

# 706.088 INFORMATIK 1

## BETRIEBSSYSTEME

# WIEDERHOLUNG

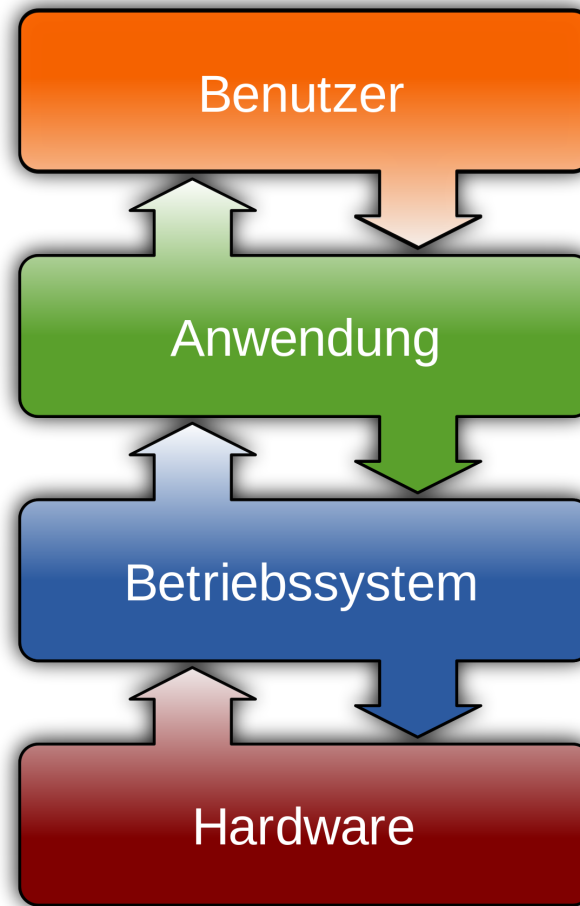
- › Analytische Problemlösung
- › Fermi Probleme
- › Algorithmus
  - ›› Türme von Hanoi (Rekursion)
- › Module
- › Klassen (Vererbung)

# BETRIEBSSYSTEME

# AUFGABEN DES BETRIEBSSYSTEMS

- › Verwaltet Ressourcen
  - ›› CPU Zeit, RAM, I/O, Prozesse
- › Verwaltet I/O
  - ›› Monitor, Tastatur, Festplatten, Netzwerk
- › Zuteilung von Prioritäten
  - ›› CPU Zeit an wichtige Programme
- › Verwaltung von Rechten
  - ›› Voneinander unabhängige Benutzer und Programme stören sich nicht.

# INTERAKTION MIT DEM BETRIEBSSYSTEM

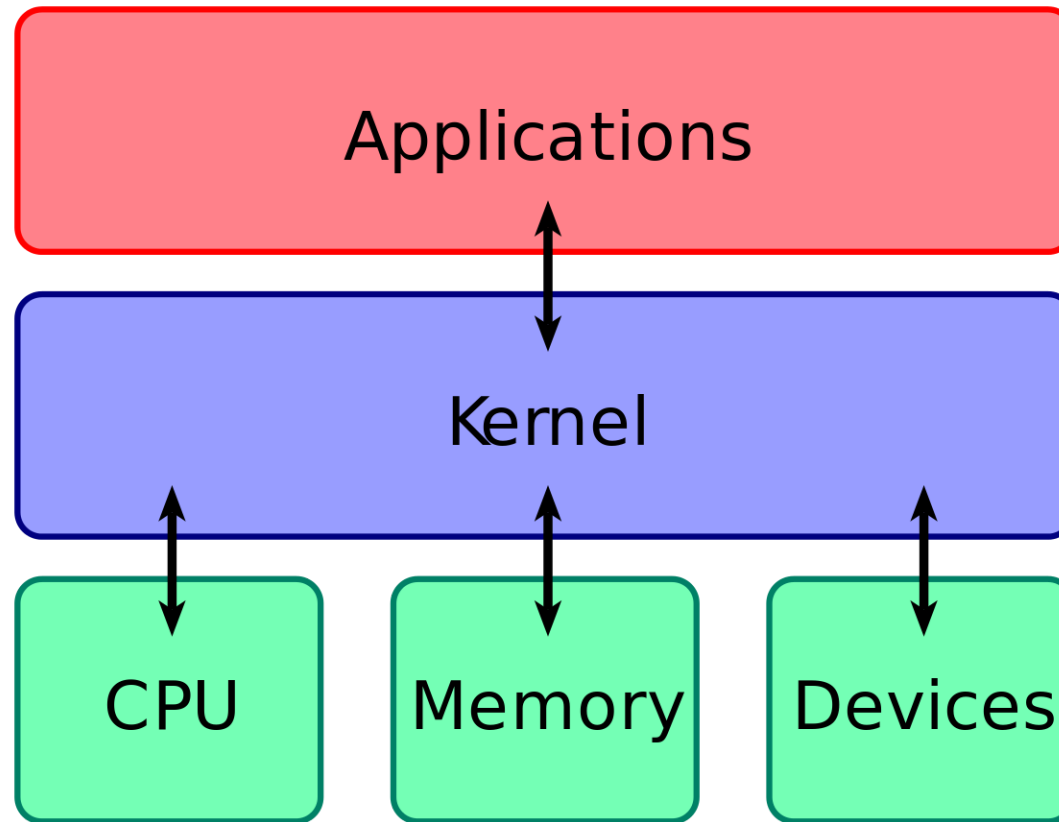


Von Golftheman [CC BY-SA 3.0](#), [Link](#)

# KOMPONENTEN EINES MODERNEN BETRIEBSSYSTEMS

- › Kernel
  - ›› Verbindet Software über Treiber und Firmware mit der Hardware
- › Prozesse
  - ›› Jedes gestartete Programm wird als eigener Prozess geführt
  - ›› Prozesse sollen unabhängig voneinander funktionieren. Stürzt Prozess A ab, soll Prozess B weiter laufen.

# KERNEL



By [Bobbio](#) - Own work, [CC BY-SA 3.0](#), [Link](#)

# KOMPONENTEN EINES MODERNEN BETRIEBSSYSTEMS

- › Speicherverwaltung
  - » Regelt welches Programm auf welchen Bereich im Speicher Zugriff hat
  - » Zugriffe eines Programms auf fremde Speicheradressen resultiert in einem Fehler
- › Multitasking
  - » Erlaubt das Ausführen von mehreren Programmen 'gleichzeitig' (eigentlich schnell hintereinander)



# KOMPONENTEN EINES MODERNEN BETRIEBSSYSTEMS

## BENUTZERSCHNITTSTELLEN

- › Grafische Benutzeroberfläche
  - » Fenstermanager
  - » "Einfachere Handhabung"
- › CLI (Command Line Interface)
  - » Konsole
  - » Server oder Remote Umgebungen

# GESCHICHTE DER BETRIEBSSYSTEME

- › bis 1945 kaum relevant
  - » Berechnungen und Software waren sehr spezialisiert
- › Zwischen 1945 und 1965
  - » Anfang 1950 konnte ein Computer nur ein Programm (gleichzeitig) ausführen
  - » 1960 erste (erschwingliche) Transistorrechner (in Rechenzentren)

# GESCHICHTE DER BETRIEBSSYSTEME

- › Von 1965 bis 1980
  - ›› Timesharing-Systeme (70er Jahre)
    - › Main-Frame mit vielen Terminals
  - ›› Prozessrechnersysteme

# MAIN-FRAME



Von Grkauls - Eigenes Werk, Gemeinfrei, [Link](#)

# MAIN-FRAME-TERMINAL



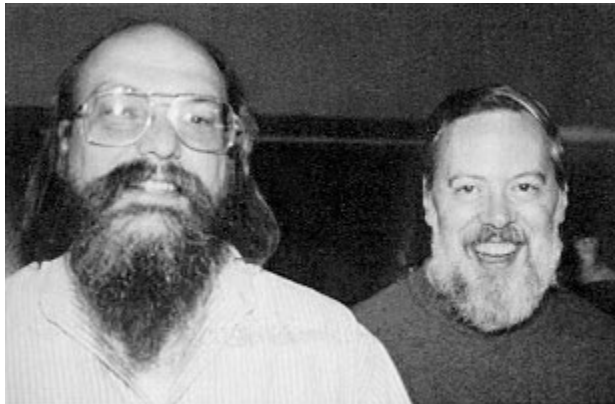
By [Jamie Cox](#) from Melbourne, USA - [Zenith Z-19 Terminal](#) Uploaded by [Mewtu](#), CC BY 2.0, [Link](#)

# GESCHICHTE DER BETRIEBSSYSTEME

- › 1980 bis heute
  - ›› Entwicklung von Mikroprozessoren
  - ›› Computer am Arbeitsplatz (Workstations)
    - › Mit grafischer Ausgabe
  - ›› Neue Anforderungen an Betriebssystem

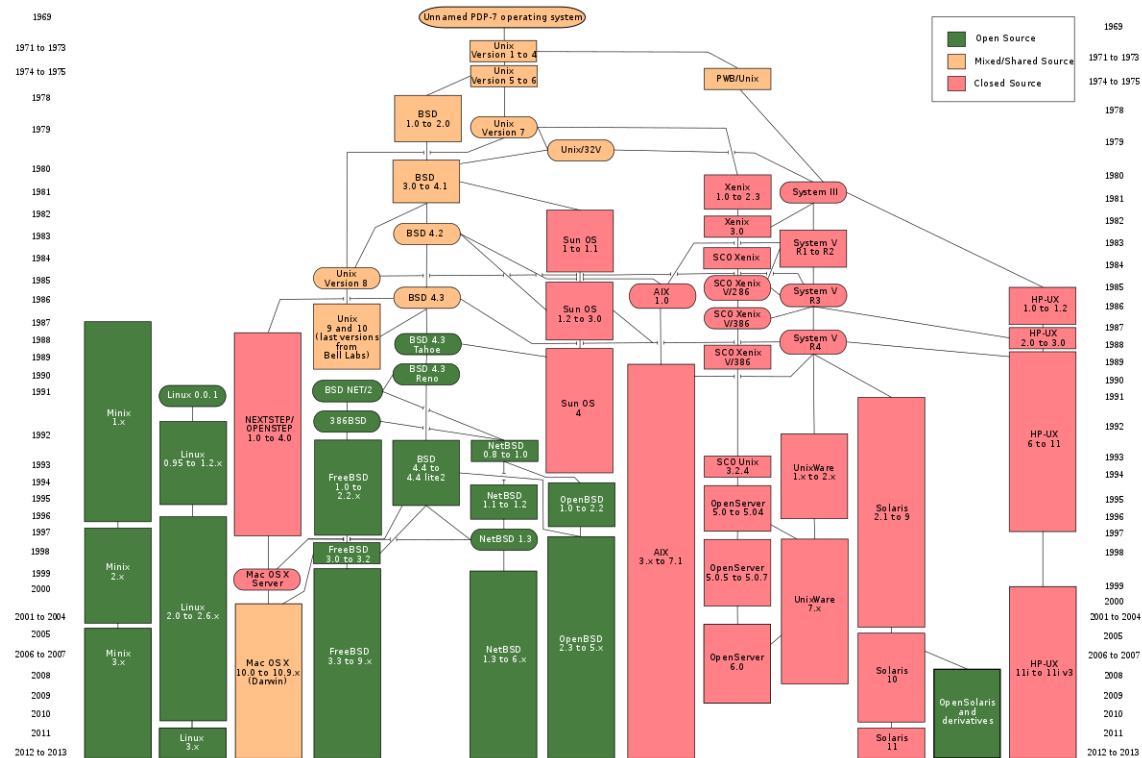
# UNIX

## UNIX' CREATORS



K. Thompson & D. Ritchie, Public Domain, [Link](#)

# UNIX EVOLUTION



By Eraserhead1, Infinity0, Sav\_vas - Levenez Unix History Diagram, Information on the history of IBM's AIX on ibm.com, CC BY-SA 3.0, Link



# GESCHICHTE VON LINUX UND BSD

- › Universität Berkley, Kalifornien, startet in den 1970ern BSD (Berkley Software Distribution)
  - ›› BSD ist ein UNIX Derivat und kann gratis im Internet bezogen werden.
  - › FreeBSD, OpenBSD, NetBSD sind freie Derivate (BSD-Lizenz)



# GESCHICHTE VON LINUX UND BSD

FreeBSD **OpenBSD**  
Idealist Richard Stallman gründet das GNU Project (Anfang der 1980er)



1: By Joseph W. Reiss for the Free Software Foundation - [Color Gnu Head](#), [FAL](#), [Link](#)

2: By Sam Williams - Taken from the cover of the O'Reilly book [w:Free as in Freedom: Richard Stallman's Crusade for Free Software](#), [CC BY-SA 3.0](#), [Link](#)

6.11

# GESCHICHTE VON LINUX UND BSD

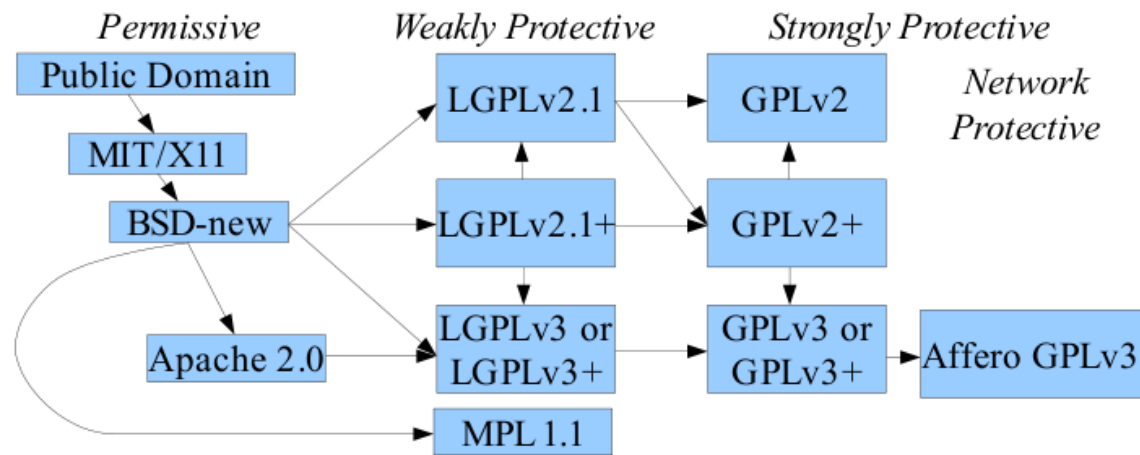
- › Komplette **freie Software** für Endanwender (versus proprietäres UNIX OS)
- › Einzelne Programme funktionieren gut, der Kernel, der das UNIX Pendant ersetzen soll, nicht.

# FREIE SOFTWARE

# 4 FREIHEITEN FREIER SOFTWARE

1. Die Freiheit, das Programm **auszuführen**, wie man möchte, für jeden Zweck.
2. Die Freiheit, die Funktionsweise des Programms zu untersuchen und eigenen Bedürfnissen der Datenverarbeitung **anzupassen**
3. Die Freiheit, das Programm **weiterzuverbreiten** und damit seinen Mitmenschen zu helfen
4. Die Freiheit, das Programm **zu verbessern** und diese Verbesserungen der Öffentlichkeit freizugeben, damit die gesamte Gemeinschaft davon profitiert

# FREIE SOFTWARE



By David A. Wheeler - <http://www.dwheeler.com/essays/floss-license-slide.html>, CC BY-SA 3.0, Link

Mehr Information zu Freier Software 

# GESCHICHTE VON LINUX UND BSD

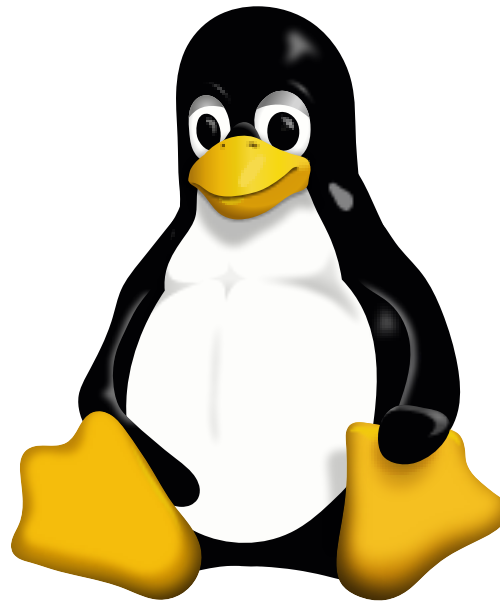
- › 1991 veröffentlicht der finnische Student Linus Torvalds die erste Version des **LINUX Kernel**



# GESCHICHTE VON LINUX UND BSD

- › Linux und GNU werden zusammengeführt und es entsteht: **GNU/Linux**, heute meist Linux genannt.

8.2



By [Larry Ewing](#), [Simon Budig](#), [Garrett LeSage](#) - [g.org/simon/penguin/">g.org/simon/penguin/](#)>[1], [garrett/Tux](#) on GitHub, CC0, [Link](#)

8.3



Wer hat bereits Linux verwendet?

A large, stylized green Android robot head logo serves as a background for the text. It has two white circular eyes and two antennae on top.

**WER HAT BEREITS LINUX VERWENDET?**

Android™ basiert auf dem Linux Kernel!

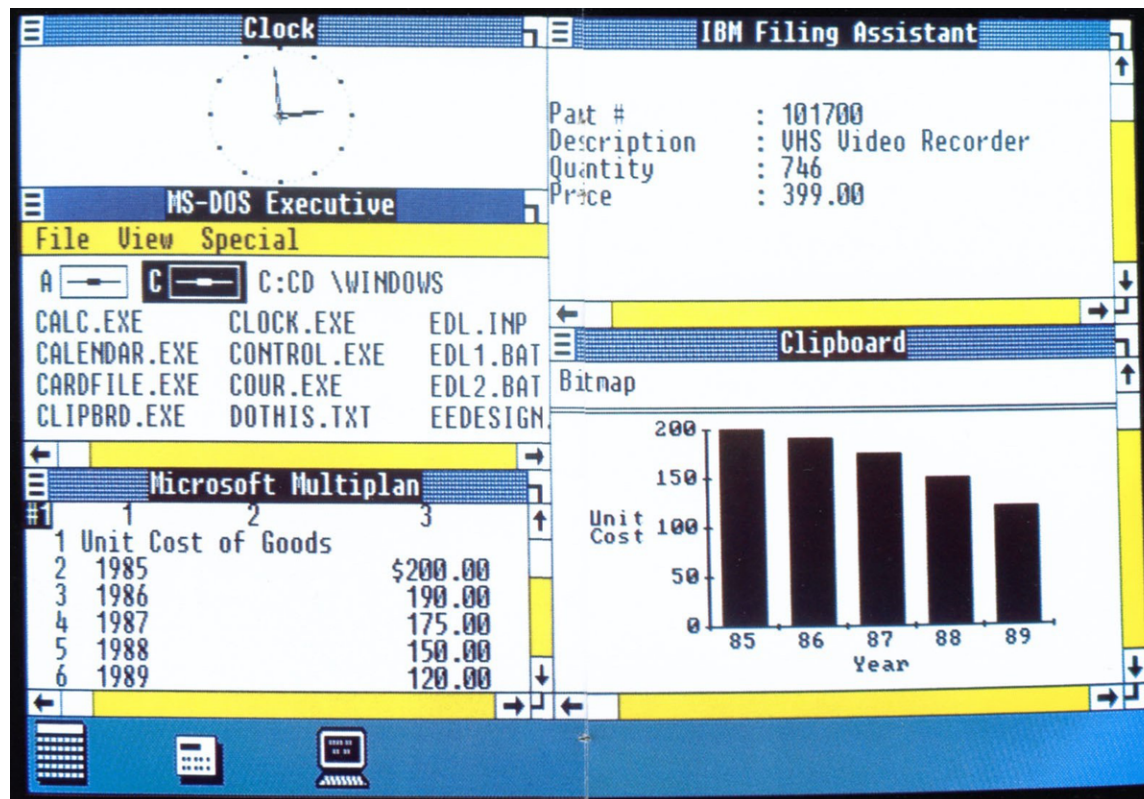
Wer hat schon BSD verwendet?

**Mac OS X** ist ein BSD Derivat.

(basiert auf NEXTSTEP/OPENSTEP, das auf BSD basiert)

# MS WINDOWS

proprietäres Betriebssystem, grafische Oberfläche



By Microsoft - [https://commons.wikimedia.org/wiki/File:Microsoft\\_Windows\\_1.0\\_pages2\\_3.jpg](https://commons.wikimedia.org/wiki/File:Microsoft_Windows_1.0_pages2_3.jpg), Public Domain, Link

# BETRIEBSSYSTEM ARCHITEKTUREN (KERNELS)

- › Monolithischer Kernel
  - » starke Abhängigkeiten zwischen Komponenten
- › Microkernel
  - » User-space übernimmt mehr Aufgaben des Kernels
- › Hybrid Kernel
  - » Mix aus monolithischem und micro Kernel
- › Exokernels
  - » Research Kernel für sehr minimalistischen Ansatz

# RESSOURCENVERWALTUNG

- › Interrupts
- › Multi-Tasking
- › Scheduler
- › Deadlock

# INTERRUPTS

- › Wichtige Funktion von Betriebssystemen
- › Ermöglicht mit der Umwelt zu interagieren
- › Ohne Interrupts müsste OS durchgehend alle möglichen Input-Quellen überwachen
- › Interrupt wird durch Hardware/Software ausgelöst und meldet dem OS "Handlungsbedarf"
- › Je nach Interrupt-Typ wird entsprechend geantwortet (abgearbeitet)

# PROZESSOR-ZEIT & MULTI-TASKING

Multi-Tasking bedeutet mehrere Aufgaben 'gleichzeitig' auszuführen

- › Prozesse werden sehr schnell nacheinander 'ausgeführt' (Zeit-Multiplexing)
- › Echte Parallelisierung ist nur auf Mehr-Kernsystemen möglich (Multiprocessing)
- › Betriebssystem steuert welcher Prozess wieviel Zeit erhält

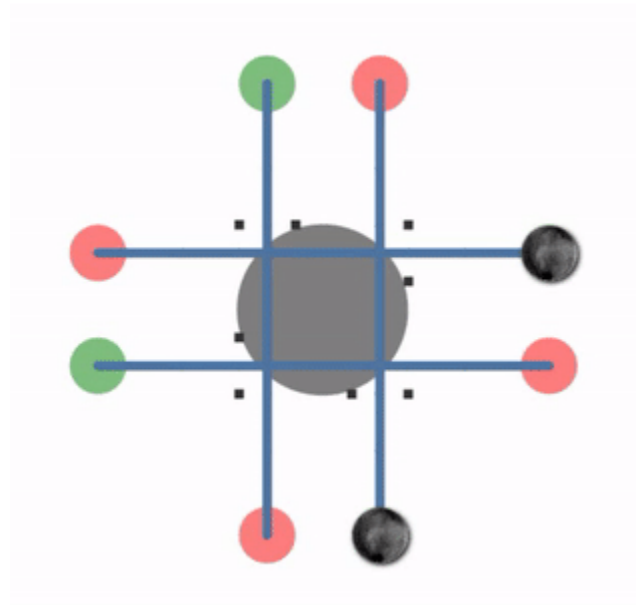


# PROZESS SCHEDULER

- › Der Kernel eines Betriebssystems beinhaltet einen so genannten Scheduler
- › Dieser bestimmt wie viel Zeit jeder der laufenden Prozesse für Berechnungen zur Verfügung gestellt bekommt
- › Er legt auch die Reihenfolge der Prozessor-Zeit-Slots fest
- › Verschiedene Strategien sind möglich:
  - » Keine Bevorzugung: First come, first serve!
  - » Bewertung der Prozesse nach Wichtig-/Dringlichkeit

# DEADLOCK

Mehrere Prozesse warten auf Ressourcen von anderen Prozessen, keiner kann weiter machen: alle stehen!



Von [Marble machine](#) - Eigenes Werk, [CC-BY-SA 4.0](#), [Link](#)

# VERMEIDUNG VON DEADLOCK UND RACE-CONDITIONS

## › **Mutex** (Mutual Exclusive)

- ›› Bestimmte Code-Bereiche werden für mehrfachen Zugriff gesperrt (=Prozess schläft)
- ›› Erst nach Freigabe, kann der nächste Prozess auf die Resource zugreifen (=Prozess aufwecken)

## › **Semaphore**

- ›› Counter (# Ressourcen), wenn 0, Prozess zu Warteschlange
- ›› Warteschlange:
  - › Speichert schlafende Prozesse und kann diese gezielt wecken

# SPEICHERVERWALTUNG UND SWAPPING

Memory Management Unit verwaltet den Speicher und lagert bei Bedarf Daten aus

- › Legt fest:
  - » Welches Programm auf welche Speicherbereiche zugreifen darf
  - » Welche Daten im Speicher ausgelagert werden
    - › Auslagern: Daten vom schnellen RAM auf langsame Festplatten
    - › Bei Bedarf müssen Daten wieder zurück ins RAM geladen werden

# ARTEN VON BETRIEBSSYSTEMEN

- › Echtzeit Betriebssysteme:
  - » meist in eingebettete Systemen (Auto, Robotik, etc.)
- › Single-tasking BS:
  - » Nur ein Programm kann ausgeführt werden
- › Multi-tasking BS:
  - » Mehrere Programme können 'gleichzeitig' ausgeführt werden

# ARTEN VON BETRIEBSSYSTEMEN

- › Einzel- & Multi-User BS:
  - » Unterscheidung der Benutzer (z.B.: über Benutzeraccounts)
  - » Programme und Speicher (Rechtmanagement) muss verwaltet werden
- › Verteilte (Betriebs)Systeme:
  - » Verbinden mehrere (verteilte) Computer zu einem System
- › Eingebette Betriebssysteme:
  - » Werden für spezielle Geräte entwickelt (z.B.: Android, iOS)

# LOC / OS

Lines of Code per Operating System\*\*

OS OS	Jahr Jahr	KLOC (LOC*1000) KLOC (LOC*1000)
UNIX 1.0	1971	10
Windows 3.1	1992	2.500
Mars Curiosity Rover		5.000
Android		12.000
Boing 787		14.000
Linux Kernel 3.1	2013	15.000
Windows 7		40.000
Mac OS 10.4	2005	86.000
Debian 5	2009	324.000
Debian 7	2012	419.000



# FROH#RAGE#TAGE



## UND GUTEN RUTSCH!

