



Betriebssysteme und Netzwerke

Vorlesung 1



Artur Andrzejak

Was ist ein Betriebssystem (**BS**)?

- ▶ „Ein Programm, das immer laufen muss“
 - ▶ Treffend, aber keine wirkliche Definition
- ▶ Besser: „Softwareschicht zwischen Hardware und den Anwendungsprogrammen“
- ▶ Intuitiv: ein Geflecht von Programmen, die den Benutzern und den (Anwendungs-)Programmen helfen, die Hardware zu verwenden
- ▶ Eine Aufzählung der Aufgaben eines BS kann das konkreter machen
 - ▶ Was sind die primären Aufgaben eines BS?
 - ▶ Auflösung kommt später

Komplexität der Betriebssysteme

- ▶ Wie viele Zeilen Quellcode hat ein „großes“ BS?
- ▶ Red Hat Linux v7.1 (April 2001): über 30 Mio. Zeilen Code
 - ▶ “If developed by conventional proprietary means, it would have required about 8,000 person-years and would have cost over \$1 billion (in year 2000 U.S. dollars)” (Wikipedia - "Source lines of code")

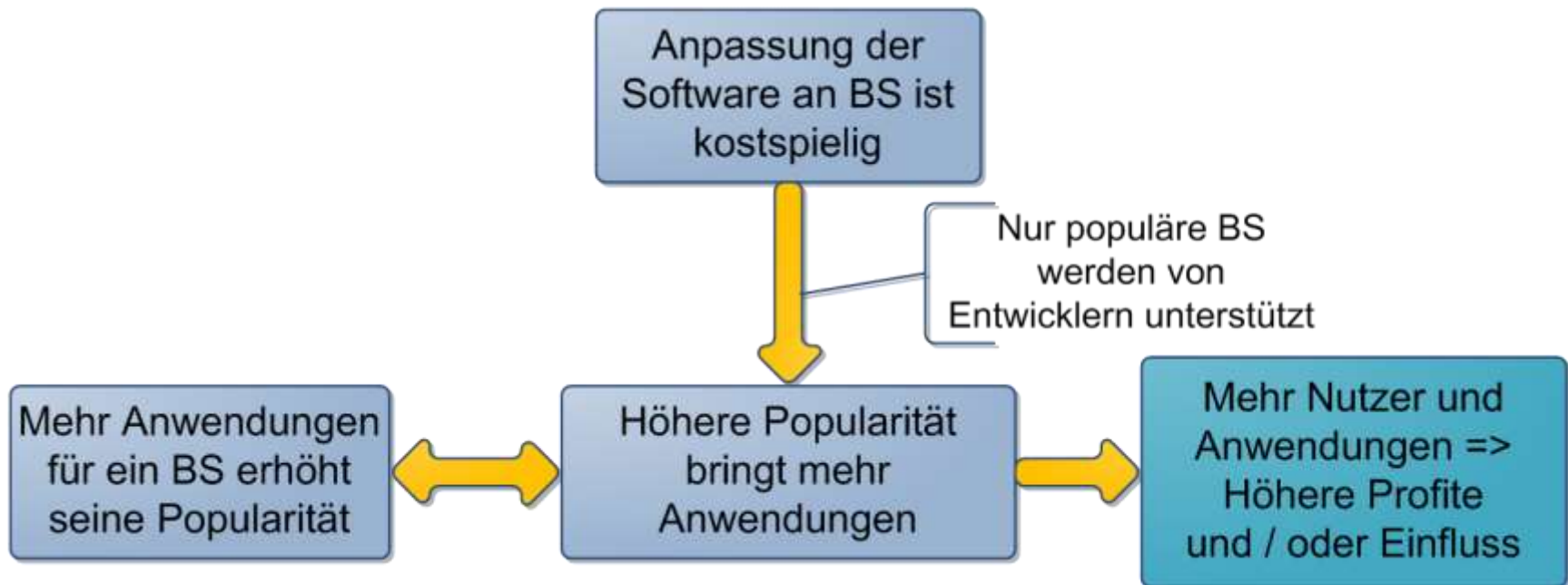
In Millionen SLOC (source lines of code)

- ▶ OpenSolaris: 9.7
- ▶ Linux kernel 2.6.32: 12.6
- ▶ Mac OS X 10.4: 86
- ▶ Debian 4.0: / Debian 5.0: 283 / 324

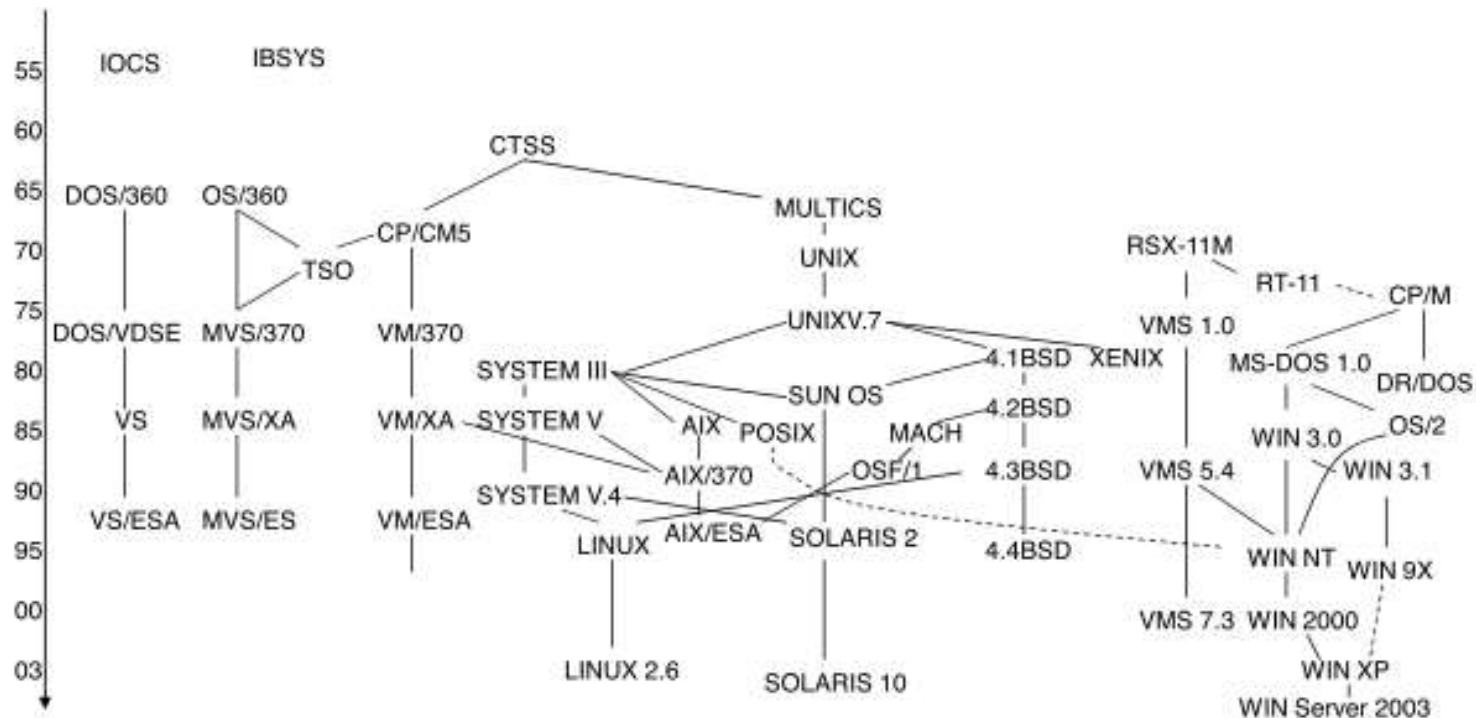
Komplexität des BS Microsoft Windows

Jahr	Version	SLOC (Mio.)
1993	Windows NT 3.1	4-5
1994	Windows NT 3.5	7-8
1996	Windows NT 4.0	11-12
2000	Windows 2000	Über 29
2001	Windows XP	40
2006	Vista	50
2009	Windows 7	40
2015	Windows 10	27-50

Betriebssysteme und Anwendungen



Geschichte der Betriebssysteme

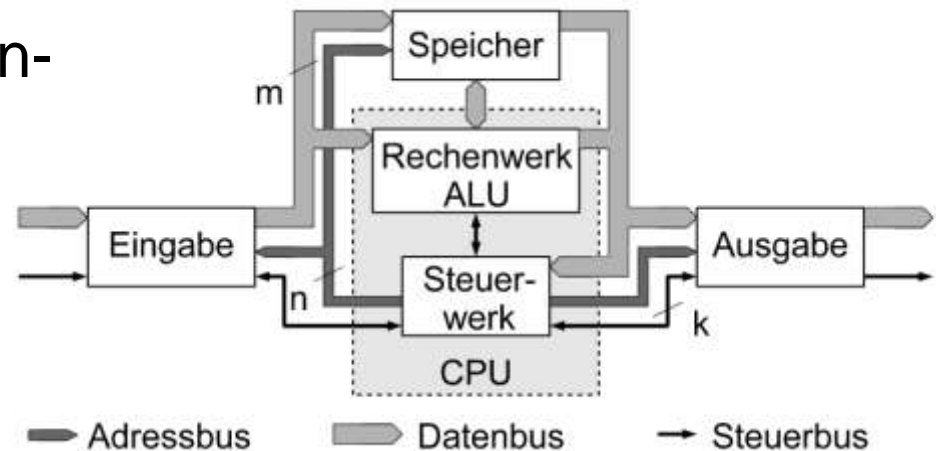
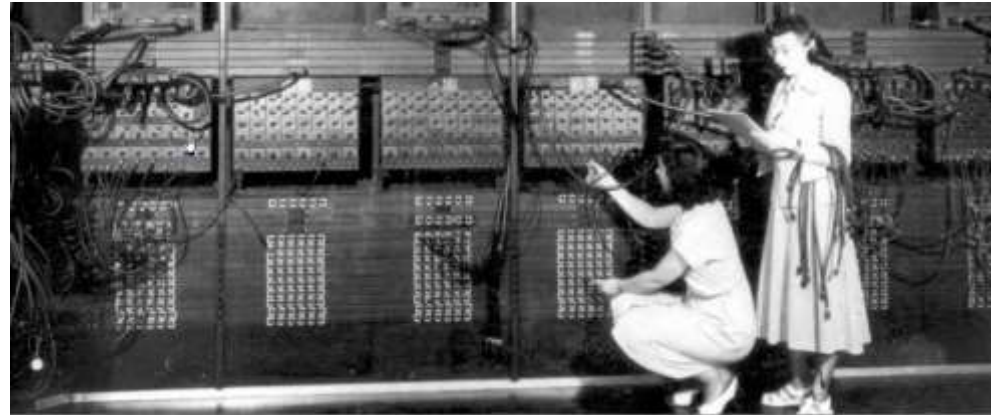


Rechner der Erster Generation 1941 - 1955

- ▶ Erste funktionierende Digitalcomputer
 - ▶ 1941: **Z3** von Konrad Zuse, Berlin (Relais)
 - ▶ 1943: **Colossus** in Bletchley Park, UK (2500 Röhren)
 - ▶ 1944: **Mark I** in Harvard Univ. (Relais, Schalter)
 - ▶ 1946: **ENIAC** von William Mauchley / J. Presper Eckert, Univ. of Pennsylvania (17.468 Elektronenröhren)
- ▶ Videos zu ENIAC
 - ▶ ENIAC: Electronic Numerical Integrator And Computer
 - ▶ <https://www.youtube.com/watch?v=goi6NAHMKog>
 - ▶ Bis ca. 2:35 (min:sec) – [01a]
 - ▶ [opt] Computer Pioneers - Pioneer Computers Part 2
 - ▶ <https://www.youtube.com/watch?v=wsirYCAocZk>
 - ▶ Ca. 2:00 bis 16:00 (min:sec)

Von Neumann-Architektur (VNA)

- ▶ ENIAC: kein Programmspeicher
 - ▶ Die „Programmierung“ erfolgte durch das Umstecken von Kabeln
- ▶ John von Neumann veröffentlichte 1945 das Konzept der VNA (Princeton-Architektur)
 - ▶ Schaltungskonzept für einen universeller Rechner
 - ▶ Speicher enthält Daten und Programmcode
 - ▶ Umsetzung in EDVAC



Details Erste Generation 1941-1955

- ▶ Rechner bestanden aus Relais und Elektronenröhren
- ▶ Programme waren in Assembler oder später in FORTRAN geschrieben
- ▶ Programme wurden „umgesteckt“, oder später aus Lochkartenstapeln eingelesen
- ▶ Ressourcenzuteilung
 - ▶ Der Programmierer trug sich in einen Aushang an der Wand, ging in den Maschinenraum, „programmierte“, und hoffte auf keinen Ausfall von einem der 10-20k Röhren
 - ▶ Programm endete früher => verlorene Zeit
 - ▶ Zeitscheibe nicht ausreichend => Programmabbruch
- ▶ Keine Betriebssysteme: jedes (Anwender-)Programm nutzte die Hardware direkt

Zweite Generation 1955 - 1964

- ▶ Die Erfindung des **Transistors** (1947) führte zur Kommerzialisierung der Computer (Mitte 1950er)
- ▶ UNIVAC I (1951): 5.2k Röhren, 18k Kristall-Dioden
 - ▶ 1905 Rechenoperationen pro Sekunde, 1000 Worte mit zwölf Dezimalstellen; Preis: 1.5 Mio USD
 - ▶ Video [01a]: <https://www.youtube.com/watch?v=goi6NAHMKog>
 - ▶ Ab 4:55 bis Ende 7:14 (min:sec)
- ▶ Preise für Computer damals: von 50 kUSD bis 1 Mio. USD (heutiger Wert: 400 kUSD bis 7 Mio. USD)
 - ▶ Große Unternehmen, obere Behörden, Universitäten konnten sich **Großrechner (Mainframes)** leisten

Ineffizienzen

▶ Üblicher Betrieb

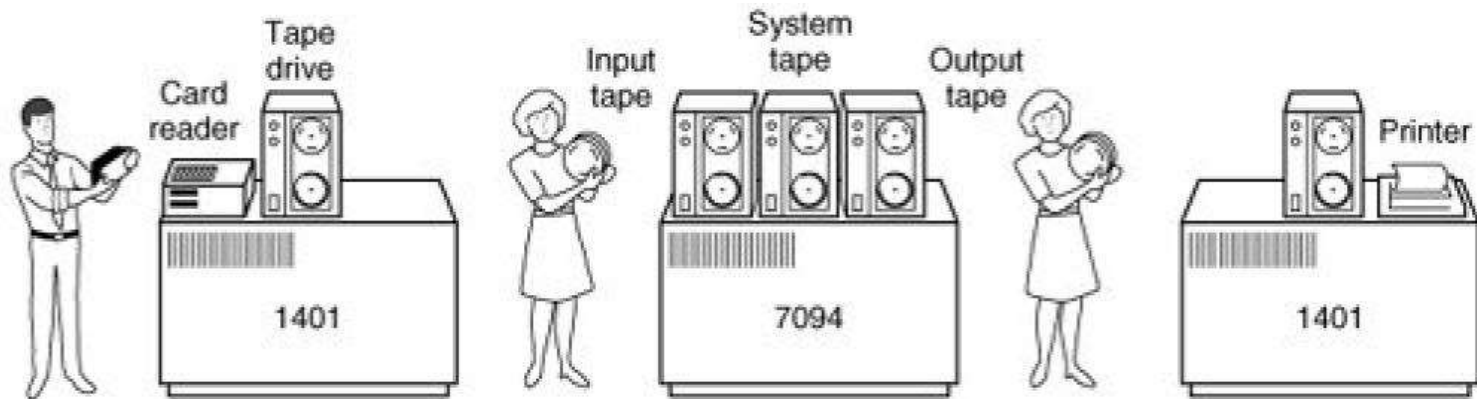
- ▶ Programmierer stanzt sein Programm (FORTRAN, Assembler) auf Lochkarten und übergab es einem Operator
- ▶ Operator las den Lochkartenstapel ein und startete die Verarbeitung
- ▶ Nach Beendigung ging er zum Drucker und brachte den Ausdruck in den Ausgaberaum
- ▶ Wenn Computer fertig war, musste Operator den nächsten Lochkartenstapel holen und einlesen lassen, ...

▶ Ineffizienzen

- ▶ Rechner wartete oft ungenutzt, bis die Daten in den Speicher kamen
- ▶ Ein Operator (Bediener) war notwendig

Stapelverarbeitung

- ▶ Verbesserung: **Stapelverarbeitung** (batch processing)
 - ▶ Neu: Einlesen der Lochkarten und Ausgabe: separate Einheiten
 - ▶ Betriebssystem liest Jobs von Magnetband und startet automatisch den folgenden Job, wenn ein Job beendet ist



Vorbereitung der Eingabe
(kleinere Maschine)

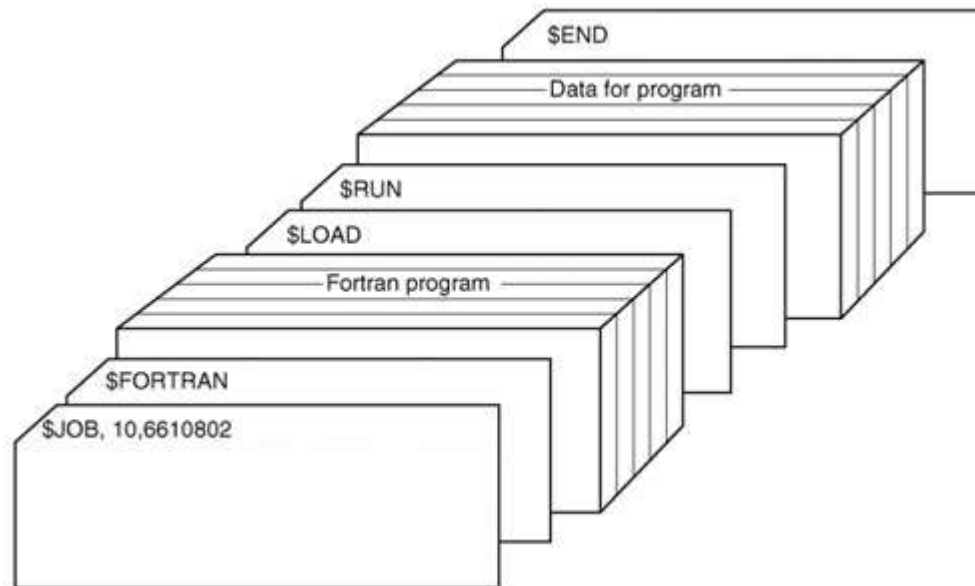
Eigentliche Verarbeitung:
„**Abarbeitung des Stapels**“
(größere Maschine)

Ausgabe
(kleinere Maschine)

- ▶ Definition: Stapelverarbeitung ist die Abarbeitung einer Reihe von Programmen ohne manuelle Intervention

Struktur eines Rechenjobs

- ▶ Betriebssysteme jener Zeit:
 - ▶ FMS (Fortran Monitor System)
 - ▶ IBSYS (IBM Betriebssystem für IBM 7094)
- ▶ Vorläufer von **Shell-Skripten**: **Kontrollkarten zur automatischen Steuerung des Ablaufs**
 - ▶ Z.B. das Einlesen des Compilers (vom Band), Kompilierung, Ausführung usw. erfolgten automatisch

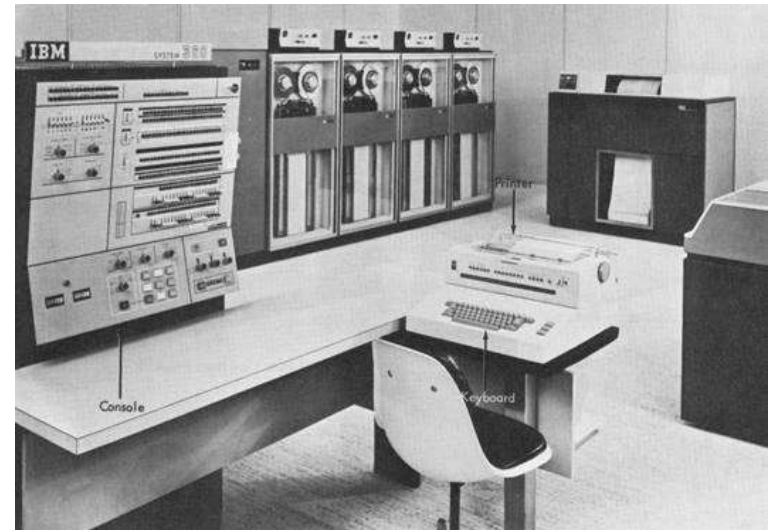


Dritte Generation 1964 - 1980

- ▶ Bis dahin gab es zwei Produktstrategien
 - ▶ **Wortorientierte**, große, wissenschaftliche Rechner für numerische Berechnungen (z.B. IBM 7094)
 - ▶ **Zeichenorientierte**, kommerzielle Rechner für das Sortieren und Ausdrucken von Bändern in Banken und Versicherungen (z.B. IBM 1401)
- ▶ Probleme
 - ▶ Zwei Produktlinien waren zu teuer
 - ▶ Beim Übergang zu einer schnelleren Maschine war die Software (**SW**) nicht mehr kompatibel
 - ▶ Auch innerhalb einer Produktlinie

Das System/360 von IBM

- ▶ IBM versuchte, dieses Problem mit einer Familie von Software-kompatiblen Rechnern zu lösen
 - ▶ **System/360**, eingeführt am 7. April 1964
 - ▶ Spektrum reichte von „1401“ bis „7094“ und größer
- ▶ Unterschiede im max. CPU-geschwindigkeit, Speicher, Anzahl der I/O-Geräte usw.
- ▶ Später hat IBM kompatible Nachfolger herausgebracht
 - ▶ 370, 4300, 3080, 3090
 - ▶ Auch die aktuelle Großrechner-architektur von IBM – System z (bzw. zSeries) ist ein Ableger davon



System/360 Model 40b

Das System/360 - Probleme

- ▶ Die gesamte SW – auch das Betriebssystem OS/360 – sollte auf allen Modellen arbeiten
 - ▶ Das erforderte ein BS, das 100 bis 1000 Mal umfangreicher als FMS war
 - ▶ => Mio. Zeilen Assemblercode, geschrieben von Tausenden von Programmierern, mit Tausenden von Fehlern ...
- ▶ Mehr im Buch des OS/360 Architekten Fred Brooks: „The Mythical Man-Month” (1975, ..,1995)
 - ▶ *“Adding manpower to a late software project makes it later.”*, http://en.wikipedia.org/wiki/The_Mythical_Man-Month
- ▶ Trotzdem, System/360 war ein riesiger Erfolg
 - ▶ 70% des Umsatzes von IBM USA in 1969

OS/360 und Multiprogrammierung

- ▶ Problem: In der Datenverarbeitung kann die **Ein-/Ausgabezeit** 80-90% der Joblaufzeit betragen
 - ▶ Die damals teure CPU ist nicht ausgelastet
- ▶ Eine wichtigste Neuerung von OS/360 war die **Multiprogrammierung** (multiprogramming)
 - ▶ Multiprogrammierung = Wechsel der Belegung von CPU zwischen mehreren Jobs
 - ▶ Um die I/O-Wartezeiten zu vermeiden
- ▶ Die Multiprogrammierung ist ein zentrales Konzept der BS – begleitet uns ständig im 1. Teil der VL
 - ▶ Macht leider das BS viel, viel komplexer
 - ▶ OS/360 hatte diese Fähigkeit nicht von Anfang an

Gegenüberstellung der Begriffe

▶ Stapelverarbeitung

- ▶ Abarbeitung einer Liste von Programmen (Jobs) ohne manuelle Intervention

▶ Multiprogrammierung

- ▶ (schneller) Wechsel der Belegung von CPU und Ressourcen zwischen mehreren Jobs
- ▶ Wechsel meist beim Zugriff auf die I/O-Geräte, um Wartezeiten zu vermeiden

▶ Multitasking bzw. Mehrprozessbetrieb

- ▶ Fähigkeit eines Betriebssystems, mehrere Jobs nebenläufig auszuführen
- ▶ Verbesserung gegenüber der Multiprogrammierung ist die (Pseudo-) Gleichzeitigkeit: die beteiligten Jobs werden in kurzen Abständen aktiviert

Andere Neuerungen der dritten Generation

▶ **Timesharing**

- ▶ Erlaubt mehreren Benutzern, an einem Computer gleichzeitig zu arbeiten, indem sie sich die Rechenzeit des einzigen vorhandenen Prozessors teilen
- ▶ Beschrieben von Bob Bemer in 1957 (einer der Väter des ASCII-Standards)
- ▶ Setzt Multiprogramming und Mehrbenutzersystem voraus

▶ Video zu Problemen des Batch-Betriebs und zu Time-Sharing

- ▶ 1963 Timesharing: A Solution to Computer Bottlenecks
- ▶ <https://www.youtube.com/watch?v=Q07PhW5sCEk> von Minute 8:00 bis ca. Minute 15:00

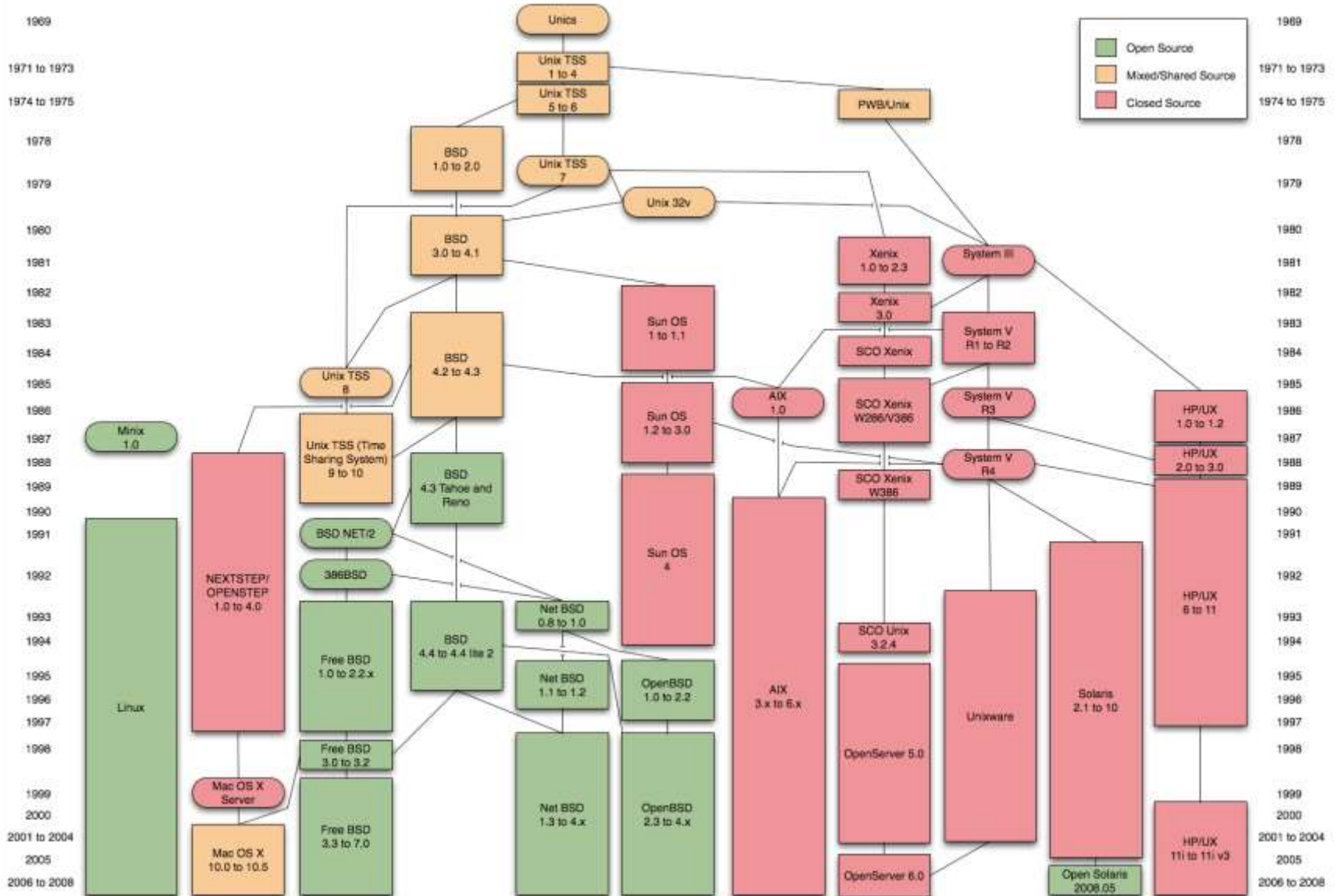
BS der dritten Generation

- ▶ Das erste Timesharing-System hieß **Compatible Time Sharing System (CTSS)** und wurde auf einer modifizierten 7094 entwickelt (MIT, J. McCarthy, 1957)
 - ▶ Nicht erfolgreich wegen Mangel an Hardware-Schutz
- ▶ Danach beschlossen MIT, Bell Labs und General Electric eine Maschine zu bauen, die Hunderte von Benutzern gleichzeitig unterstützen sollte (1963)
- ▶ => **MULTICS (MULTiplexed Information and Computing System)**, siehe www.multicians.org
 - ▶ Kommerziell kein Erfolg
 - ▶ Aber konzeptionell und wissenschaftlich prägend

BS der dritten Generation - UNIX

- ▶ In den 1960er entstanden **Minicomputer**: kleiner aber deutlich billiger als Mainframes
 - ▶ DEC PDP-1 hatte ca. 8 KB RAM, kostete aber nur 5% von IBM 7094
- ▶ Ein solcher Minicomputer (PDP-7) wurde benutzt, um eine Einbenutzerversion von MULTICS zu schreiben
- ▶ MULTICS-Weiterentwicklung führte 1971 zu **UNIX**
- ▶ UNIX Quelltexte waren frei verfügbar => **viele Versionen**
 - ▶ Wichtigste Versionen: **System V** (AT&T) und **BSD-UNIX** (Berkeley Software Distribution)
 - ▶ **POSIX**-Standard (von IEEE) definiert einen Teil der Systemschnittstelle, um die Kompatibilität zu erreichen

Es gibt viele Varianten von UNIX



UNIX Eigenschaften

- ▶ Video [01b]: Computerhelden mit coolen Frisuren
 - ▶ AT&T Archives: The UNIX Operating System
<https://www.youtube.com/watch?v=tc4ROCJYbm0>
- ▶ Ab 4:35 bis 6:30 (min:sec)
 - ▶ Unix als “Baukasten” für eigene Software
 - ▶ Pipelining
- ▶ Ab 11:35 bis 15:36 (min:sec)
 - ▶ Dateisystem
 - ▶ Hierarchische Folder (Directories)
 - ▶ Shell (ab 14:30)

Vierte Generation 1980 bis heute

- ▶ Diese Zeit ist dominiert von den PCs / Mikrocomputern – „**personal computing**“
- ▶ 1974: erste Allzweck-8-Bit-CPU – 8080 von Intel
 - ▶ Digital Research entwickelte dafür (und für Z80 von Zilog) das System **CP/M** (Control Program for Microcomputers)
- ▶ In den frühen 1980er Jahren entwarf IBM den IBM-PC und kaufte dafür das **MS-DOS** von Microsoft ein
 - ▶ Es folgten viele Windows-Betriebssysteme
- ▶ Ausprobieren: [Mac von 1984](https://goo.gl/16Gc4l) (<https://goo.gl/16Gc4l>)
- ▶ Konzeptionelle Neuerungen
 - ▶ **GUI**: Graphical User Interface
 - ▶ Netzwerkbetriebssysteme / verteilte Betriebssysteme

Windows Zeitleiste (timeline)

Zeitleiste der Windows-Versionen von 1985 bis heute																										
Typ	1980er					1990er										2000er										2010er
	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0
16-Bit	Windows 1.0	Windows 2.0	Windows 3.0	Windows 3.1	Windows 3.11																					
9x-Linie										Windows 95	Windows 98	Windows ME														
NT-Linie								Windows NT 3.1	NT 3.5	NT 3.51	Windows NT 4.0	2000	Windows XP			Windows Vista		Windows 7								
Server-OS auf NT-Basis								Windows NT 3.1 Srv.	NT 3.5 Srv.	NT 3.51 S.	Windows NT 4.0 Server	Windows 2000 Server	Windows Server 2003			Windows Server 2008										
CE-Linie											CE 1.0	CE 2.0	CE 3.0	CE 4.0	Mobile 2003	Mobile 5.0	Mobile 6.0	Mobile 6.1	Mobile 6.5	Mobile 7.0						



Linux-Versionen

- ▶ Populärste Linux-Versionen (laut [Geek Trio](#))
 - ▶ Ubuntu
 - ▶ Fedora (Red Hat's open project)
 - ▶ openSUSE
 - ▶ Debian
 - ▶ Mandriva (ehemals Linux Mandrake)
 - ▶ Linux Mint
 - ▶ PCLinuxOS
 - ▶ Slackware
 - ▶ Gentoo
 - ▶ CentOS
- ▶ Diagramme der Timelines / Distributions
 - ▶ <http://futurist.se/gldt/>

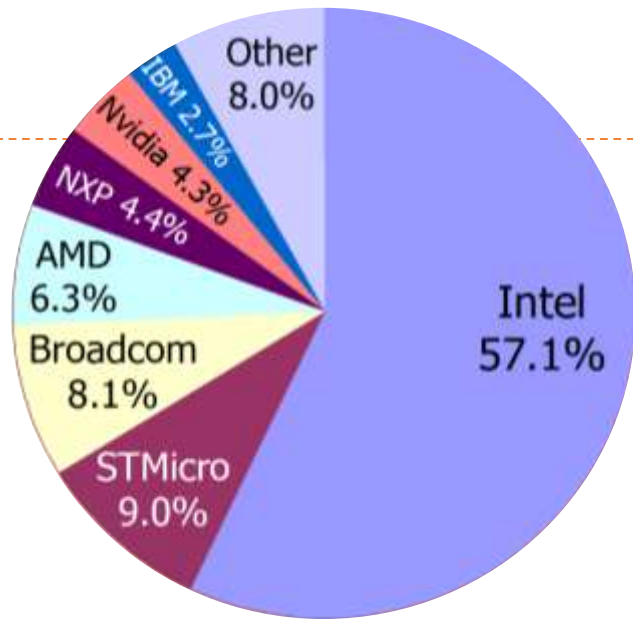
Fünfte Generation ab ca. 2007 bis heute

- ▶ BS für Mobiltelefone, Mediaplayer, Netbooks, und Tablet-Computer
- ▶ Android, Apple iOS, BlackBerry OS, Symbian, Windows Mobile/Phone
- ▶ Wesentliche Neuerungen
 - ▶ GUI für Touch-Bedienung und „no windows“-Betrieb
 - ▶ Enge Kopplung von Anwendungen via BS-Schnittstellen
 - ▶ Unterstützung von Sensoren und diversen Kommunikationskanälen im BS

BS	Marktanteil (12/2015)	Marktanteil (10/2016) (Quelle)
Android		
Apple iOS		
Win Phone		
Black Berry		
Andere		

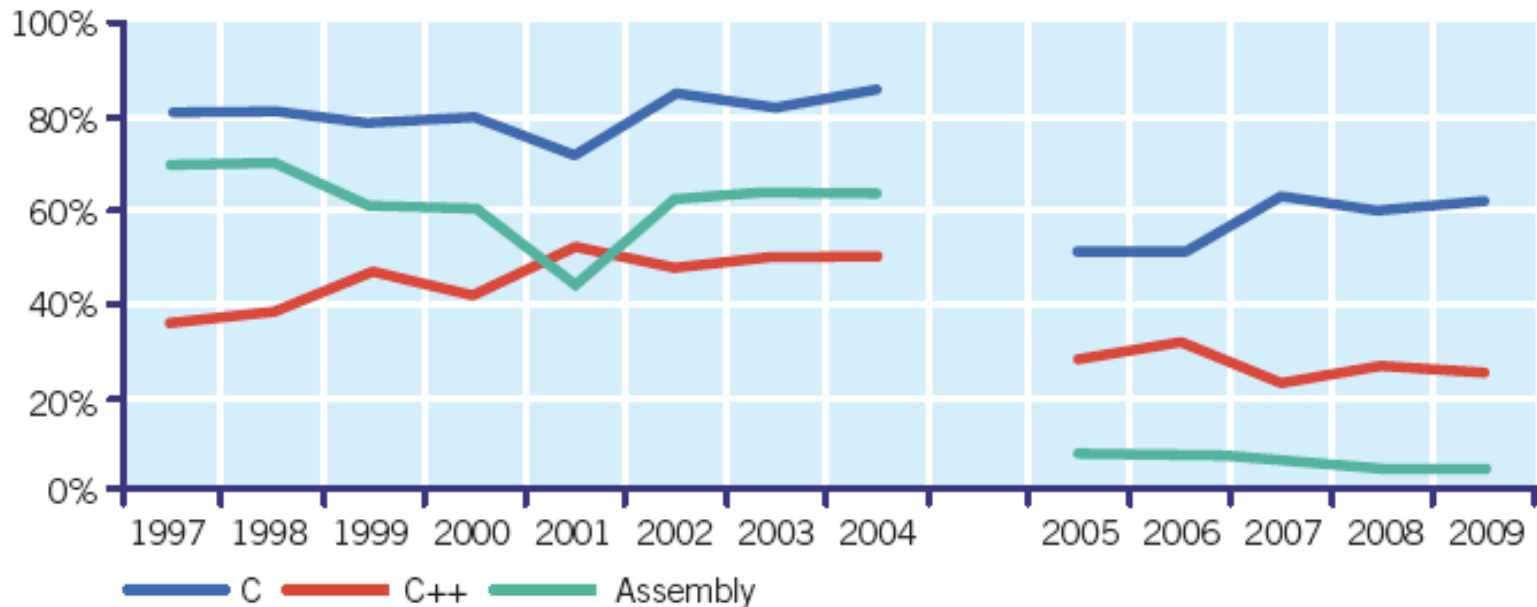
Mikroprozessoren

- ▶ Wer ist der größte Chiphersteller weltweit?
- ▶ Intel hat den größten Marktanteil (Sep 2012)
- ▶ Gesamtwert: 94.2 Mrd. USD
- ▶ Aber: Intel macht nur ca. 2% der Mikroprozessoren!
- ▶ Rest: 98% der CPUs sind sog. „**embedded processors**“: für Autos, Drucker, Kabelmodems, Telefone,...
- ▶ „**ARM** ... is considered to be market dominant in the field of processors for mobile phones ... and tablet computers...”
 - ▶ 13.04.2015; http://en.wikipedia.org/wiki/ARM_Holdings
- ▶ Mehr dazu: <http://www.computerweekly.com/news/2240226532/Arm-is-a-competitor-we-take-very-seriously-says-Intel>



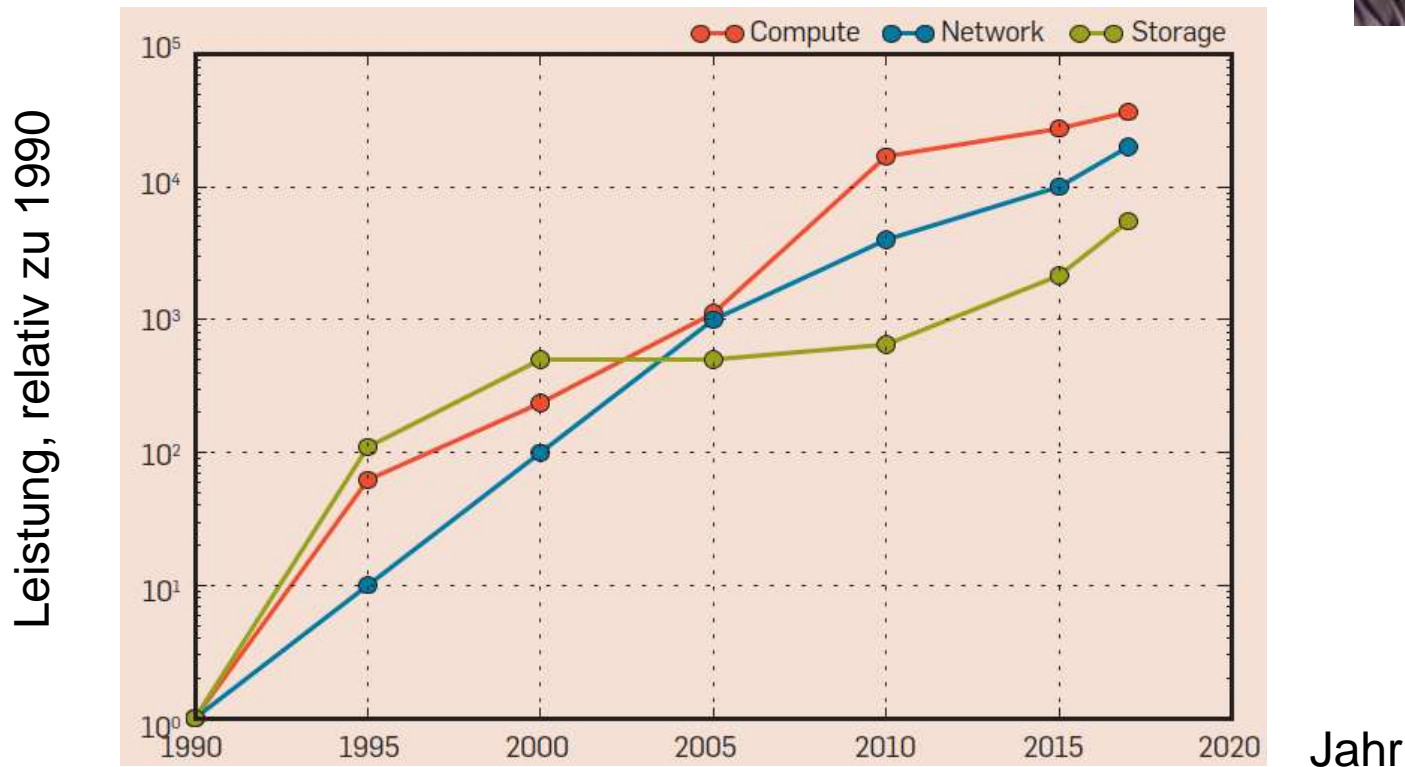
Real (wo)men program in C

- ▶ Die Programmiersprache der Wahl für Geräte mit “embedded processors” ist vorrangig **C**
- ▶ Antworten auf die Frage (essentiell):
 - ▶ "My current embedded project is programmed mostly in _____?"



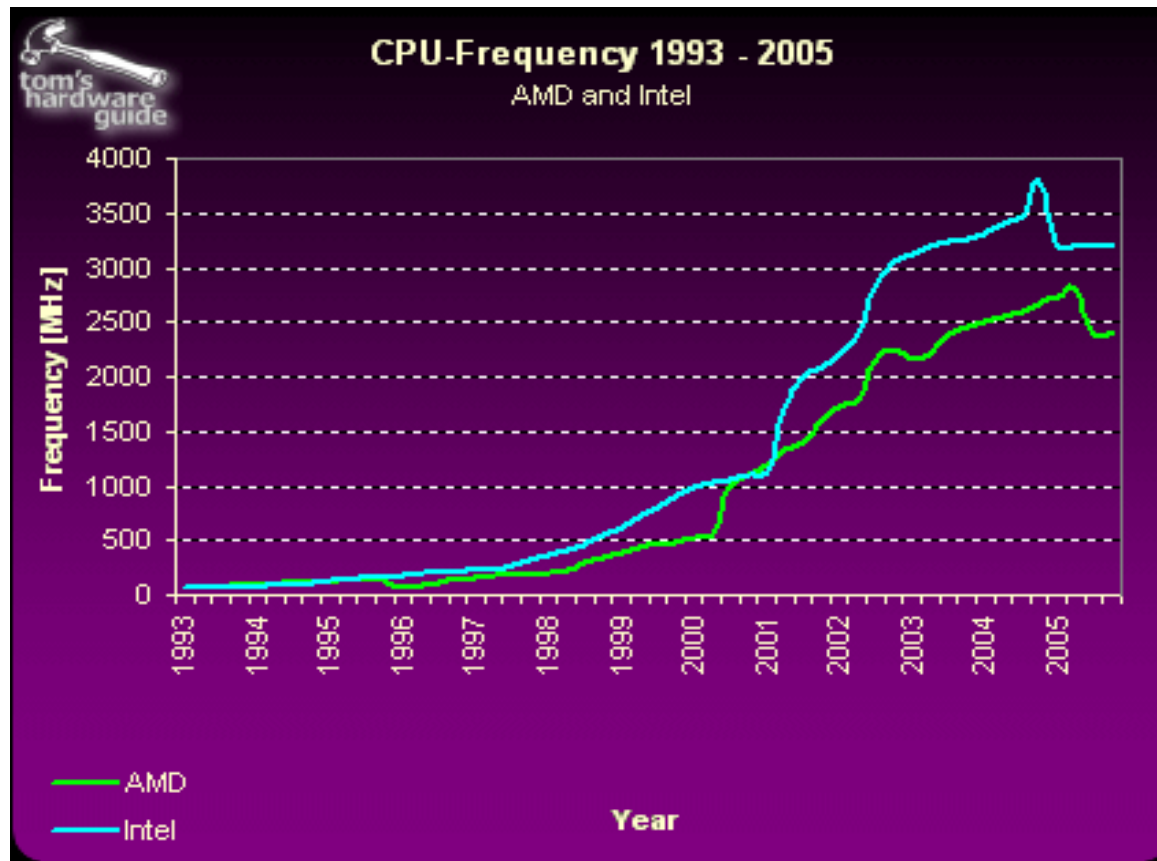
Moore's Law und die Folgen

- ▶ Moore's Law (1965): „*The density of transistors on a chip doubles every 18 months, for the same cost*”



Moore's Law und die CPU-Frequenzen

- ▶ Seit ca. 2006, die Leistung eines einzigen CPU-Kerns steigt nicht mehr nach dem Moore'schen Gesetz
- ▶ Gilt das Gesetz weiterhin?



- ▶ Ja (noch)
- ▶ => Mehr Kerne pro Chip

Wichtige Konzepte bei der Evolution der BS

- ▶ Stapelverarbeitung (batch processing)
- ▶ Multiprogramming
- ▶ Time-Sharing
- ▶ Personal Computing
- ▶ GUIs, verteilte Betriebssysteme
- ▶ BS für mobile Geräte
- ▶ Eingebettete Systeme/Prozessoren

Quellen / Weiterführende Literatur

- ▶ Tannenbaum, Kapitel 1
- ▶ Silberschatz et.al, Kapitel 2 (wenig davon)
- ▶ Per Brinch Hansen, *The evolution of operating systems*. In *Classic Operating Systems: From Batch Processing to Distributed Systems*, P. Brinch Hansen, (Ed.), 2000, Springer Verlag, New York.
- ▶ Wikipedia

Danke schön.

Hilft uns die Geschichte?

- ▶ Computerindustrie ist (primär) technologiegetrieben
 - ▶ PCs existieren nicht, weil Millionen Menschen seit der Steinzeit danach verlangten, sondern weil es jetzt möglich ist, Computer billig herzustellen
- ▶ Technologische Veränderungen lassen oft ein Konzept veralten und verschwinden
 - ▶ Aber die nächste technologische Änderung kann das alte Konzept wieder „auferstehen“ lassen
 - ▶ Es lohnt sich also, auch „alte“ Konzepte anschauen

Beispiel Prozessorentwicklung

- ▶ Die ersten Prozessoren hatten **festverdrahtete Befehlssätze**
- ▶ Die **Mikroprogrammierung** kam mit CISC Prozessoren
 - ▶ Festverdrahtete Befehle veralteten
- ▶ Mit RISC Prozessen veraltete die Mikroprogrammierung
 - ▶ Festverdrahtung wurde wieder „cool“
- ▶ Die Funktionsweise der **JVM** und **Common Language Runtime** (Teil von .NET) sind der Mikroprogrammierung wieder ähnlich

OS/360 Versionen

- ▶ 1. Version: **PCP** (Primary Control Program)
 - ▶ Konnte nur ein Programm (bzw. Job) auf einmal ausführen
- ▶ 2. Version: **MFT** (Multiprogramming with a Fixed number of Tasks)
 - ▶ Nur eine fest definierte Anzahl von Jobs kann (pseudo-) gleichzeitig ausgeführt werden
 - ▶ Speicher wird fest für jeden Job vor seinem Start zugewiesen
- ▶ 3. Version: **MVT** (Multiprogramming with a Variable number of Tasks)
 - ▶ Beliebige Anzahl von Jobs sowie eine dynamische Speichernutzung (Veränderungen zur Laufzeit)