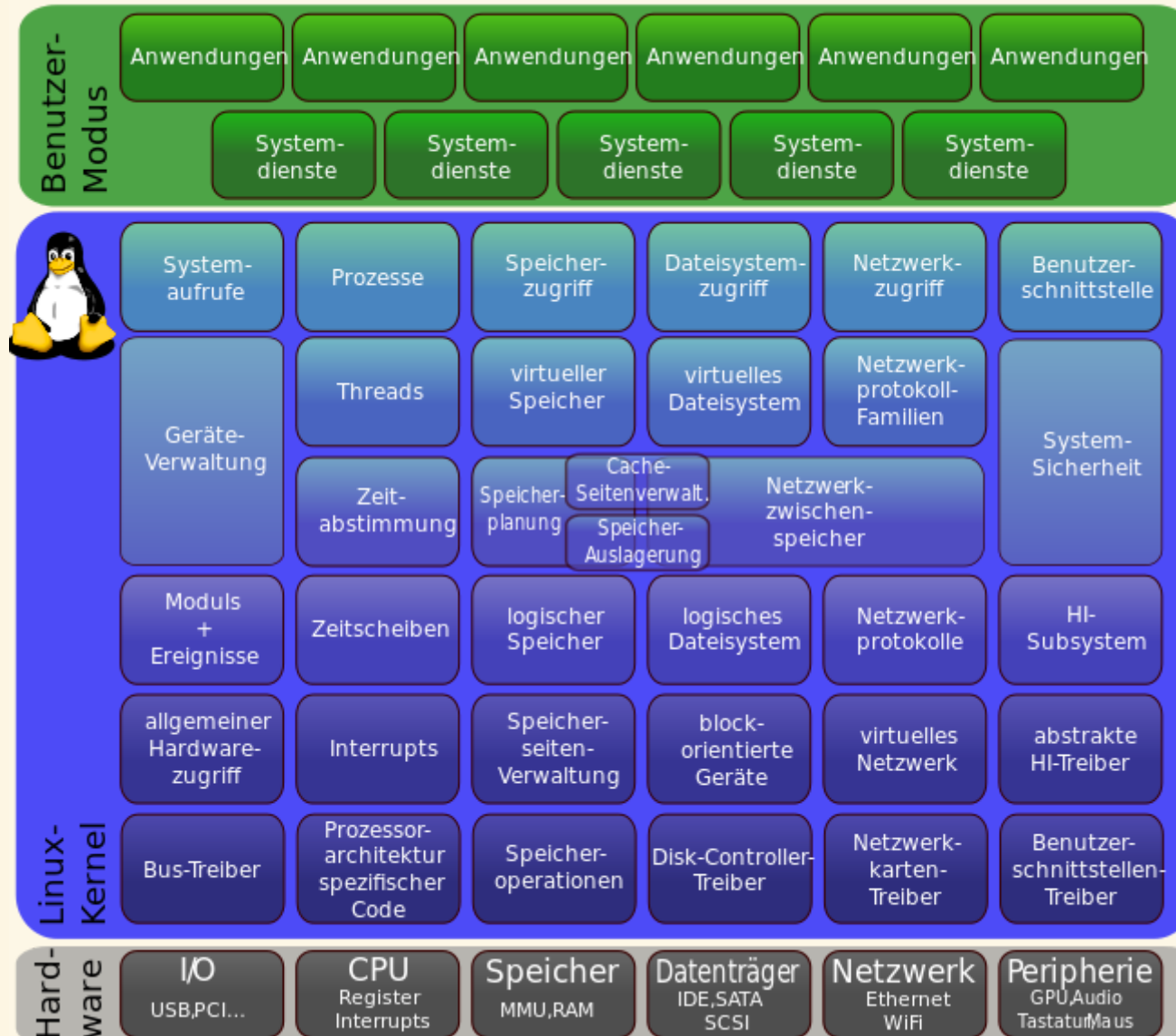
Torvalds entwickelte den Kernel und komplettierte diesen mit Software aus dem GNU Projekt zu dem Betriebssystem Linux. Linux ist ein freies Betriebssystem, das unter der GPL Lizenz zur Verfügung gestellt wird.



# EINSATZGEBIETE:

- Server Betriebssysteme (Marktanteil ca. 35% im Jahr 2015 lt. W3Techs, Dunkelziffer weitere 30%)
- Smartphones (Android)
- Embedded Geräte (auch Raspberry Pi!)
- Desktop Betriebssysteme
- Supercomputer

# KERNEL





# DISTRIBUTIONEN

Eine Distribution fasst den Linux-Kernel mit verschiedener Software zu einem Betriebssystem zusammen, das für die Endnutzung geeignet ist. Dabei passen viele Distributoren und versierte Benutzer den Kernel an ihre eigenen Zwecke an. Beliebte Distributionen:

- Debian
- Ubuntu
- Suse
- Redhat
- usw..





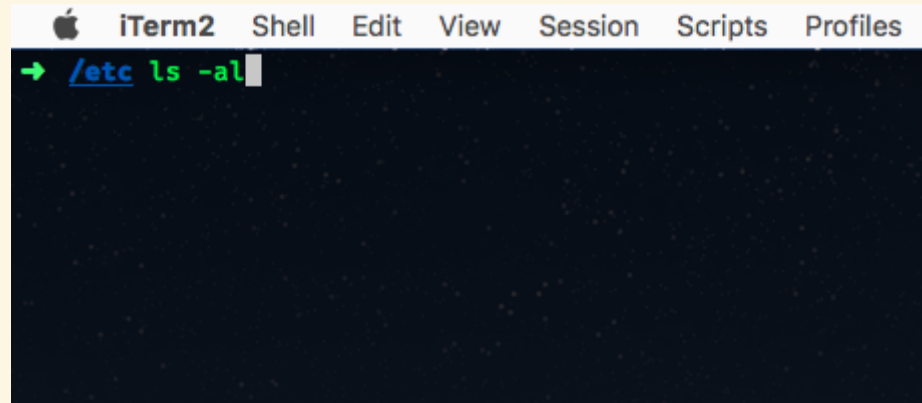
# VERZEICHNISSTRUKTUR

Verzeichnis	Beschreibung
/bin	Von: <i>binaries</i> ; enthält für Linux unverzichtbare Programme
/boot	Enthält zum Booten benötigte Dateien; Kernel, initiale Ramdisk
/dev	Von: <i>devices</i> ; enthält alle Gerätedateien, über die die Hardware im Betrieb angesprochen wird
/etc	Von: <i>editable text configuration</i> enthält Konfigurations- und Informationsdateien des Basissystems
/home	Von: <i>home-directory</i> (Heimatverzeichnis); enthält pro Benutzer ein Unterverzeichnis
/lib	Von: <i>libraries</i> ; enthält Bibliotheken für dynamisch gelinkte Programme des Basissystems

Verzeichnis	Beschreibung
/proc	Von: <i>processes</i> ; enthält Schnittstellen zum aktuell geladenen Kernel und seinen Prozeduren
/root	Ist das Homeverzeichnis des Superusers (root)
/sbin	Von: <i>system binaries</i> ; enthält Programme für essentielle Aufgaben der Systemverwaltung; Beispiele: shutdown
/usr	Von: <i>user</i> ; enthält die meisten Systemtools, Bibliotheken und installierten Programme; der Name ist historisch bedingt - früher, als es /home noch nicht gab, befanden sich hier auch die Benutzerverzeichnisse
/var	Von: <i>variable</i> (variabel); enthält Dateien die zur Laufzeit des Systems geschrieben werden; Beispiele: Log-Dateien (/var/log)

# ÜBUNG 1 - SETUP

# SHELL & TERMINAL



Die **Shell** oder auch **Kommandozeileninterpreter** ist ein Programm das Textbefehle interpretiert und dem Betriebssystem zur Ausführung übergibt. Bevor es grafische Benutzeroberflächen (GUIs) gab, war dies die einzige Eingabemöglichkeit um mit dem Computer zu interagieren.

Das **Terminal** oder auch **Terminalemulator** ist ein Programm das Keyboardeingaben an die Shell übergibt. Es gibt verschiedene Terminalemulator-GUIs die erweiterte Features wie z.B. Scrolling und Tabs unterstützen.

# TERMINAL BASICS

Befehl	Beschreibung
<code>ctrl + c</code>	Beendet den gestarteten prozess; Leert die Eingabezeile
<code>ctrl + l</code>	Leert den Terminalscren
<code>↑, ↓</code>	Navigiert in der Historie der letzten Befehle
<code>ctrl+r,</code> <code>ctrl+s</code>	Sucht in der Befehlshistorie

# SHELL BASICS

Befehl	Beschreibung
<code>man</code>	<b>manual</b> pages - öffnet die Befehlsbeschreibung
<code>--help</code>	Konvention vieler CLIs um eine Hilfe anzuzeigen

## Beispiele:

- `man git`
- `git --help`

# DATEIEN & ORDNER

Befehl	Beschreibung	Beispiel
<code>pwd</code>	<b>print working directory</b> - Zeigt den aktuellen Pfad an.	<code>pwd</code>
<code>mkdir</code>	<b>make directory</b> - legt ein neues Verzeichnis an	<code>mkdir myfolder</code>
<code>touch</code>	Legt eine leere Datei an oder modifiziert deren Zeitstempel	<code>touch myfile</code>
<code>cd</code>	<b>change directory</b> - wechselt den Pfad	<code>cd myfolder, cd ..</code>
<code>ls</code>	<b>list</b> - zeigt die Inhalte eines Verzeichnisses an	<code>ls -a</code>
<code>rm</code>	<b>remove</b> - entfernt Dateien/Ordner	<code>rm -rf myfolder</code>



Befehl	Beschreibung	Beispiel
mv	<b>move</b> - verschiebt Dateien/Ordner	mv folder1/* folder2/
cp	<b>copy</b> - kopiert Dateien/Ordner	cp -R folder1 folder2

Befehl	Beschreibung	Beispiel
<code>find</code>	Datei-/Ordnersuche	<code>find / -name "myfile"</code>
<code>df</code>	display free disk space - freien Speicherplatz der Laufwerke anzeigen	<code>df -H</code>
<code>du</code>	display disk usage - Speicherverbrauch von Dateien/Ordnern anzeigen	<code>du -H</code>

# ÜBUNG 2 - BASICS

# PIPES/STREAMS

Befehl	Beschreibung	Beispiel
	Pipe Operator - verbindet Ein- und Ausgabe zweier Programme	<code>cat file   grep "text"</code>
grep	Sucht in einer Datei bzw. einem Stream	<code>grep -i "hello" myfile</code>
less	Zeigt eine große Datei bzw. einen Datenstrom scrollbar an	<code>cat longfile   less</code>
tail; head	Zeigt den Anfang/das Ende einer Datei bzw. eines Datenstroms an	<code>cat file   tail -n 5</code>
xargs	Führt einen Befehl für jede Zeile eines Datenstroms aus	<code>seq 10   xargs -I {} echo "Zahl {}"</code>

# DATEIRECHTE

Befehl	Beschreibung
chmod	<b>change mode</b> - ändert die Zugriffsberechtigung
chown	<b>change owner</b> - ändert den Owner (Datei/Ordner)

	read <i>r</i> :4	write <i>w</i> :2	execute <i>x</i> :1	command symbolic	command octal
users	✓			chmod u=r	chmod 400
groups		✓		chmod g=r	chmod 040
others			✓	chmod o=r	chmod 004

- Oktalberechnung:  $rw\bar{x} = 4 + 2 + 1 = 7$

# PROZESSE

Befehl	Beschreibung	Beispiel
<code>top</code>	Zeigt aktive Prozesse <i>live</i> an	<code>top (o, cpu)</code>
<code>ps</code>	<b>process status</b> - zeigt aktive Prozesse als Liste	<code>ps -aux</code>
<code>kill</code>	Beendet einen Prozess	<code>kill 1001</code>
<code>netstat</code>	<b>network status</b> - zeigt offene Netzwerkverbindungen	<code>netstat -tulpen</code>
<code>nohup &amp;</code>	<b>no hangup</b> - startet einen Prozess im Hintergrund	<code>nohup myapp &amp;</code>
<code>which</code>	Zeigt den Speicherort eines Programms an	<code>which ps</code>

# ÜBUNG 3 - PIPES



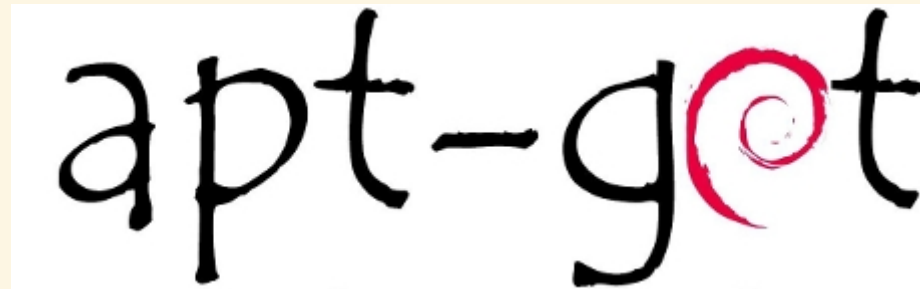
# SYSTEM & NUTZER

Befehl	Beschreibung	Beispiel
<code>adduser</code>	Fügt einen Nutzer hinzu	<code>adduser hugo</code>
<code>deluser</code>	Entfernt einen Nutzer	<code>deluser --remove-home hugo</code>
<code>su</code>	<b>substitute</b> - wechselt den Benutzer	<code>su hugo</code>
<code>whoami</code>	Zeigt den aktuell aktiven Nutzer an	
<code>sudo</code>	Führt einen Befehl als Root aus	<code>sudo deluser ...</code>
<code>shutdown</code>	Beendet/Startet den Computer neu	<code>shutdown -h now,</code> <code>shutdown -r now</code>

## UND VIELE WEITERE..

- awk, sed, jq, seq, uniq, date, ...

# DEBIAN SOFTWAREPAKETE



Befehl	Beschreibung
apt-get update	Neueinlesen der Paketlisten
apt-cache search	Sucht in der Paketliste
apt-get install	Installation eines Paketes
apt-get remove	Deinstallation eines Paketes
apt-get purge	Wie remove, entfernt aber zusätzlich die Konfiguration

Befehl	Beschreibung
apt-get upgrade	Update vorhandener Pakete; hält Pakete stabil, siehe dist- upgrade
apt-get dist- upgrade	Update vorhandener Pakete, installiert wenn nötig zusätzliche Pakete oder entfernt nicht mehr benötigte

**END**