

## Einführung Betriebssysteme

Computing-Plattformen und Netzwerke

auf Basis der Unterlagen von  
Prof. P. Mandl (HS München)

# Organisation

---

- Termine
- Folien
- Praktische Aufgaben

→ Moodle

## Umfrage

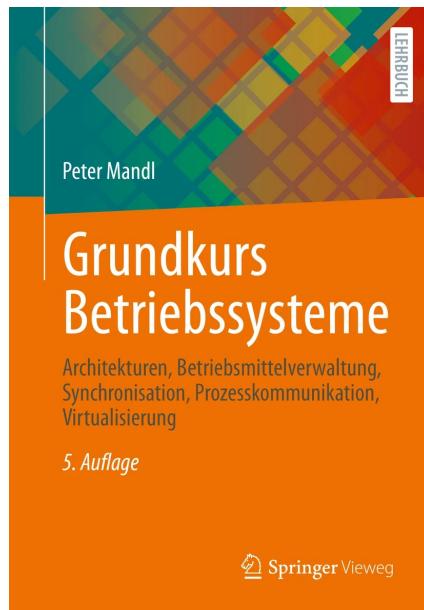
---

Welches Betriebssystem benutzt du?



# Quellen und Literaturempfehlung

- Folien der Vorlesungsreihe **Betriebssysteme** basieren maßgeblich auf den Arbeiten von Prof. Peter Mandl (Hochschule München)



P. Mandl:  
Sehr gute Einführung  
in Betriebssysteme  
→ Link: Moodle

A. Tanenbaum:  
Standardwerk für  
Betriebssysteme  
→ Bibliothek

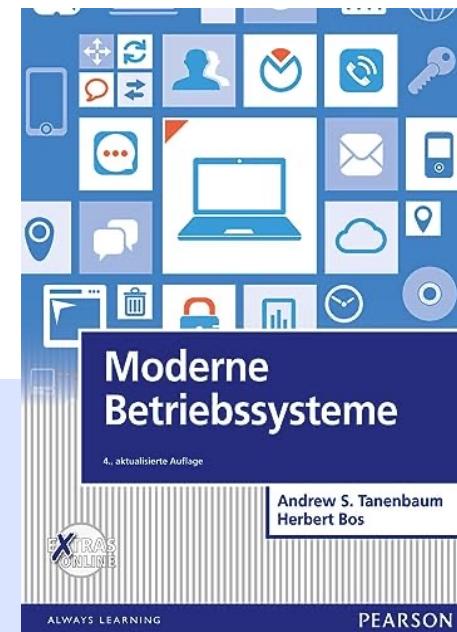


Bild: Springer Verlag

Bild: Pearson Education

## Umfrage

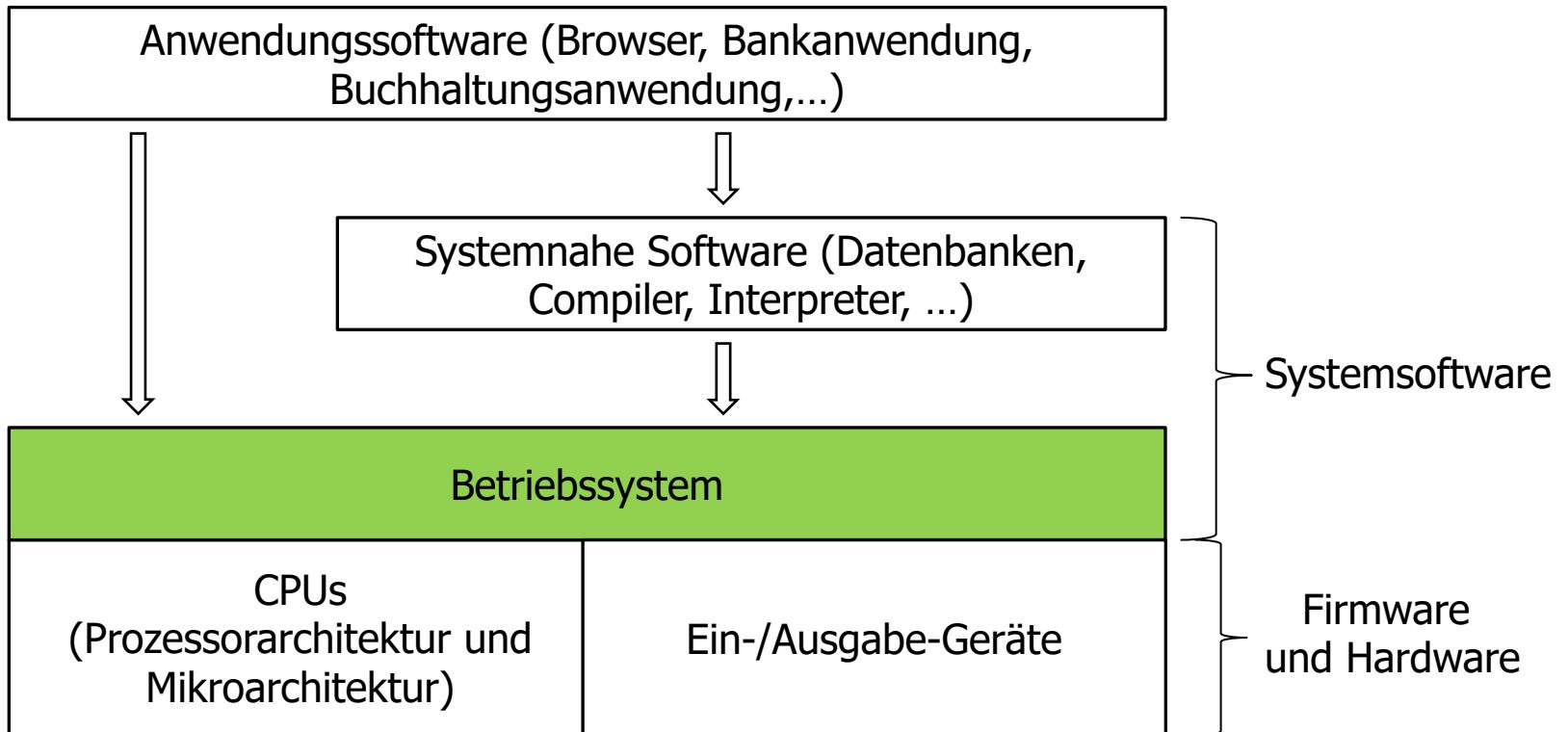
---

# Wozu brauchen wir Betriebssysteme?



# Überblick über das Gesamtsystem

---



# Grundfunktionen des Betriebssystems (1)

---

- Betriebssystem ist **Software**, die den Anwender bzw. Anwendungsentwickler von Details der Hardware entlasten soll

## Grundfunktionen des Betriebssystems (2)

---

- Modern strukturierte Betriebssysteme **kapseln** den Zugriff auf die Betriebsmittel
  - Der Zugriff funktioniert also nur über Betriebssystemfunktionen (Systemdienste)

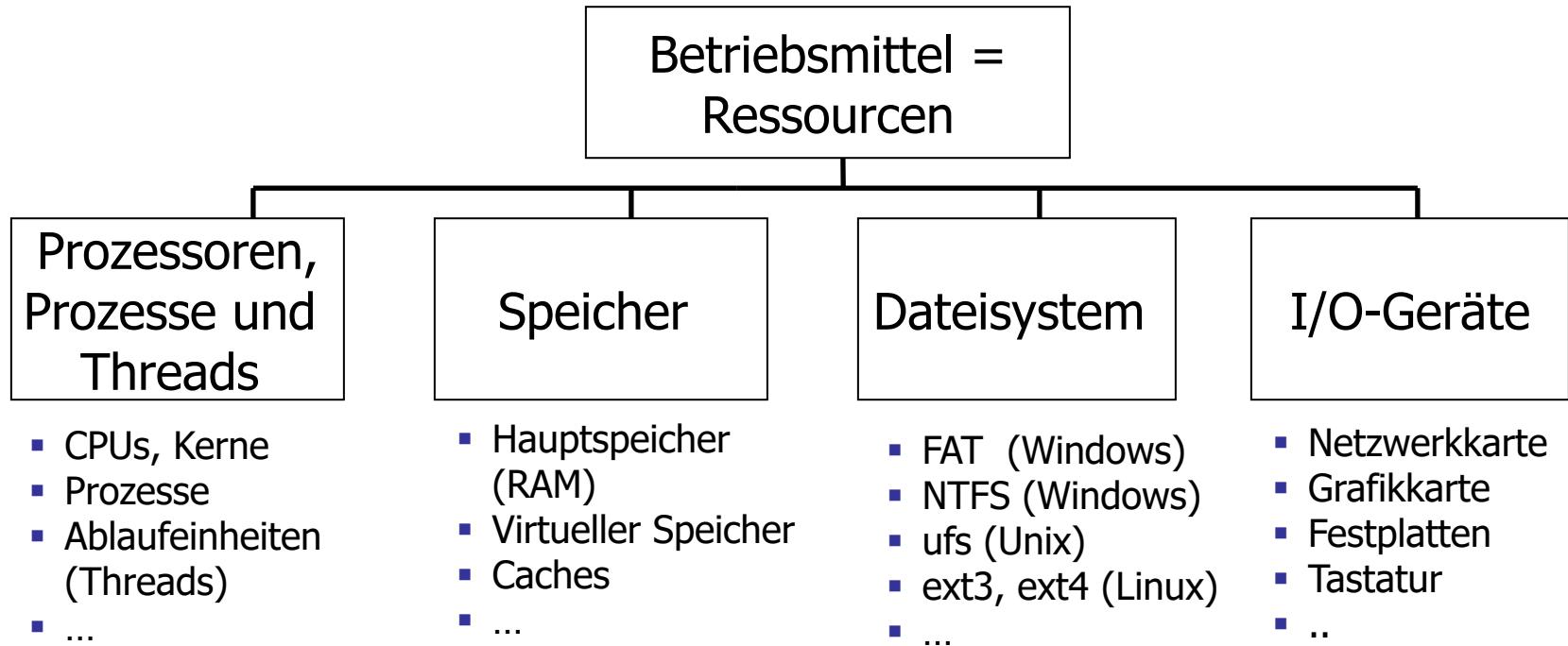
# Grundfunktionen des Betriebssystems

---

- Das Betriebssystem übernimmt die  
**Betriebsmittelverwaltung**
  - Konfliktfreier Zugriff muss sichergestellt werden
  
- **Betriebsmittel** sind
  - real (Hardware) oder virtuell (Software)
  - shared oder exklusiv
  - entziehbar und nicht entziehbar

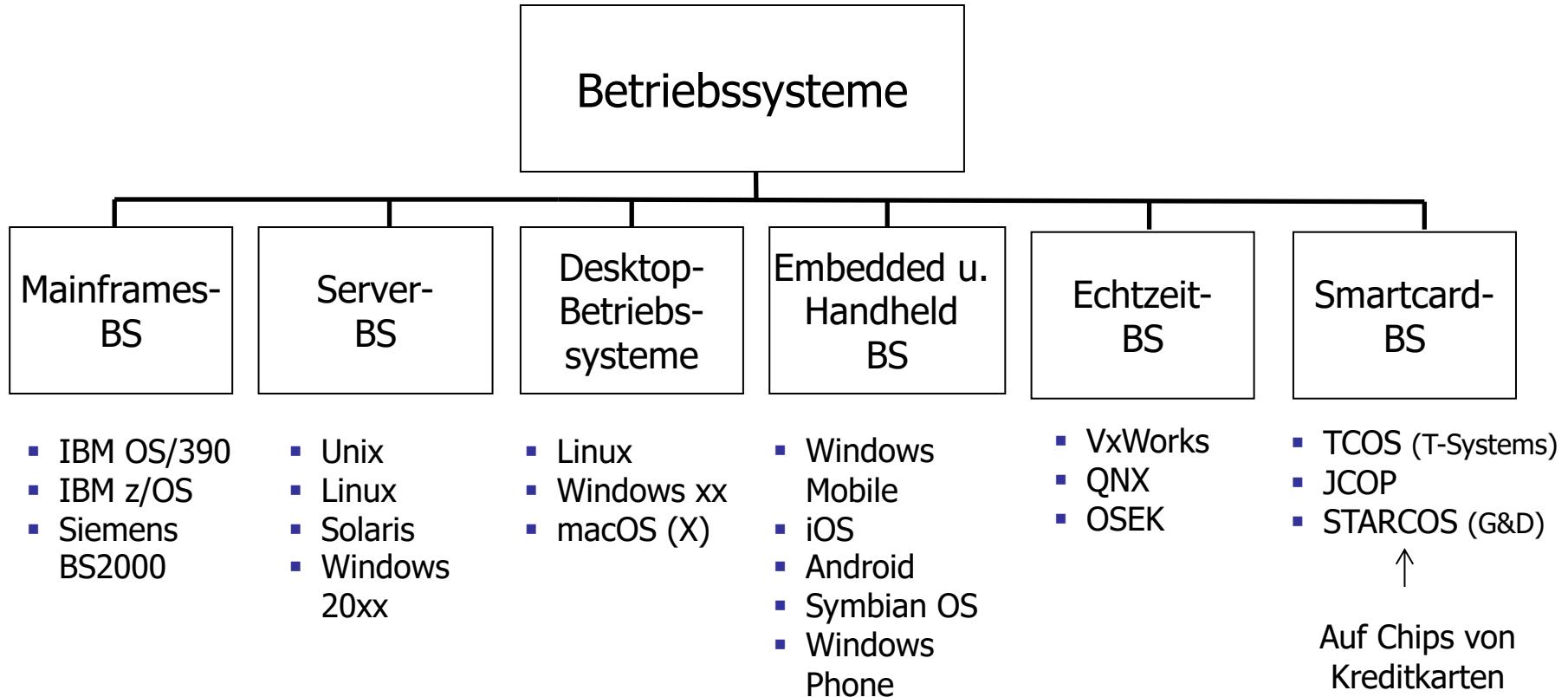
# Betriebsmittelverwaltung

---



# Arten von Betriebssystemen

---



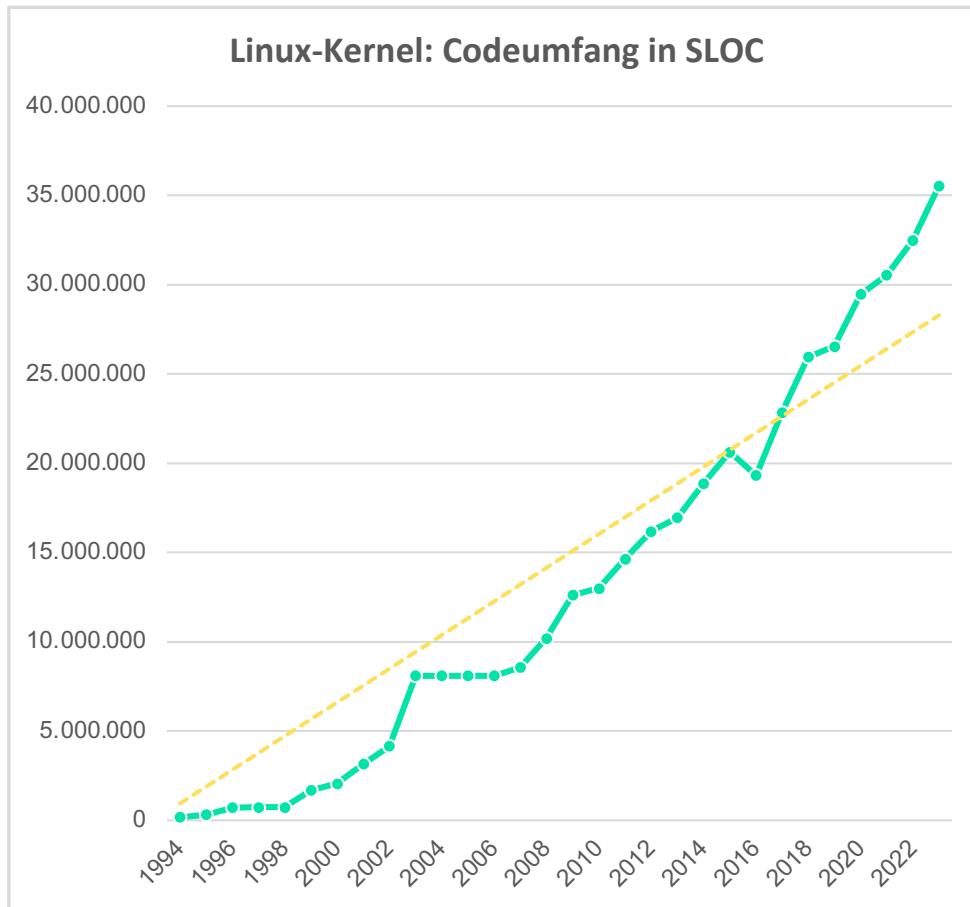
# Codeumfang einzelner Betriebssysteme

| Jahr | AT&T          | BSD             | Minix     | Linux      | Solaris           | Win NT             |
|------|---------------|-----------------|-----------|------------|-------------------|--------------------|
| 1976 | V6, 9K        |                 |           |            |                   |                    |
| 1979 | V7, 21K       |                 |           |            |                   |                    |
| 1980 |               | 4.1, 38 K       |           |            |                   |                    |
| 1982 | Sys III, 58 K | 4.2, 98 K       |           |            |                   |                    |
| 1984 |               | 4.3, 179 K      |           |            |                   |                    |
| 1987 | SVR3, 92 K    |                 | 1.0 13 K  |            |                   |                    |
| 1989 | SVR4, 280 K   |                 |           |            |                   |                    |
| 1991 |               |                 |           | 0.01, 10 K |                   |                    |
| 1993 |               | Free 1.0, 235 K |           |            |                   | 3.1, 6 M           |
| 1994 |               | 4.4 Lite, 743 K |           | 1.0, 165 K | 5.3, 850 K        | 3.5, 10 M          |
| 1996 |               |                 |           | 2.0, 470 K |                   | 4.0, 16 M          |
| 1997 |               |                 | 2.0, 62 K |            | 5.6, 1.4 M        |                    |
| 1999 |               |                 |           | 2.2, 1 M   |                   |                    |
| 2000 |               | Free 4.0, 1.4 M |           |            | 5.8, <b>2.0 M</b> | 2000, <b>29 M</b>  |
| 2007 |               |                 |           |            |                   | Vista, <b>50 M</b> |

Windows 7: 70 M

Vgl. auch Tanenbaum, 2002: K = 1.000 LOC, M = 1000.000 LOC  
LOC = lines of code      SLOC = source lines of code

# Codeumfang Linux-Kernel



- Stand Mär 2023, Linux-Version 6.1.12: ca. **35,5 Mio SLOC**
- Stand Mär 2022, Linux-Version 5.17: ca. **32,4 Mio SLOC**
- Stand Mär 2021, Linux-Version 5.17: ca. **30,5 Mio SLOC**
- Stand: Oktober 2020, Linux-Version 5.9: ca. **29,4 Mio SLOC**
- Stand: Juli 2019, Linux-Version 5.2: ca. **26,5 Mio SLOC**
- Stand: Februar 2018, Linux-Version 4.15: **25,3 Mio SLOC**
- Stand: Februar 2017, Linux-Version 4.10: **22,8 Mio SLOC**
- Stand: Januar 2016, Linux-Version 4.4: **20,8 Mio SLOC**
- Stand: Februar 2015, Linux-Version 3.19: **19 Mio SLOC**
- Stand: Januar 2014, Linux-Version 3.13: **18 Mio SLOC**
- Stand: März 2013, Linux-Version 3.8: **16 Mio SLOC**
- Stand: März 2012, Linux-Version 3.2: **15 Mio LOC**

Abgeleitet von [http://de.wikipedia.org/wiki/Linux\\_\(Kernel\)](http://de.wikipedia.org/wiki/Linux_(Kernel))

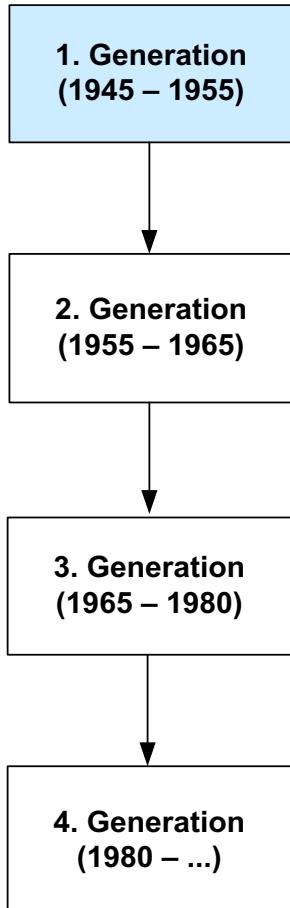
Linux-Version 1.0: **170.581 SLOC**

---

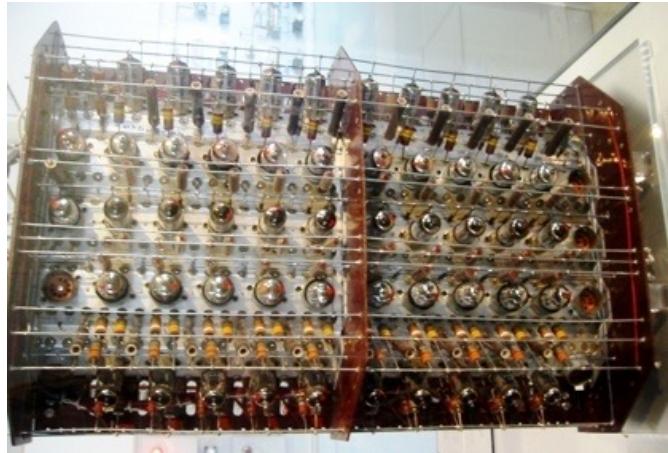
# Historische Entwicklung von Betriebssystemen

# Historische Entwicklung von Rechnern und Betriebssystemen

---



- Minimale Betriebssysteme
- Röhrencomputer
- Maschinensprache, kein Assembler
- Lochkarten ab 1950



Röhrencomputer der Rechenanlage ORACLE  
Deutsches Museum

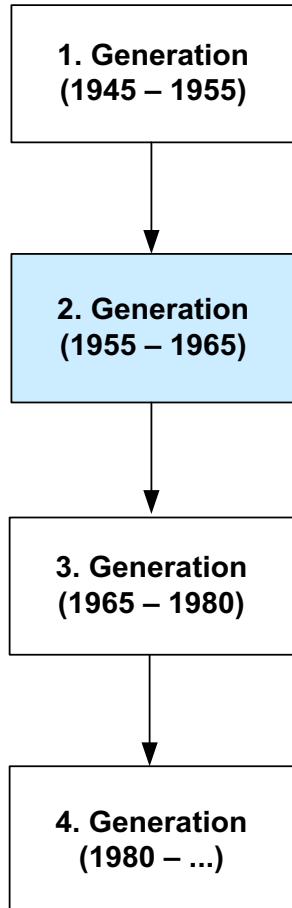


Lochkartenleser von Control Data  
Quelle: Wikipedia

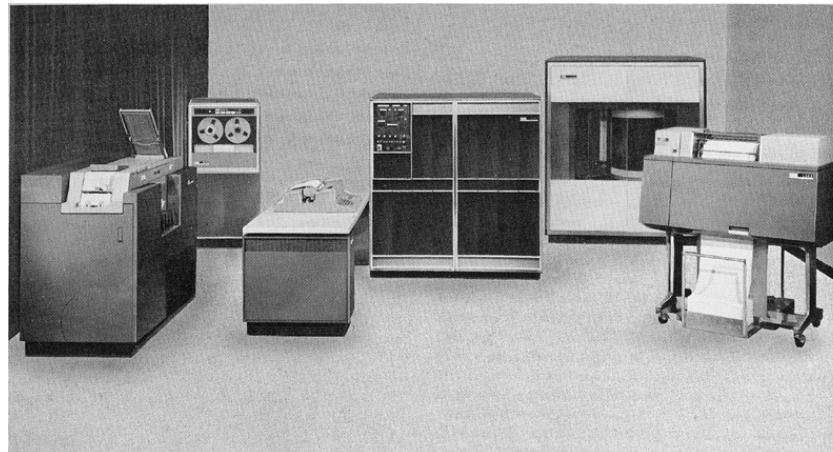
- Weitere Rechenanlagen:
- ZUSE Z22 (BRD)
  - D1/D2 (DDR)
  - Colossus (GB)
  - ENIAC (USA)
  - IBM 305 RAMAC

# Historische Entwicklung von Rechnern und Betriebssystemen

---

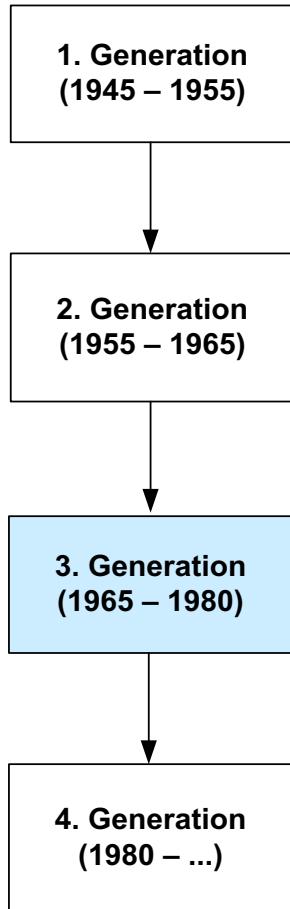


- Etwas komplexere Betriebssysteme
- Transistorencomputer
- Assemblersprachen
- Mainframes, Batchverarbeitung: Jobs hintereinander ausgeführt
- IBM 1401, 7094



IBM-1401-Anlage  
Quelle: IBM

# Historische Entwicklung von Rechnern und Betriebssystemen



PDP-11 von digital  
Quelle: Wikipedia

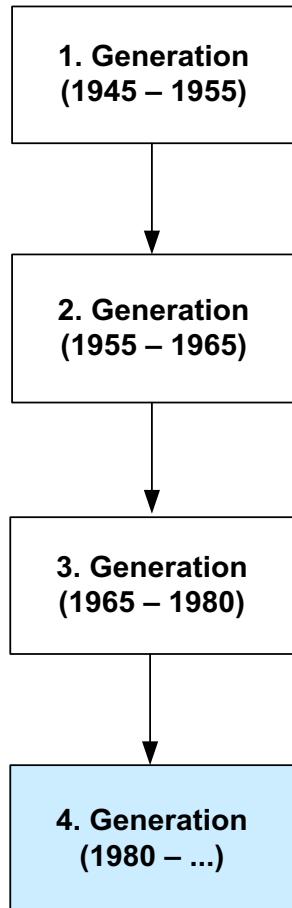
- Umfangreiche Betriebssysteme wie OS/360, BS1000, MULTICS, Unix
- Integrated Circuits
- Hochsprachen
- Mainframes, Multiprogramming, Timesharing (Mehrbenutzerbetrieb)
- IBM-Systeme, Siemens-Systeme, DEC PDP-11, ...



S/360-System  
Quelle: Wikipedia

# Historische Entwicklung von Rechnern und Betriebssystemen

---



IBM PC, Modell IBM 5150  
Quelle: IBM



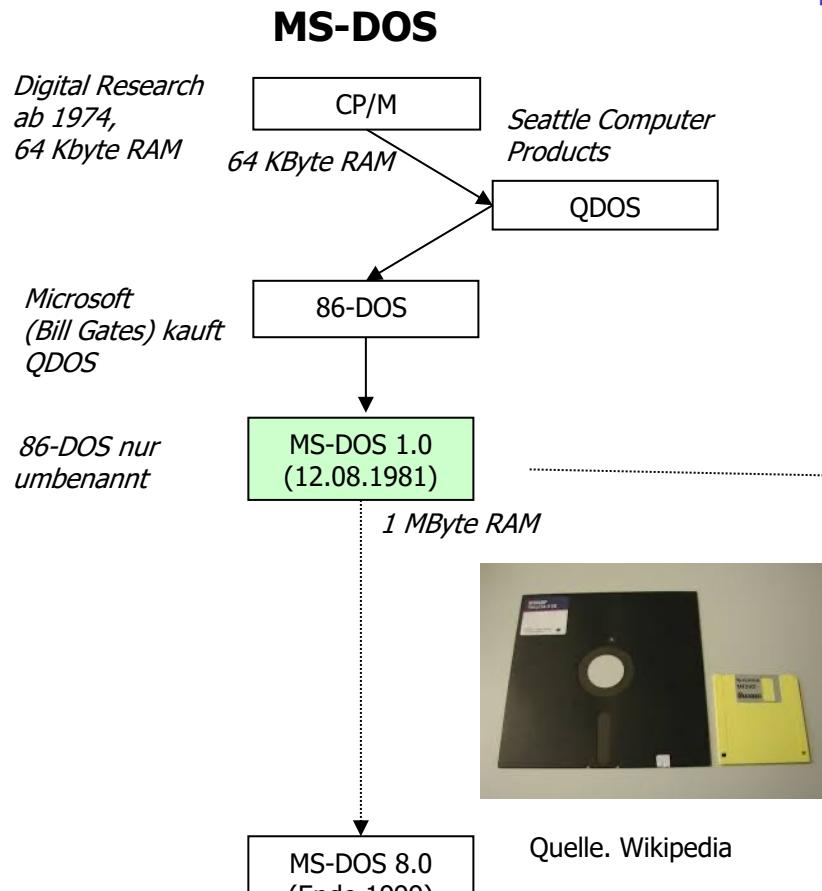
IBM zSeries  
Quelle: IBM



Enterprise Server von Sun  
Quelle: Sun Microsystems

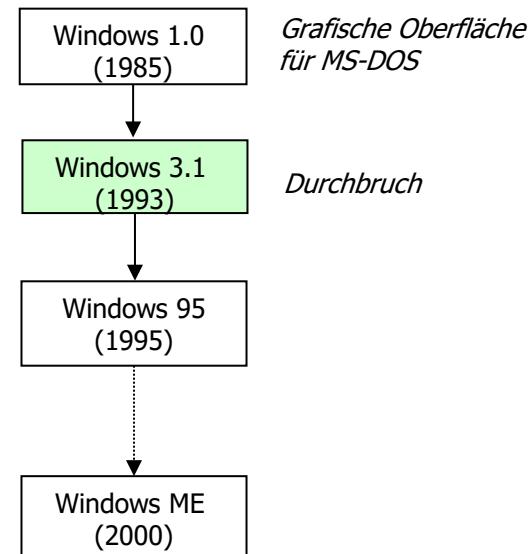
- Komplexe Betriebssysteme
- Large Scale Integration
- Objektorientierte Sprachen
- Deutliche Verbesserung des User Interface
- PCs, Workstations, Server, Mainframes, Verteilte Systeme
- MS-DOS, Unix, Windows, IBM-OS/390, z/OS, Mac OS X, Android ...

# Historische Entwicklung: Windows (1)



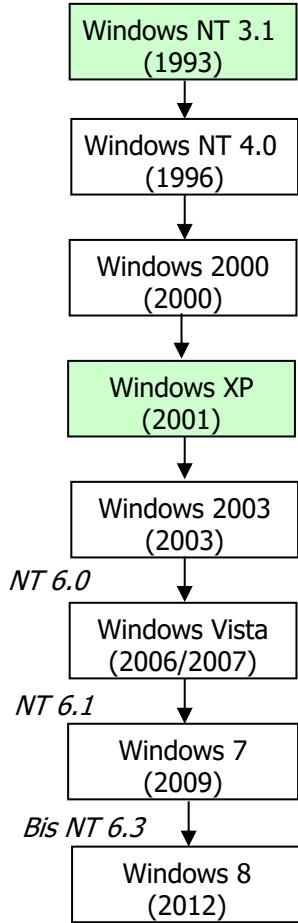
- **MS-DOS V1.0** wurde von IBM mit einem 8088-basierten IBM PC herausgegeben
  - Single-User-System
  - 8-Bit-Betriebssystem
- **Windows 1.0** war das erste graphische User-Interface für MS-DOS

## Windows

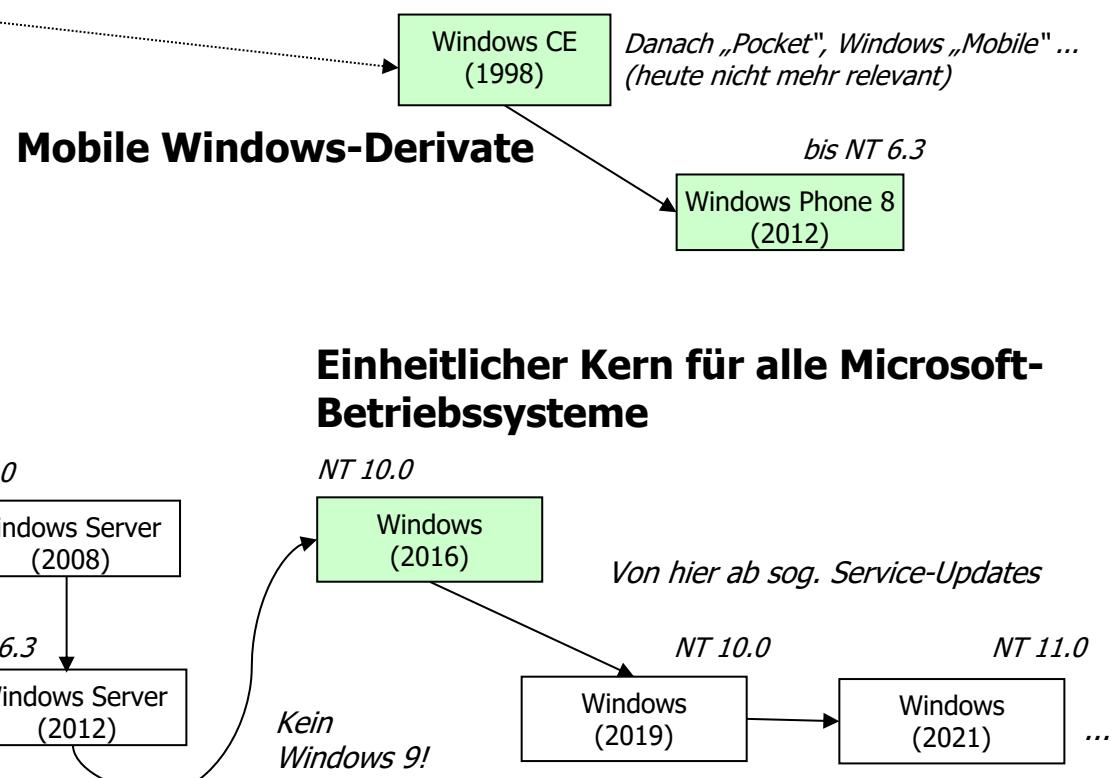


# Historische Entwicklung: Windows (2)

## Windows NT



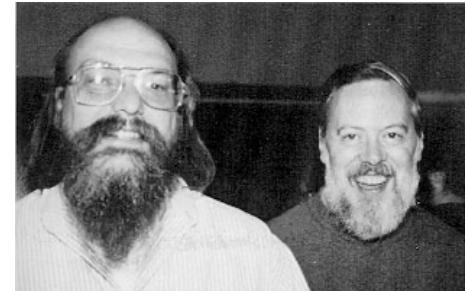
- Windows NT war eine **vollständige Neuentwicklung** von Microsoft
- Anfangs User-Interface von Windows 95
- Keine 100%-MS-DOS-Kompatibilität
- Bis heute setzt Windows auf diesem Kern auf



# Kurze Geschichte von Unix

---

- Unix entstand aus **MULTICS**  
(Multiplexed Information and Computing Service)  
→ ein nicht erfolgreiches System
- Erste Single-User Version von Unix in den Bell Labs  
auf einer PDP-7 von **Ken Thompson** (links) und  
**Dennis Ritchie** (rechts) entwickelt (1969)



# Kurze Geschichte von Unix

---

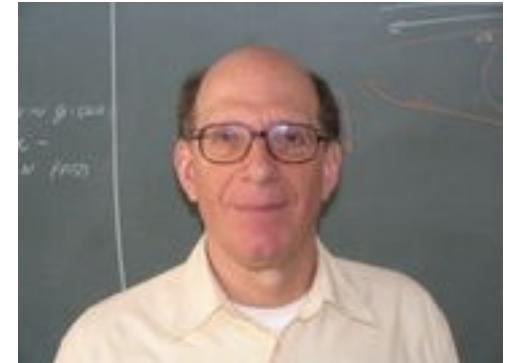
- Zwei inkompatible Hauptversionen entstanden
  - Die Berkeley University entwickelte das **BSD** (Berkeley Software Distribution)
    - Vorbild für Sun OS von Sun Microsystems
    - Heute gibt es viele Nachfolger: FreeBSD, NetBSD, OpenBSD, DragonFly BSD, macOS (sehr erfolgreich)
  - **System V** von AT&T (wechselte mehrfach den Besitzer)
- Weitere Unix-Derivate heute haben ihre Feinheiten: HP-UX, Sun Solaris, Sinix, Reliant Unix (Fujitsu Siemens), AIX (IBM),...

# Kurze Geschichte von Unix

---

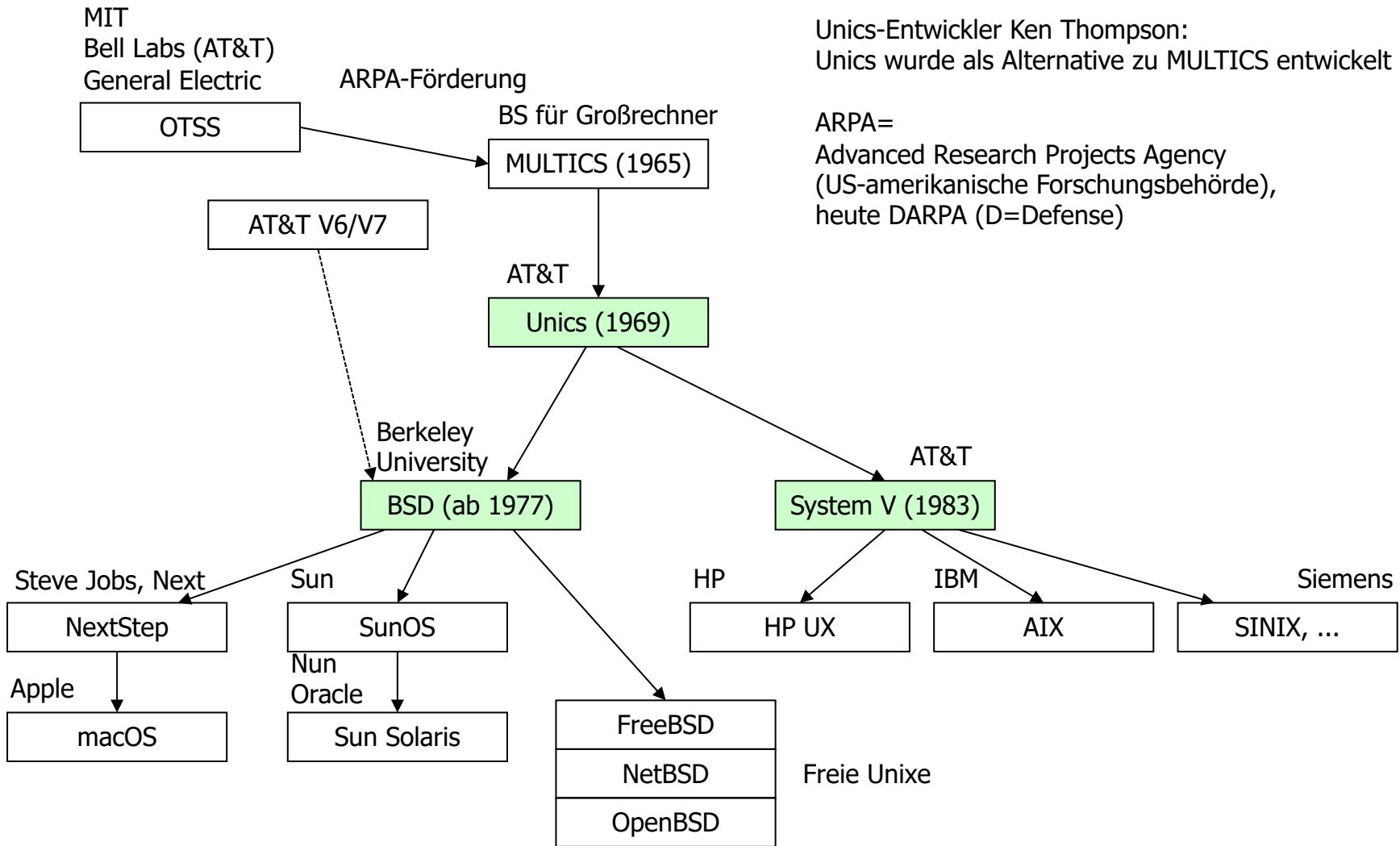
- Tanenbaum entwickelte 1987 einen kleinen Unix-Clone namens **MINIX**  
(ca. 12.500 Lines of Code)

- Heute: MINIX 3 als Forschungsprojekt für zuverlässige Betriebssysteme;  
Open-Source-Projekt: [www.minix3.org](http://www.minix3.org)



- Aus MINIX entstand **Linux** durch **Linus Torvalds** (ehemals finnischer Student) als Open Source Unix
  - Erfreut sich heute immer weiterer Verbreitung über Distributoren, die auch zusätzlich Service anbieten
  - Mischung aus System V, BSD und eigenen Erweiterungen

# Unix-Entwicklung



# Wichtigster Unix-Standard: POSIX

---

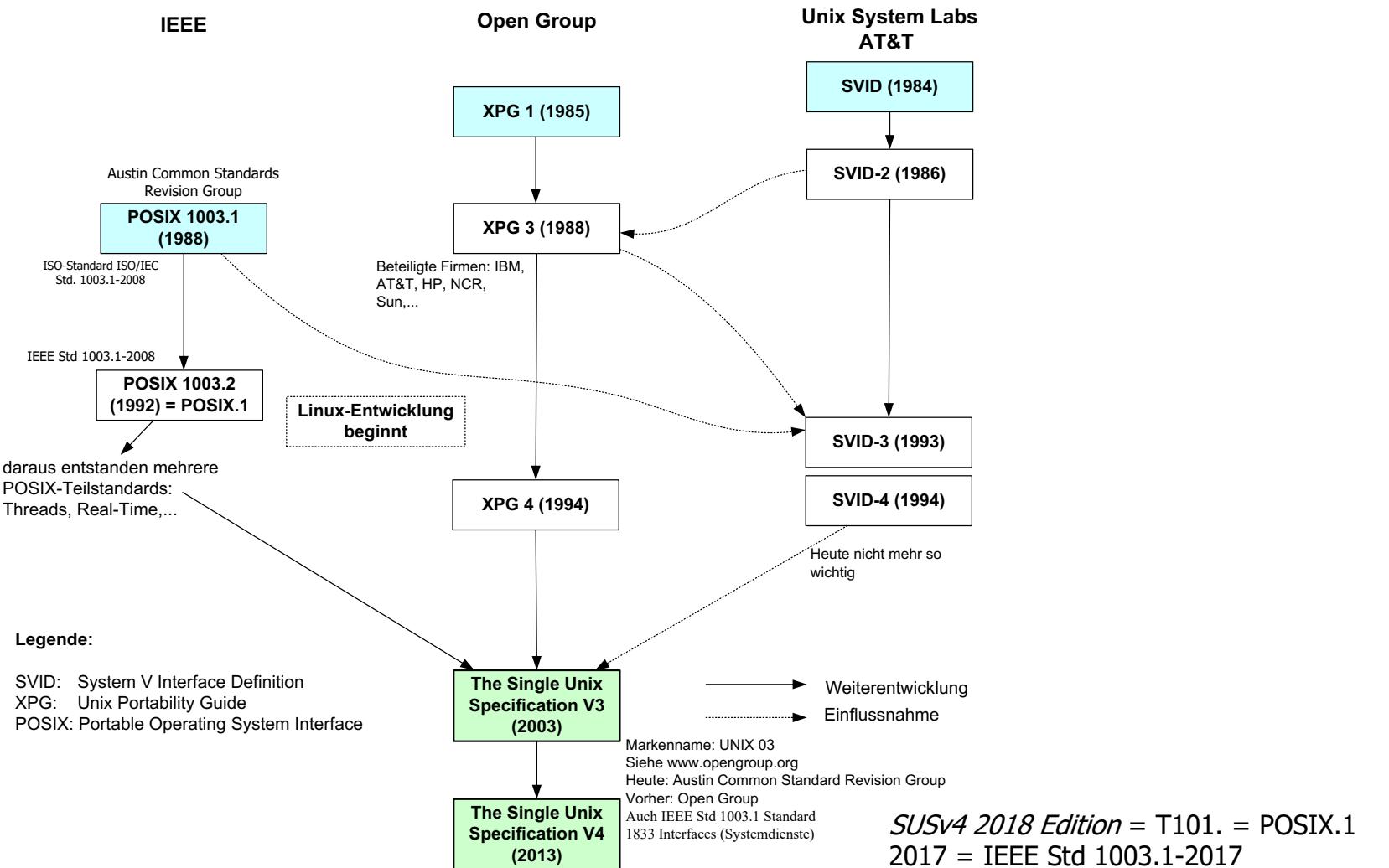
- **POSIX:** Portable Operation System Interface
  - Genaue Bezeichnung: ISO/IEC/IEEE 9945
- IEEE entwickelte diesen Standard schon Mitte bis Ende der 80er Jahre
- Wichtigste Standardisierungskomponenten: System Call Interface
  - Wird von fast allen Herstellern unterstützt (AIX, HP-UX, MINIX, macOS, ...)
  - Linux unterstützt POSIX nicht vollständig

Anwendungen

System Call Interface

Betriebssystem

# Exkurs: Unix-Normierung, historische Entwicklung



# Linux-Geschichte

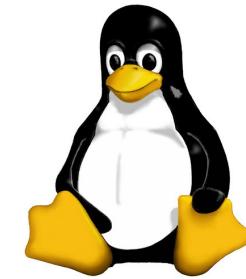
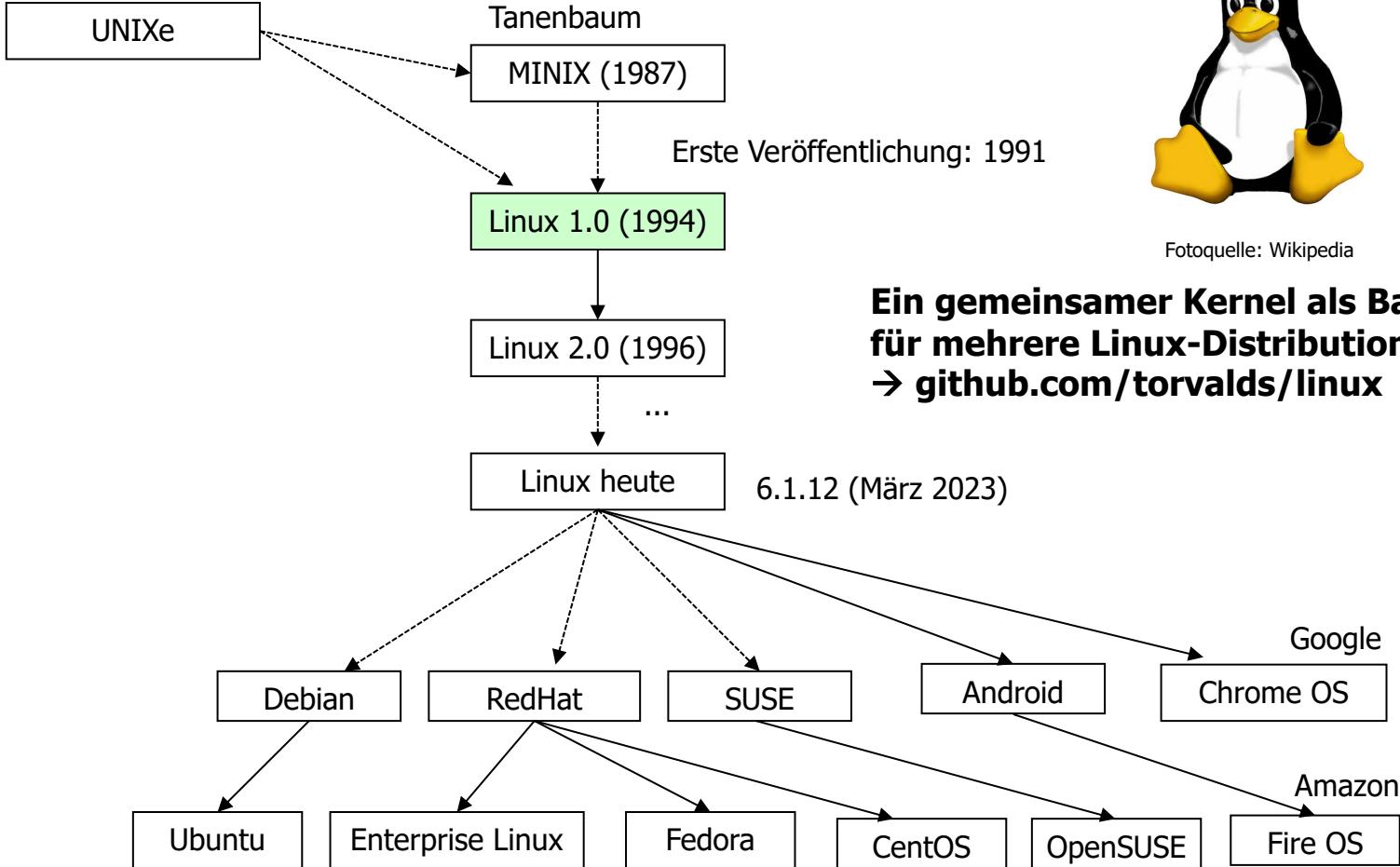
---

- 25. August 1991 (Linus Torvalds)
  - Ankündigung des ersten **Linux-Kernels** über Usenet-Posting
  - Anm.: Kernel = Betriebssystemkern
- Ende 1992 wurde die erste Version unter einer GNU General Public License gestellt
- Heute gibt es viele Distributionen
  - Red Hat → Fedora
  - Red Hat → Enterprise Linux
  - CentOS (Community Enterprise Operating System)
  - SUSE Linux → OpenSUSE
  - Debian Linux (Desktop-Betriebssystem)
  - Debian → Ubuntu
  - Android (Google → Open Handset Alliance) (Oktober 2008)
  - Android → Fire OS (Amazon), ...



Fotoquelle: Wikipedia

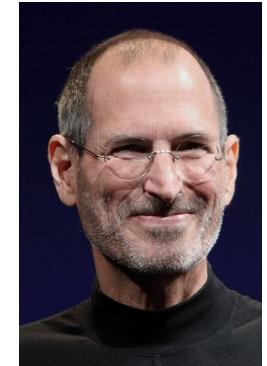
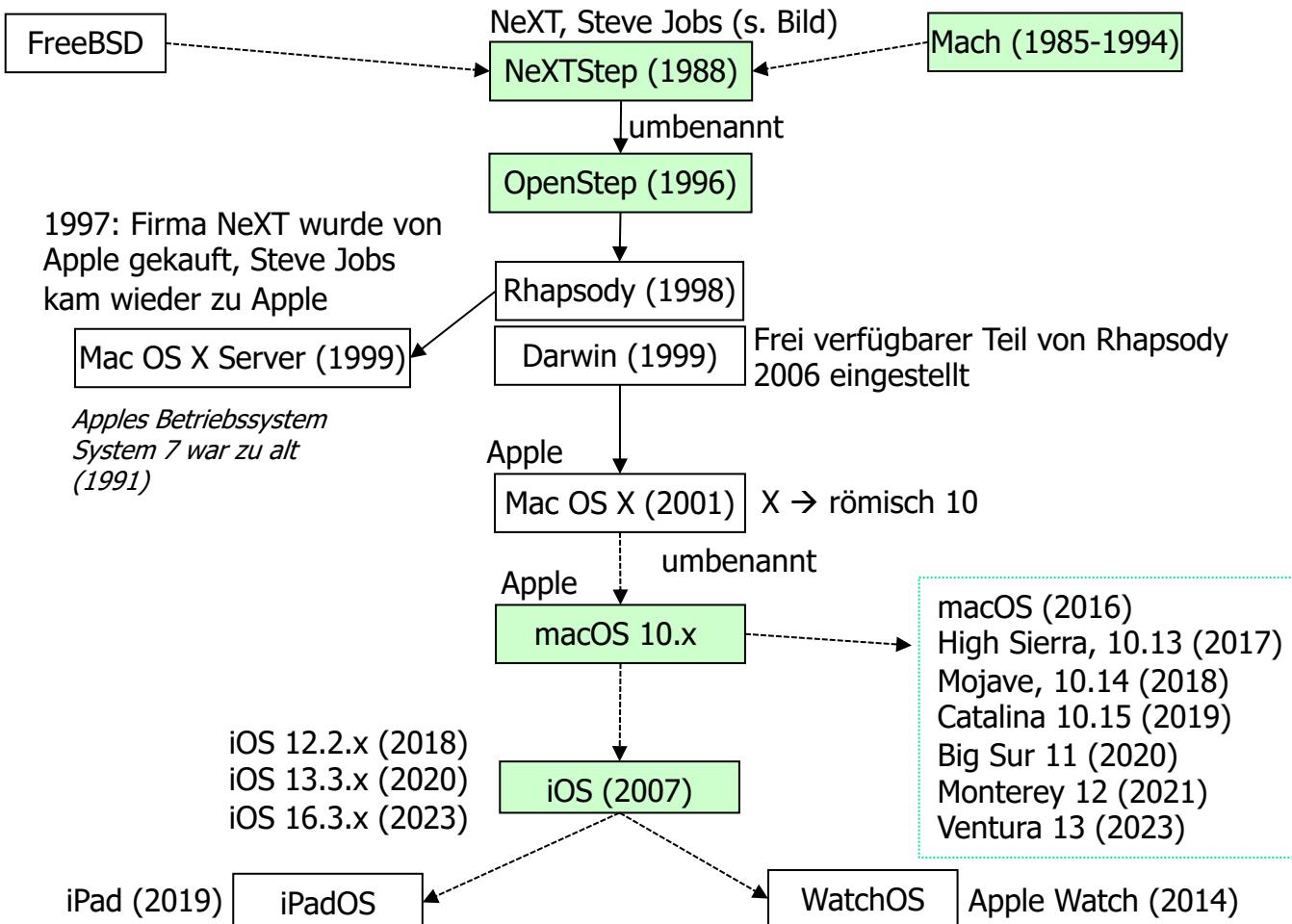
# Linux-Entwicklung



Fotoquelle: Wikipedia

**Ein gemeinsamer Kernel als Basis  
für mehrere Linux-Distributionen**  
→ [github.com/torvalds/linux](https://github.com/torvalds/linux)

# macOS-Entwicklung



Fotoquelle: Wikipedia