

Linux als Arbeitsplattform im Wissenschaftlichen Umfeld

Jörg Marks, Physikalisches Institut, INF 226
marks@physi.uni-heidelberg.de

■ Programm Überblick

- ✘ Linux Einführung und Arbeitsumgebung
- ✘ Installation und Überblick über wichtige/nützliche Programme
- ✘ Mikroeinführung in die Programmiersprache C/C++
- ✘ Installation und Funktion des Analysewerkzeugs ROOT
- ✘ Mikroeinführung in die Verwendung von GPUs
- ✘ Virtualisierung

■ Organisatorisches

- ✘ 2 Leistungspunkte:
 - Anwesenheitspflicht mit Übungen im Tutorialmodus
- ✘ Kurs web page

http://www.physi.uni-heidelberg.de/~marks/linux_einfuehrung/

Informationen zur Veranstaltung (1)

■ Ziel

- Installation und Nutzung eines Computers mit LINUX Betriebssystem zum Arbeiten im wissenschaftlichen Umfeld
 - ◆ Installationswerkzeuge, Linux Konfiguration (Systemadministration)
 - ◆ Shell Kommandos und shell Skripte
 - ◆ Kennenlernen wichtiger Werkzeuge und Dienstprogramme
- Mikroeinführung in die C/C++ Programmierung mit dem Ziel
 - ◆ C/C++ Quellcode bildet die Grundlage für UNIX/LINUX-artige Betriebssysteme.
 - ◆ Kennenlernen von Grundbegriffen des Sprachraums um Dienstprogramme kompilieren zu können.
- Installation und Einführung in das Datenanalysewerkzeug ROOT
 - ◆ Input / Output von Messungen und Resultaten
 - ◆ Nutzung vorhandener Funktionen zur graphischen Darstellung von Messungen
 - ◆ Datenauswertung
- Mikroeinführung in Nutzung von GPUs
 - ◆ Beispielhaft Erläutern von C Quellcode zum Rechnen auf der Laptop GPU
- Virtualisierung
 - ◆ Installation und Funktion von Virtual Box und Docker Container Software
- Beispielorientiert Konzepte so erläutern, dass Sie mit den Erklärungen selbständig systemrelevante Aufgaben lösen können.
 - ◆ Kurs in Stil eines Tutorials
 - ◆ Grundlagen für problemorientierte Lösungen und für ein Selbststudium schaffen

Informationen zur Veranstaltung (2)

■ Voraussetzungen

- **Keine Vorkenntnisse** notwendig, aber hohe Informationsdichte und Tempo
- User ID zur Benutzung der CIP Pools der Fakultät für Physik

■ Struktur des Kurses

- Wechsel zwischen Vorlesung und Übungen am Laptop
- Wechsel zwischen selbstständigem Üben und Übungen in Kleingruppen
- Kurszeiten: 8.-12.4. 9:00-12:30 und Mo-Do 14:00 -17:00
in INF 226 1. Stock Raum 204
- Kurs Web Page:
http://www.physi.uni-heidelberg.de/~marks/linux_einfuehrung/
 - Vorlesungstransparente und Text Files mit Notizen
 - Beispiel Code
 - Übungsaufgaben
 - Lösungsvorschläge

■ Voraussetzungen für einen Leistungsnachweis (2 LP)

- Anwesenheitsliste!
- Aktive Mitarbeit und Übungen in Tutorial Form

1. Teil des Programms

■ Einführung LINUX

- Historisches und Überblick zum Linux Kernel und Betriebssystem
- Installation von OpenSuSE auf dem zur Verfügung gestellten Laptop
- Betriebssystem Anpassungen von OpenSuSE
- KDE Benutzeroberfläche, Programmwerkzeuge, Editor, Shell Commands
- Anpassungen der Anwendungsprogramme zur Textverarbeitung
- Remote Zugang zum CIP Pool

■ Web - Informationen und Tutorials

- Tutorial Linux Befehle
<http://www.ee.surrey.ac.uk/Teaching/Unix/>
- Tutorial Shell Skripting von Rheinwerk Computing
http://book.rheinwerk-verlag.de/shell_programmierung/shell_001_000.htm#Xxx999207
- Tutorial awk
<https://www.tutorialspoint.com/awk/index.htm>

Linux Einleitung



- Als **Linux** bezeichnet man freie Mehrbenutzer Betriebssysteme, die auf einem Linux-Kernel basieren. Frei bedeutet hier, das der Linux-Kernel unter einer GNU General Public License steht, die erlaubt die Software auszuführen, zu studieren, zu ändern und zu verbreiten (kopieren).
- **Betriebssysteme** bestehen aus einer Ansammlung von Computerprogrammen die die Systemressourcen eines Computers wie Prozessoren, Arbeitsspeicher, Festplatten und Peripheriegeräte anderen Computerprogrammen (Anwendungsprogramme) über standardisierte Schnittstellen zur Verfügung stellen.
- **Computerprogramm**
Computer prozessieren Daten, führen Berechnungen durch, fällen Entscheidungen mit Hilfe eines Satzes von Anweisungen.
Eine Aktion wird mit Hilfe von Schlüsselwörtern beschrieben, die durch eine Programmiersprache, z. B. C/C++ oder Python, charakterisiert werden.
Die Schlüsselwörter werden im Klartext in einer Datei gespeichert und von einem Übersetzungsprogramm (Compiler) in Maschinensprache umgewandelt

Historisches



- Mitte der sechziger Jahr wurde bei AT&T an Mehrbenutzer-Betriebssystemen gearbeitet.
- Nach dem Rückzug von AT&T aus dem Großprojekt Multics entwickelten Thompsen und Ritchie Ende der sechziger Jahre eine erste in Assembler geschriebene Version, UNIX, auf einer PDP 7.
- Anfang der siebziger Jahre erfolgte eine erste Implementierung des Betriebssystems UNIX in der von Ritchie entwickelten prozeduralen Programmiersprache C. Gleichzeitig wurde ein erstes Netzwerk zwischen grossen US Universitäten aufgebaut, das eng an die Unix Entwicklung gekoppelt war. UNIX wurde gegen eine Gebühr für die Tapes frei zur Verfügung gestellt.
- Alle heute verwendeten Betriebssysteme (Windows, Mac OSX, Linux, ... und Android) entwickelten sich letzten Endes aus diesen ersten Ansätzen
 - diese Entwicklung stellt einen Schlüsselbaustein unserer heutigen Computing Technologie dar.

Unix/Linux Einführung



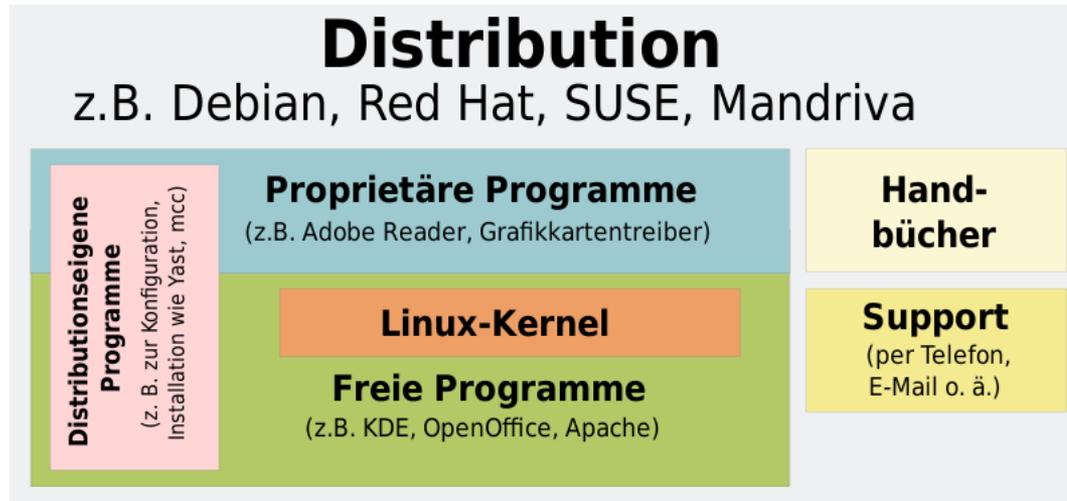
"...the number of UNIX installations has grown to 10, with more expected..."

- *Dennis Ritchie and Ken Thompson, June 1972 (Bell Lab.)*

- 1973 release von Unix V4, die in die Programmiersprache C portierte Version von Unix (frei verfügbar).
- 1987 schrieb Andrew S. Tanenbaum Minix als Lehrbetriebssystem, weil der Unix source code nicht mehr frei verfügbar war.
- 1991 veröffentlichte Linus Torvalds unter der GNU General Public License (GPL) den ersten Linux Kernel für die x86 Architektur.
- **Linux wird synonym für Linux Distributionen verwendet.** Diese verwenden alle den Linux Kernel, der auf Grund des Lizenzierungsmodells von einer Vielzahl von Entwicklern in internationaler Zusammenarbeit gewartet, verbreitet und weiterentwickelt wird. Die Linux Distributionen unterscheiden sich durch die mitgelieferten Dienstprogramme.

Unix/Linux Einführung

- **Linux Distributionen** sind Repositories von auf einander abgestimmter freier meist GPL lizensierter Software mit einem Linux-Kernel.

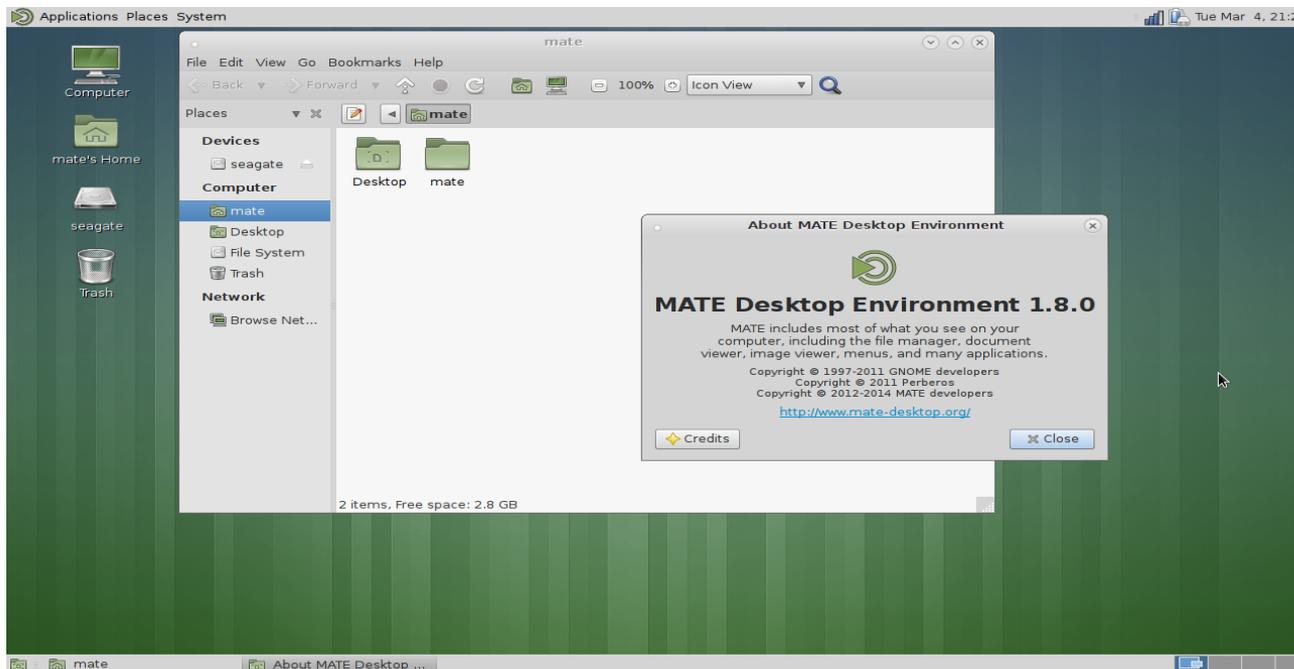


- Der Zugang und Installation der Software erfolgt über einen Paketmanager.
- Regelmäßige Releases der Distributionen und Updates
- Im Gegensatz zu Windows und MacOS enthalten die Distributionen in der Regel eine vollständige Sammlung von Dienstprogrammen.
- Distributoren sind sowohl Firmen, aber auch gemeinnützige Organisationen und Privatpersonen.
- Entwicklungslinien der verschiedene Distributionen sind hier zu finden https://upload.wikimedia.org/wikipedia/commons/1/1b/Linux_Distribution_Timeline.svg

Unix/Linux Einführung

• Linux Distributionen

- Linux Kernel (Hardware → Software Interface)
- GNU Software Umgebung enthält das Basissystem mit den zahlreichen Systemdiensten (Daemons).
- Desktop Systeme enthalten X Window Systeme, auf denen distributions-spezifische Benutzeroberflächen bereitgestellt werden.
Beispiele: KDE, Gnome, XFCE, LXDE, Unity



- Fenstermanager
- Maus-,
Tastaturanbindung
- Taskleiste
- Benachrichtigungsfeld
- Systemsteuerung
- Terminal shell
- Dateimanager
- Texteditor

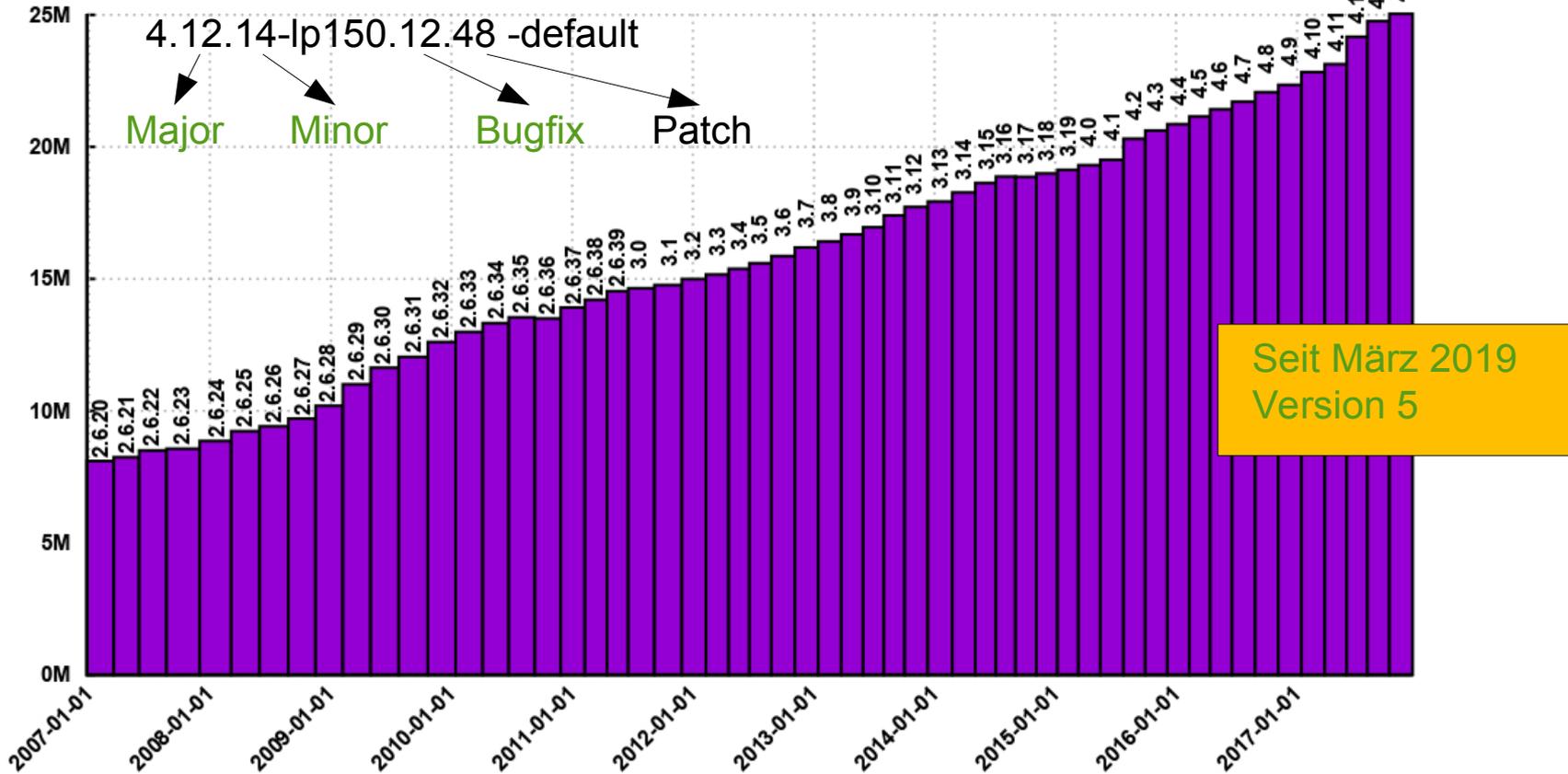
Unix/Linux Einführung

- Linux Distributionen
 - Linux Kernel (Hardware → Software Interface)
 - GNU Software Umgebung enthält das Basissystem mit den zahlreichen Systemdiensten (Daemons).
 - Desktop Systeme enthalten X Window Systeme, auf denen distributions-spezifische Benutzeroberflächen bereitstellen werden.
Beispiele: KDE, Gnome, XFCE, LXDE, Unity
 - Freie GPL Anwendungsprogramme
 - Internetbrowser
 - e-mail Klienten
 - Office Programme
 - Compiler und Interpreter
 - Textsatzprogramme
 -
 - Proprietäre Programme
 - mathematica
 - maple
 -

Linux Kernel

- Linux Kernel Distributionen

OpenSuSE 15.0 Produktionsversion
Support bis 2019



- Linux Kernel Entwicklung nicht in der Hand von Einzelpersonen, sondern in der Hand einer weltweiten Gemeinschaft. Letzte Kontrolle haben Linus Torvalds und einige Vertraute Entwickler bei der Aufnahme in die offizielle Version.

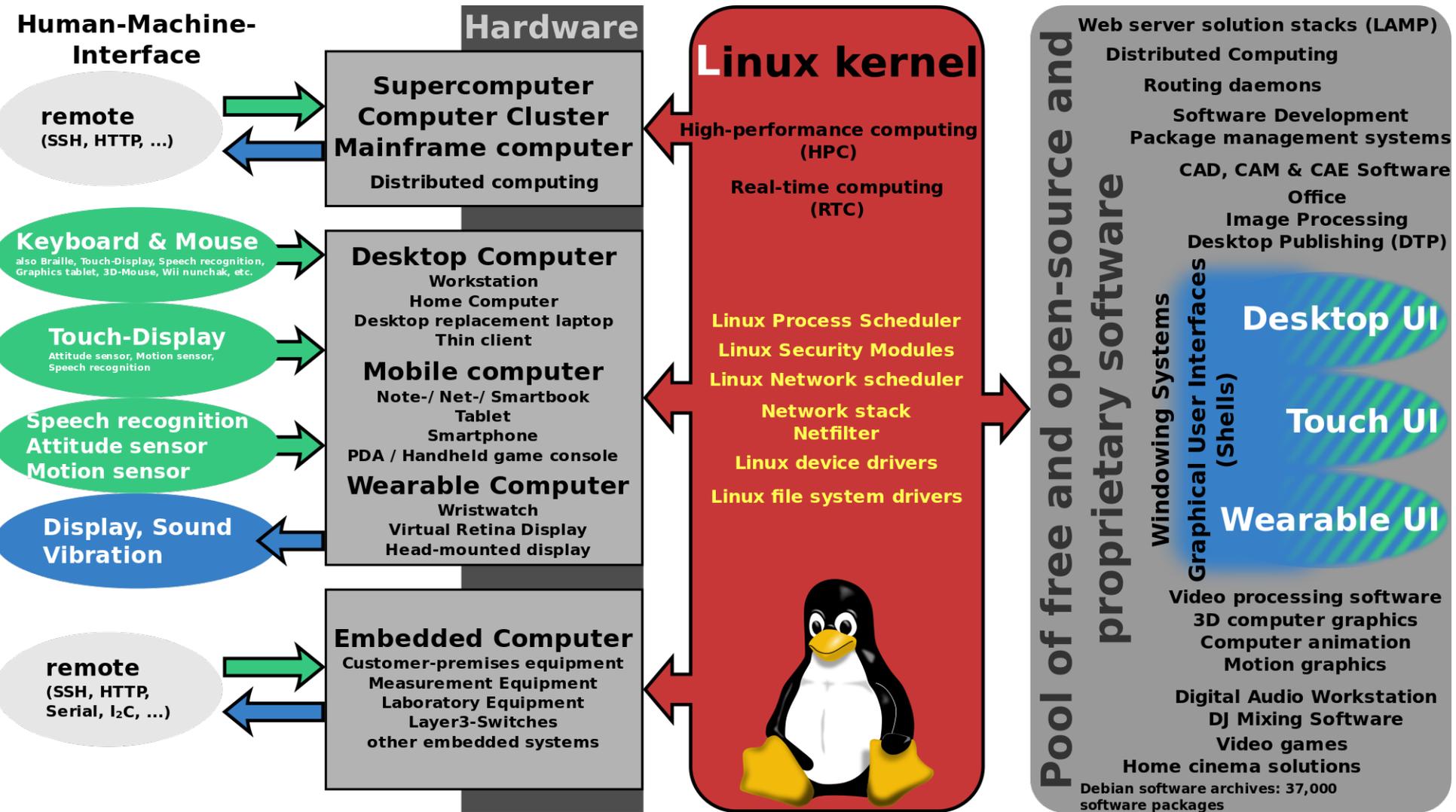
Linux Kernel

- Linux Kernel ist ein modularer monolithischer Kernel und stellt eine einheitliche Schnittstelle (API) zur Verfügung, die unabhängig von der Rechnerarchitektur ist.
- Kernel Programmiersprache ist GNU-C mit einige Assembler Anteilen
- Monolithischer Kernel mit einer Auslagerung von Funktionalität in Kernel-Module, die das Nach- oder Neuladen von Systemfunktionen (Treiber) erlauben.
- Linux selbst übernimmt das System vom BIOS-Bootloader nach den Systeminitialisierungen durch das BIOS.
- Das eigentliche Betriebssystem besteht aus dem Kernel und weiteren grundlegenden Bibliotheken und Programmen (die den Computer erst bedienbar machen)
- Linux hat sich dank des verwendeten GNU Compilers GCC zu einem weitreichend portierbarem System mit den unterschiedlichsten Anwendungsbereichen entwickelt.
- Die Kernelversion 4.12 hat 59808 Dateien mit 24.173.535 Zeilen Code und eine Größe von 99 Mbytes (July 2017, OpenSuSE 15).

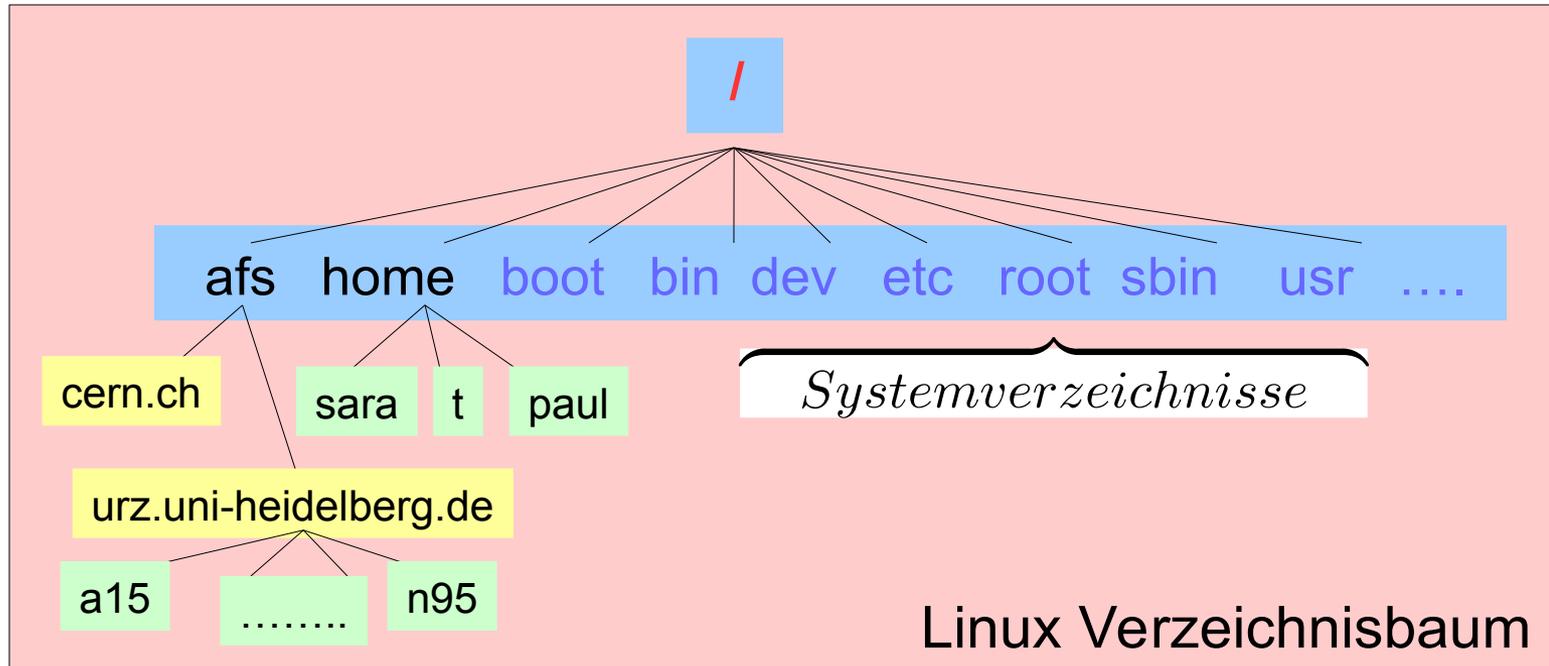


- Kernelgröße: 16 KB (1. Unix version, 1970) → 230 KB (1. Linux version, 1991) → 100 MB (Linux Kernel version, 2018)

- Linuxkernel Einsatzbereiche



Linux Verzeichnisstruktur



- Interaktion mit Userprogrammen der Linux Distributionen erfolgt über eine graphische Oberfläche oder über die shell (Eingabe Fenster).
- Für unseren Kurs brauchen wir einige shell Kommandos.

Zusammenfassung einiger shell Kommandos mit link zu einem guten Online Tutorial:

http://www.physi.uni-heidelberg.de/~marks/root_einfuehrung/Folien/UnixEinfuehrung.pdf

Mikroübersicht zum Arbeiten im CIP Pool:

http://www.physi.uni-heidelberg.de/~marks/root_einfuehrung/Folien/cipEinfuehrung.pdf