

Vorlesung

Betriebssysteme

Teil 1 Einführung

Vorlesung basiert auf Veranstaltungen und Materialien von u.a.

- Prof. Dr. Heinz-Josef Eikerling
 - Hochschule Osnabrück
- Prof. Dr. Christian Müller-Schloer
 - Leibniz Universität Hannover
- Prof. Dr. habil. Jürgen Brehm
 - Leibniz Universität Hannover
- Prof. Dr. Jörg Hähner
 - Universität Augsburg

Konstruktive Kritik ist herzlich erbeten!

Literatur

- **Andrew S. Tanenbaum: Moderne Betriebssysteme. Pearson Studium, 4., aktualisierte Auflage.**
- Mandl, Peter: Grundkurs Betriebssysteme - Architekturen, Betriebsmittelverwaltung, Synchronisation, Prozesskommunikation, 5. Aufl. 2013
- William Stallings: Operating Systems - Internals and Design Principles, 6th Ed., Pearson, 2007
- M. Garrelts: Bash Guide for Beginners
<http://www.tldp.org/LDP/Bash-Beginners-Guide/html/Bash-Beginners-Guide.html>
- S. Grägert: POSIX-Programmierung mit UNIX <http://www.graegert.de>

Organisatorisches

- 2h Vorlesung + 2h Praktikum
- Vorlesung:
 - Vorbereitung zum Praktikum
 - Übungsaufgaben zur Klausurvorbereitung
- Prüfung:
 - Klausur
- Planung Praktikum:
 - Anwesenheit = (Übungsblatt vorlegen + Lösungsversuche zum vorherigen Aufgabenblatt)
 - maximal drei Fehltermine
 - Praktikum in D327
 - **Erster Termin**
 - **Gruppe 1 und 4: 20.04.2018**
 - **Gruppe 2 und 3: 23.04.2018**

Hinweise

- Vorgehen:
 - Präsentation der Inhalte über Folien
 - Folien, Aufgaben, weitere Unterlagen werden per Ilias bereit gestellt (PW gates)
 - Testfragen und Übungen während der Vorlesung
- Praktikum:
 - Aufgaben werden in Eigenarbeit in Gruppen gelöst
 - Abgabe im Praktikumstermin (keine online-Abgabe)
 - Klärung von Fragen und Bewertung im Praktikum
 - Probleme sofort melden

Motivation & Ziele

Motivation:

- Software für Rechensysteme wird unterteilt in zwei Gruppen:
 - **Systemprogramme** zum Betrieb des Computers.
 - **Anwenderprogramme** erfüllen die Anforderungen der Anwender.
- Das Betriebssystem (*operating system*) ist das **wichtigste Systemprogramm**.

Ziele der Veranstaltung: Vermittlung von

- **Wissen:**
 - Aufbau von Betriebssystemen
 - Grundlegende Konzepte (u.a. nebenläufige Prozesses, Multi-Threading,...)
- **Praktischen Kenntnissen:**
 - Analyse und Konzeption von Betriebssystemkomponenten
 - Realisierung in **C**
 - Programmierung in der Shell
 - **Keine** Klickanleitung „wie installiere ich Windows“

Grundvoraussetzungen

Grundlegende Kenntnisse:

- Aufbau von Rechensystemen (CPU, Speicher, E/A,...)
 - Was ist ein Register? ALU? MMU?

Programmierkenntnisse in C:

- Grundlegende Konzepte:
 - Datentypen, Kontrollstrukturen, Schleifen, Funktionen, Zeiger / Speicherverwaltung
 - Entwicklungsprozess (edit, compile, link, execute)
- Programmierung in der Unix (Linux) Umgebung
 - gcc, make, ...
 - Beispiel: Ubuntu 16.04, ab 26.04.2018 Ubuntu 18.04
- Binärsystem
 - Bits und Bytes

Aufbau der Veranstaltung

- Einführung
 - Aufgabe eines Betriebssystems
 - Aufbau von Betriebssystemen
 - Programmierung in C
- Prozesse
 - Prozesse und Threads
 - Prozessstatus
 - Scheduling
- Prozesskommunikation
 - Kommunikation
 - Synchronisierung
- Verklemmungen (Deadlocks)
 - Modellierung von Verklemmungen
 - Behebung und Vermeidung von Verklemmungen

Aufbau der Veranstaltung (2)

- Shell
 - Bedienung der Shell
 - Shell-Scripting
- Speicherverwaltung
 - Partitionierung
 - Virtueller Speicher
- Dateiverwaltung
 - Dateinamen
 - Dateisysteme
- Geräteverwaltung
 - Ein/Ausgabe
 - Treiber
- Sicherheit

Einführung: Definition

Definitionen Betriebssystem:

- DIN 44300:

„Die Programme eines digitalen Rechensystems, die zusammen mit den Eigenschaften dieser Rechenanlage die Basis der möglichen Betriebsarten des digitalen Rechensystems bilden und die insbesondere die Abwicklung von Programmen steuern und überwachen.“

- A. Silberschatz:

„An operating system is similar to a government... Like a government the operating system performs no useful function by itself.“

Testfrage: Aufgaben eines Betriebssystems

- Nennen Sie einige BS-Aufgaben!

Aufgaben des Betriebssystems (1)

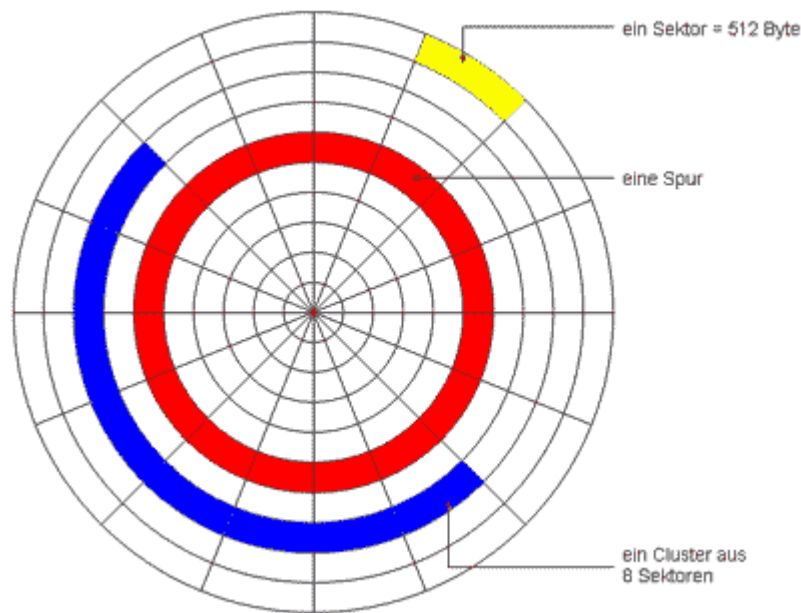
- Boot
- BS prüft **Zugriffsberechtigung** (login).
- BS stellt Verbindung zum **Benutzer** her: Kommandointerpretation, Shell, Benutzeroberfläche (Benutzer: Mensch, Programm).
- BS verwaltet **Daten** in Form von Dateien (Files): Zugriff auf Datenfiles auf Massenspeicher (Schreiben, Lesen), Kopieren, Löschen, Benennen, Ordnen. Falls Datei ein ausführbares Programm enthält: BS startet **Ausführung des Programms**.

Aufgaben des Betriebssystems (2)

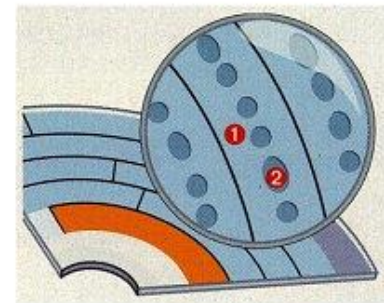
- BS verwaltet die **Ressourcen**: CPU, Speicherplatz, FD, HD, Netzausgang, Drucker, aber auch SW-Ressourcen: Programme, Prozesse, Tabellen.
- BS verbirgt Komplexität der HW-Maschine durch Vortäuschung einer komfortableren Maschine (**Abstraktion, Virtualisierung**).

Aufgaben des Betriebssystems (3)

- BS vereinfacht den **Gerätezugriff**.
- Beispiel: Massenspeicher



DVD-Recording



Die spiralförmige Spur trägt die Daten in einer Folge von reflektierenden und nicht reflektierenden Stellen (Land 1 / Pit 2).

Sie entsprechen den digitalen Werten 0 und 1.

DVD-Video (Standard)

Einführung: Aufgaben

Aufgaben eines Betriebssystems:

- **Verwaltung** der Betriebsmittel (Ressourcen)
 - *Prozessverwaltung* (erzeugen, löschen, zuteilen, synchronisieren)
 - *Speicherverwaltung*
 - Verwaltung des *Dateisystems*
 - Verwaltung von *Geräten*
 - Verwaltung der *Benutzer*
- **Abstraktion** von der Hardware
 - Die Eigenschaften der Hardware werden vor dem Benutzer verborgen.
 - Die Benutzung der Hardware wird durch eine einheitliche Schnittstelle gewährleistet.
 - Die Hardware stellt zusammen mit dem Betriebssystem eine *abstrakte Maschine* dar, auf der die Benutzerprogramme aufsetzen.

Fingerübung: Linuxkonsole



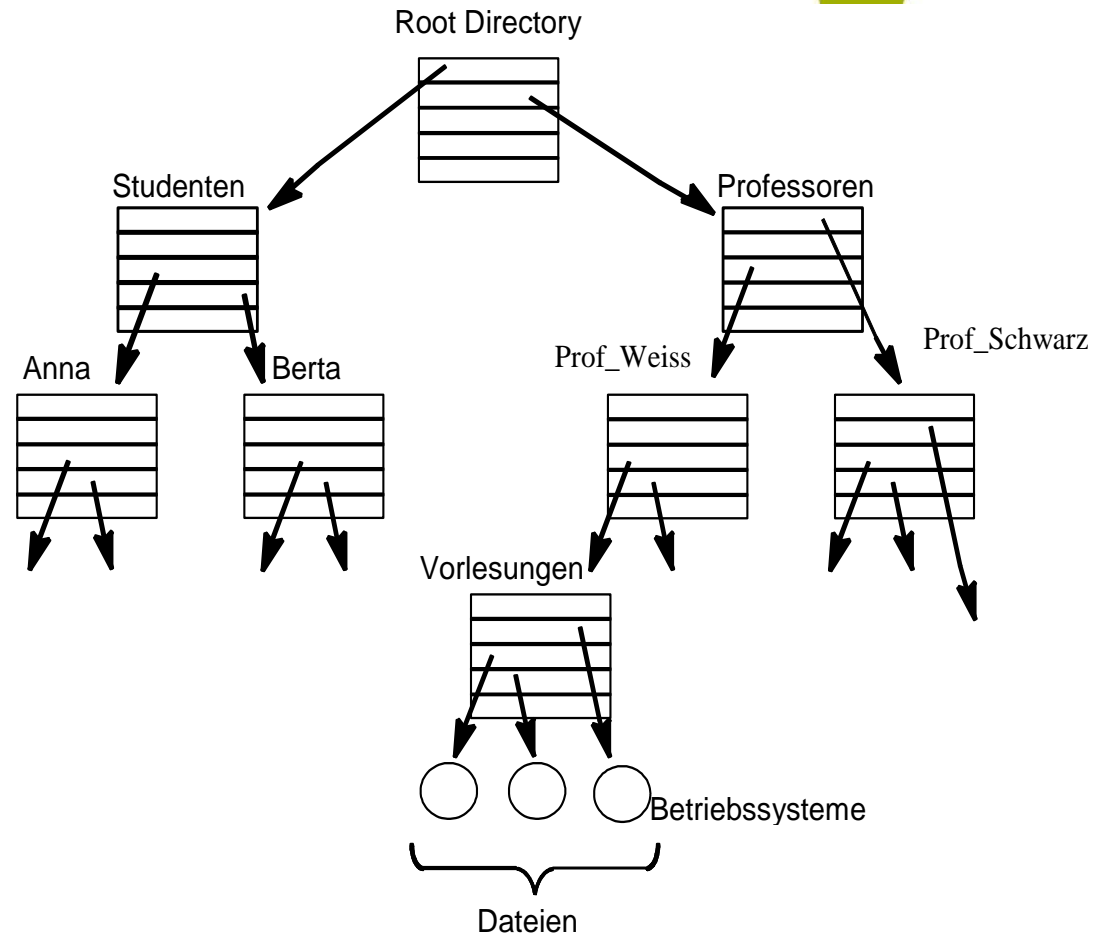
I ❤️ #!/bin/bash

Dateien

- Eine UNIX-Datei ist eine Zeichenfolge (**Bytefolge**).
- Strukturierung durch den Benutzer
- 4 Dateiartern:
 - normale Dateien: Programme, Texte
 - Verzeichnisdateien (Directories): enthalten Verweise auf Dateien und weitere Verzeichnisdateien
 - Gerätedateien (special files)
 - Pipes

Verzeichnisse

- Dateien werden in Verzeichnissen (directories) abgelegt. Directories können weitere Directories enthalten: **Hierarchie**
- Das „höchste“ Directory heißt **Root Directory**.



File-System (File-Baum)

Adressierung von Dateien

- Durch Angabe ihres **Pfadnamens** (path name), d. h. alle Directories, ausgehend von root, die bis zu der gesuchten Datei durchlaufen werden müssen.
- z.B. **/Professoren/Prof_Weiss/Vorlesungen/Betriebssysteme**
- **Absolute Pfadnamen** beginnen mit "/", also bei root.
- Genau ein Directory ist jeweils als "**Working Directory**" definiert.
- **Relative Pfadnamen** beginnen im Working Directory.
z. B. Working Directory = **/Professoren**:
rel. Pfadname: **Prof_Weiss/Vorlesungen/Betriebssysteme**

Testfrage

- Pfadnamen: Das Working Directory sei auf /homes/schroeder/fischer eingestellt. Wohin weist der Pfadname „/homes/schulz/merkel?“
- Mögliche Antworten:

/homes/schroeder/fischer

/homes/schroeder/fischer/schulz/merkel

/homes/schulz/merkel

/homes/schulz/merkel/schroeder/fischer

Schutzmechanismen

- Jede Datei, jedes Directory erhält einen 9-bit-Code (Schutzbits, Protection Code):



R (read): lesen
 W (write): schreiben
 X (execute): ausführen (für Verzeichnis: suchen)

- Eine „1“ bedeutet: das entsprechende Recht wird gegeben.
- Beispiel: 111 101 001 heißt: **rwX r-X --X**
Eigentümer darf lesen, schreiben, ausführen
Gruppe darf lesen und ausführen
alle anderen dürfen nur ausführen.
- Überprüfung der Zugriffsberechtigung beim **Öffnen** der Datei.

Testfrage

- Eine Datei habe die Schutzbits "111 101 001". Sie sind nicht der Eigentümer, gehören aber zur Gruppe. Dürfen Sie schreiben?
- Mögliche Antwort:
 - ja
 - nein

cd, pwd

- Wie kommt man mit dem cd-Kommando ...
 - in sein home-Verzeichnis?
 - in das übergeordnete Verzeichnis?
 - in das root-Verzeichnis?
- Wie kann das aktuelle Arbeitsverzeichnis angezeigt werden?

Lösung zu cd, pwd

- `cd`
- `cd ..`
- `cd /`

- `pwd`

man

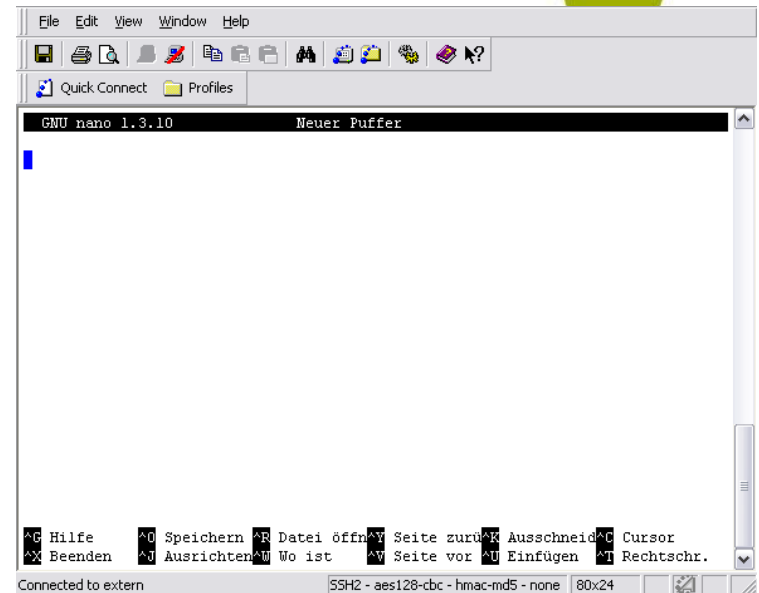
- Besorgen Sie sich Informationen zum man-Kommando.
 - Welches Programm wird zum Anzeigen der man-Pages verwendet (Pager)?
 - Wie kann man auf einer man-Page
 - rückwärts blättern?
 - nach einem Begriff suchen?
 - zur nächsten Fundstelle springen?
 - Wie verlässt man die man-Page?
 - Wie können Sie das man-Kommando verwenden, um herauszufinden, welche Kommandos das UNIX System für die Uhrzeit vorsieht?

Lösung man

- **man man**
 - liefert Informationen zum man-Kommando
- **Pager: less;**
 - Hilfe dazu über h
- **CTRL-B**
 - zurückblättern
- **/<Suchbegriff>**
 - sucht in der man-Page nach dem Suchbegriff
 - n Springt zur nächsten Fundstelle
-
- **q** verlässt die man-Page
-
- **man -k <Schlüsselwort>**
 - liefert Informationen zu einem gegebenen Schlüsselwort
 - z.B. man -k time
liefert Informationen zu Funktionen für die Zeitmanipulation.

Texteditor

- Editor "nano" in der Shell
 - Sonderbefehle
 - Beenden des Editors mit Nachfrage zum Speichern
 - STRG-X
 - Speichern
 - STRG-O
 - Laden
 - STRG-R
 - Suchen
 - STRG-W
 - usw.



ls

- Das Kommando ls hat verschiedene Optionen. Benutzen Sie das Kommando man, um festzustellen, mit Hilfe welcher Option das Folgende am Bildschirm ausgegeben werden kann
 - alle, auch die mit . beginnenden Dateien eines Verzeichnisses
 - Dateien eines Verzeichnisses in unsortierter Reihenfolge (in der die Dateien auf der Festplatte gespeichert sind)
 - Langformat Auflistung mit Dateityp, Zugriffsrechten, Anzahl von Hardlinks, Besitzername, Dateigröße u.s.w.
 - die horizontale Auflistung von mit Komma getrennten Dateien.

Lösung zu ls

- **-a**
 - do not ignore entries starting with .
- **-f**
 - do not sort, enable -aU, disable -lst
- **-l**
 - use a long listing format
- **-m**
 - fill width with a comma separated list of entries
- **-u**
 - with -lt: sort by, and show, access time
 - with -l: show access time and sort by name
 - otherwise: sort by access time

mkdir, cp, mv, rm, rmdir

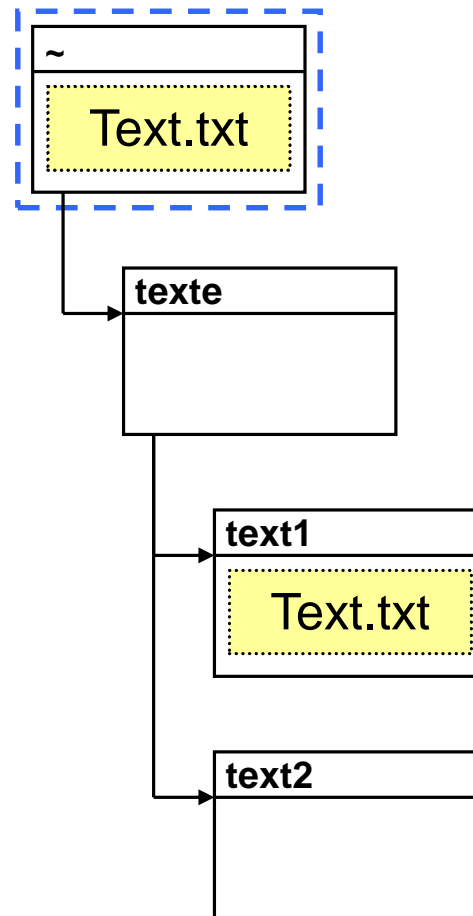
- Erstellen Sie das Verzeichnis `texte` in Ihrem Home-Verzeichnis.
 - Erstellen sie mittels `nano` eine Datei `Text.txt` mit beliebigem Inhalt
 - Kopieren Sie sich die Datei `Text.txt` ins Verzeichnis `texte`.
- Erstellen Sie das Unterverzeichnis `text1` im Verzeichnis `texte`.
 - Verschieben Sie die Datei `Text.txt` ins Verzeichnis `text1`.
- Erstellen Sie das Unterverzeichnis `text2` im Verzeichnis `texte`.
 - Kopieren Sie die Datei `Text.txt` aus dem Verzeichnis `text1` ins Verzeichnis `Text2`.
- Löschen Sie die Datei `Text.txt` aus dem Verzeichnis `text1`.
- Löschen Sie das Verzeichnis `text1`.

Lösung 1 zu mkdir, cp, mv, rm, ...

- `mkdir texte`
- `nano Text.txt`
- `cp Text.txt texte`
- `cd texte`
- `mkdir text1`
- `mv Text.txt text1`
- `mkdir text2`
- `cp text1/Text.txt text2`
- `cd text1`
- `rm Text.txt`
- `cd ..`
- `rmdir text1`

Lösung 2 zu mkdir, cp, mv, rm, ...

- `mkdir texte`
- `nano Text.txt`
- `cp Text.txt texte`
- `cd texte`
- `mkdir text1`
- `mv Text.txt text1`
- `mkdir text2`
- `cp text1/Text.txt text2`
- `cd text1`
- `rm Text.txt`
- `cd ..`
- `rmdir text1`



Ausgabeumlenkung cat, more, sort

- Lenken Sie die Ausgabe des Kommandos `ls -f /etc` in die Datei `lsf.txt` um.
 - Geben Sie die Datei `lsf.txt` am Bildschirm aus.
 - Geben Sie die Datei `lsf.txt` seitenweise am Bildschirm aus.
 - Geben Sie die sortierte Datei `lsf.txt` am Bildschirm aus
 - Alphabetisch
 - Umgedreht alphabetisch
 - Sortiert sowie seitenweise

Lösung zu Ausgabeumlenkung cat, more, sort

- `ls -f /etc > lsf.txt`
- `cat lsf.txt`
- `more lsf.txt`
- `cat lsf.txt | sort | more`
- `cat lst.txt | sort -r | more`
- `sort lsf.txt | more`

**Vielen Dank für Ihre
Aufmerksamkeit**