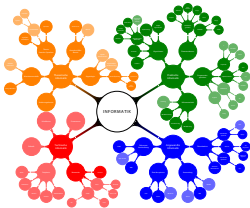


Kapitel 3

Betriebssysteme

Von Äpfeln und Pinguinen

Vorlesung Einführung in die Informatik 1 vom 31. Oktober 2013 von [Till Tantau](#)



Lernziele von Kapitel 3

1. Aufgaben von Betriebssystemen kennen
2. Konzepte der Datei-, Benutzer- und Prozessverwaltung kennen
3. Mit Dateien, Pfaden und Dateirechten umgehen können
4. Aufgaben von Treibern erläutern können

Gliederung von Kapitel 3

Das Betriebssystem ist ein großes Programm, das unter anderem folgende Dinge leistet:

- ▶ *Druckauftragsverwaltung*
- ▶ *Dateiverwaltung*
- ▶ *Nutzerverwaltung*
- ▶ *Prozessverwaltung*

Zur Diskussion

Um welche Ressourcen könnte sich ein Betriebssystem noch kümmern?

Man kann Betriebssysteme unterscheiden nach:

- ▶ Funktionsumfang
 - ▶ Vom Betriebssystem eines Fahrradcomputers . . .
 - ▶ . . . über Windows und Linux . . .
 - ▶ . . . bis zum Betriebssystem einer Mainframe.
- ▶ Abstammung
 - ▶ Unix-Familie: Linux, MacOS X, Solaris, AIX, . . .
 - ▶ MS-DOS-Familie: MS-DOS, DR-DOS, Windows
 - ▶ Eigenentwicklungen: PalmOS, BeOS, OS/2, . . .

- 1981 IBMs erste »Personal Computer« werden mit dem »Microsoft Disk Operating System« (*MS-DOS*) ausgeliefert, da ein eigenes nicht zur Verfügung steht.
- 1986 Microsofts *Windows* wird vorgestellt. Es soll den Umstieg von MS-DOS auf OS/2 erleichtern.
- 1987 IBMs eigenes Betriebssystem *OS/2* wird vorgestellt.
- 1991 Microsoft treibt Windows voran, welches sich gegen OS/2 durchsetzt.

- ab 1960 Verschiedene Firmen stellen Varianten des Betriebssystems *Unix* her.
- 1983 Das *GNU-Projekt* (»Gnu is Not Unix«) macht es sich zur Aufgabe, ein *freies* Unix zu schaffen.
- 1991 *Linus Torvald* startet ein Projekt, das GNU-Unix auf PCs mit Intel-Prozessoren zu portieren.
- heute *Linux* ist das größte und verbreitetste freie Betriebssystem.

Ressource I: Die Verwaltung von Druckaufträgen ist scheinbar einfach.

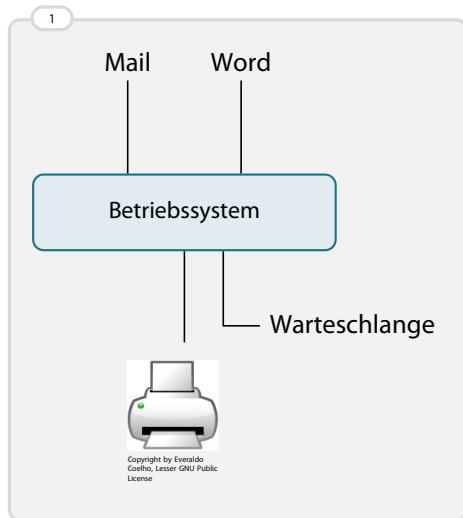
Problemstellung

- ▶ Anwendungsprogramme sollen Texte und Graphiken drucken können.
- ▶ Jedoch können mehrere Anwendungsprogramme gleichzeitig versuchen, etwas zu drucken.

Lösung

- ▶ Anstatt Daten direkt an die Drucker zu schicken, erstellen Anwendungsprogramme *Druckaufträge*,
- ▶ die sie an das Betriebssystem schicken.
- ▶ Dieses reiht sie in eine *Warteschlange* ein und schickt sie dann in der Reihenfolge ihres Eintreffens an den Drucker.

Beispiel, wie die Druckauftragsverwaltung arbeitet.



Beispiel, wie die Druckauftragsverwaltung arbeitet.

2

3-9

Mail



Word



Betriebssystem

Warteschlange



Copyright by Everaldo
Coelho, Lesser GNU Public
License

Beispiel, wie die Druckauftragsverwaltung arbeitet.

3

3-9

Mail

Word

Betriebssystem

Warteschlange



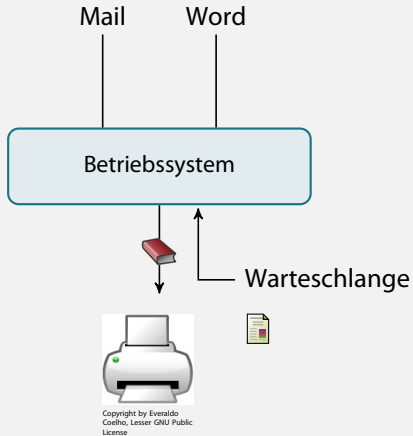
Copyright by Everaldo
Coelho, Lesser GNU Public
License



Beispiel, wie die Druckauftragsverwaltung arbeitet.

4

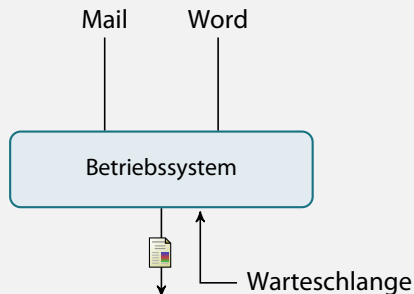
3-9



Beispiel, wie die Druckauftragsverwaltung arbeitet.

5

3-9



Copyright by Everaldo
Coelho, Lesser GNU Public
License

Problemstellung

3-10

- ▶ Die *Daten* der Benutzer müssen verwaltet werden.
- ▶ Daten schließt *Texte*, *Graphiken*, *Filme*, *Programme* und viele Dinge mehr ein,
- ▶ die *sehr unterschiedliche Größe* haben.
- ▶ Weiterhin soll der Zugriff *schnell* sein.

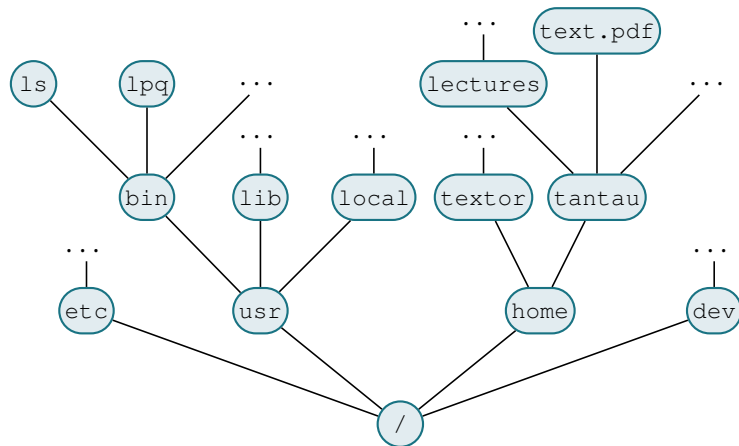
Lösung

- ▶ Alle Daten werden als *Folgen von Bytes* gespeichert.
- ▶ Jede Einheit von Daten wie ein Dokument oder ein Programm oder ein Film bildet eine *Datei* (File) und bekommt einen *Namen*.
- ▶ Zur größeren Übersichtlichkeit werden diese Dateien in einem *Verzeichnisbaum* angeordnet.

- ▶ Dateien sind Folgen von Bytes.
- ▶ Ihre *Länge* oder *Größe* wird in Byte gemessen, wobei die kürzesten Dateien Null Bytes haben, die längsten mehrere Gigabyte.
- ▶ Ihr *Name* ist eine vom Inhalt unabhängige Zeichenfolge, die am besten keine Sonderzeichen enthält.
Der Name besteht oft aus zwei durch einen Punkt getrennte Teile, wobei der zweite Teil (*Suffix* oder *Dateierweiterung* genannt) angibt, um welche Art Datei es sich handelt.

- ▶ Verzeichnisse enthalten Dateien und wieder Verzeichnisse (Unterverzeichnisse).
- ▶ Genau wie Dateien haben Verzeichnisse einen Namen.
- ▶ Das oberste Verzeichnis nennt man das *Wurzelverzeichnis* (root directory).

Ein typischer Unixverzeichnisbaum.



- ▶ Die Folge von (Unter-)Verzeichnisnamen, die zu einer Datei führt, bezeichnet man als *absoluten Pfad*.
- ▶ Innerhalb des Pfades werden die verschiedenen Verzeichnisnamen durch Schrägstriche getrennt (Unix) oder durch umgedrehte Schrägstriche (Windows).

Beispiel: `/home/tantau/text.pdf`

Beispiel: `\windows\system32\windows.exe`

- ▶ Kommt innerhalb eines Pfades » . . « vor, so ist damit das Verzeichnis eins höher gemeint.

Zur Übung

Welche Datei bezeichnet der Pfad

`/usr/../../home/tantau/../../dev/null?`

- ▶ Ein *relativer Pfad* setzt einen Pfad relativ zu einem *aktuellen Verzeichnis* fort.
- ▶ Auf das aktuelle Verzeichnis kann man sich mittels `»` `.` `«` beziehen.
- ▶ **Beispiel:** Wir befinden uns im Verzeichnis `/home`. Dann ist `./tantau/text.pdf` die Datei `/home/tantau/text.pdf`.
- ▶ **Beispiel:** Wir befinden uns im Verzeichnis `/home/textor`. Dann ist ebenfalls die Datei `../tantau/text.pdf` dieselbe Datei wie oben.

Zur Übung

Sie befinden sich im Verzeichnis `/usr/local/bin`. Welche Datei bezeichnet dann `../../local../bin/lpq?`

Problemstellung

- ▶ Die Daten von vielen Personen (Benutzern) sollen auf einem Computer gespeichert werden.
- ▶ Dabei soll es Benutzern gestattet sein, manche Daten anderer Benutzer zu lesen, andere aber wieder nicht.

Lösung

- ▶ Jeder Benutzer bekommt ein Verzeichnis, in dem seine Daten liegen.
- ▶ Für jede Datei gibt es *Rechte*, die festlegen, wer was darf.
- ▶ Für jeden Benutzer speichert das Betriebssystem ein *Password* und Nutzer müssen sich *einloggen* und damit *authentifizieren*.

Das Unix-Betriebssystem regelt zunächst den *Besitz* von Dateien:

1. Jede Datei gehört genau einem bestimmten *Nutzer*.
 - ▶ Besitzer ist, wer die Datei angelegt hat.
 - ▶ Nur der Besitzer kann die Rechte ändern.
2. Jede Datei gehört genau einer bestimmten *Gruppe*.
 - ▶ Eine Gruppe ist eine Menge von Benutzern,
 - ▶ wobei Benutzer aber Mitglieder mehrerer Gruppen sein können.

Genau wie Dateien werden Gruppen auch vom Betriebssystem verwaltet.

Das Unix-Betriebssystem kennt drei Arten von Rechten:

1. *Leserecht (r-Recht)*

Das Recht, den Inhalt einer Datei / eines Verzeichnis zu lesen.

2. *Schreibrecht (w-Recht)*

Das Recht, den Inhalt einer Datei / eines Verzeichnis zu ändern.

3. *Ausführungsrecht (x-Recht)*

Das Recht, ein Programm auszuführen, dessen Code in der Datei liegt. Fehlt dieses Recht für ein Verzeichnis, so kann man es nicht in einem Dateipfad benutzen (»man kann nicht hineinwechseln«).

Für jede Datei wird gespeichert, welche der drei Rechte bestimmte Personengruppen haben:

1. Es wird gespeichert, welche Rechte der Besitzer hat.
2. Es wird gespeichert, welche Rechte Angehörige der Gruppe haben.
3. Es wird gespeichert, welche Rechte alle anderen Benutzer haben.

Dies führt zu einer Rechtestabelle wie im folgenden Beispiel:

	read	write	execute
user	yes	yes	yes
group	yes	no	yes
other	no	no	no

Dies schreibt man auch kurz so: `rwxr-x---`.

Zur Übung

Die Nutzer `adam` und `eva` sind beide in der Gruppe `eden`, aber nur `adam` ist in der Gruppe `men`. In einem Verzeichnis finden sich folgende Dateien:

Dateiname	Besitzer	Gruppe	Rechte
<code>apple.txt</code>	<code>eva</code>	<code>eden</code>	<code>rwxr-----</code>
<code>snake.txt</code>	<code>adam</code>	<code>eden</code>	<code>rw-rw-r--</code>
<code>eat</code>	<code>adam</code>	<code>men</code>	<code>rwxrwxr-x</code>

Welche Dateien können jeweils Adam und Eva lesen und welche schreiben?

Problemstellung

- ▶ Benutzer wollen mehrere Programme gleichzeitig ausführen.
- ▶ Falls Prozesse dabei böse Dinge versuchen, muss man sie gewaltsam stoppen können.

Lösung

- ▶ Das Betriebssystem verwaltet eine *Prozessliste*.
- ▶ Jeder Prozess hat eine *Nummer* und *gehört* einem Benutzer.
- ▶ (Nur) der Benutzer kann seine Prozesse *töten*.

Zur Diskussion

Was könnten »böse Dinge« alles sein?

Beim Drucken gibt es *zwei unterschiedliche Probleme*:

3-22

1. Die Druckaufträge müssen verwaltet werden.

Dies beinhaltet Tätigkeiten wie:

- ▶ Nummerieren der Druckaufträge,
- ▶ Verschicken von Rückmeldungen,
- ▶ Auflistung der angeschlossenen Drucker.

2. Das Übermitteln der Daten an den Drucker.

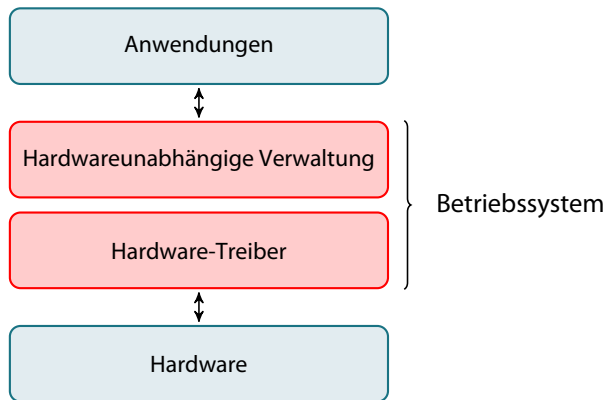
Dies beinhaltet Tätigkeiten wie:

- ▶ Umwandlung der Daten in ein Format, das der Drucker versteht (jeder Drucker spricht eine eigene »Sprache«),
- ▶ oder Senden der Daten an das richtige Kabel und den richtigen Anschluss.

Wichtige Beobachtung

Die erste Aufgabe ist *hardwareunabhängig*. Die zweite ist *druckerabhängig* und muss *für jeden Drucker anders programmiert werden*.

Ein Betriebssystem besteht aus zwei Schichten.



- ▶ Ein *Treiber* ist ein austauschbarer Teil des Betriebssystems.
- ▶ Er ist ein Programm, das für *eine bestimmte Hardware* eine *bestimmte Aufgabe* löst.
- ▶ Baut man eine neue Hardware in einen Computer ein, so muss man in der Regel einen passenden Treiber installieren.

Beispiel

Ein *Druckertreiber* kümmert sich darum,

- ▶ zu druckende Daten in ein Format zu konvertieren, das ein bestimmter Drucker versteht, oder
- ▶ den richtigen Anschluss (beispielsweise USB) anzusprechen.

Er kümmert sich *nicht* darum, Druckaufträge zu verwalten.

Benutzer können auf zwei Arten mit dem Betriebssystem kommunizieren:

1. Mittels einer *Shell*.
Dazu mehr im nächsten Kapitel.
2. Mittels eines *graphischen Tools*.
Diese findet man in graphischen Oberflächen in der Regel in einer Art »Systemmenü«.

Programme kommunizieren mit dem Betriebssystem auf zwei Arten:

1. Ebenfalls mittels einer *Shell*.
2. Mittels so genannter *Systemaufrufe*.
Hier gibt es für jeden *Verwaltungsakt* des Betriebssystems ein kleines Programm, das man aufrufen kann.

Betriebssysteme

3-26

Ein *Betriebssystem* verwaltet verschiedene Ressourcen, insbesondere

- ▶ den Hauptspeicher,
- ▶ die Festplatten,
- ▶ die Benutzerliste,
- ▶ die Prozesse (laufende Programme),
- ▶ die Bildschirme (Graphik) und
- ▶ die Peripherie (Drucker, Maus, Tastatur, Soundkarten, etc.).

Für jede Ressource enthält das Betriebssystem

- ▶ einen allgemeinen Teil, der unabhängig von der Hardware ist, und
- ▶ einen austauschbaren, hardwareabhängigen Teil, genannt *Treiber*.

Dateien

- ▶ Dateien haben einen *Name*,
- ▶ eine bestimmte *Länge*, die in Bytes gemessen wird, und
- ▶ einen *Namenssuffix*, der die »Art« der Datei angibt.

Dateibaum

- ▶ Dateien werden in einem *Baum* angeordnet.
- ▶ Ein *absoluter Pfad* in diesem Baum startet bei der »Wurzel /«.
- ▶ Ein *relativer Pfad* startet im »aktuellen Verzeichnis«.
- ▶ In einem Pfad steht . . für »einen Schritt zurück«.

Rechte

Es gibt drei Rechte:

- ▶ Das Leserecht »`r`«.
- ▶ Das Schreibrecht »`w`«.
- ▶ Das Ausführungsrecht »`x`«.

Es gibt drei Arten von Besitz:

- ▶ Man ist Besitzer der Datei (»`u`« wie *user*).
- ▶ Man gehört zur Gruppe der Datei (»`g`« wie *group*).
- ▶ Man ist jemand anderes (»`o`« wie *other*).

Man nutzt Folgen von neun Zeichen wie `rwXrw-rw-`, Rechte an einer Datei oder einem Verzeichnis zu notieren.