

1.1 Anwendung/Einführung

1.1.1 Was ist ein Betriebssystem, was macht ein Betriebssystem?

Die Definition von Betriebssystem (= *operating system*, auch OS, BS, Betriebsprogramm, Systemsoftware) in einem Satz ist schwierig. Am einfachsten kann man beschreiben, was es *nicht* ist. Man kann es abgrenzen von der

- Hardware (alle nicht programmierbaren Bausteine),
- Entwicklungssoftware (Programme zum Schreiben neuer Programme),
- Anwendersoftware (Programme, die vom Anwender eines Systems willentlich aufgerufen und beendet werden).

Eine Zusammenstellung der Aufgaben eines Betriebssystems kann am ehesten definieren, was ein Betriebssystem ist:

- a) Betriebsmittel (=Ressourcen, z.B. Prozessor, RAM, Platz auf der Festplatte, Netzwerkkarte, Drucker) bereitstellen, verwalten und zuteilen,
- b) Für Programme (und Programmierer) eine Möglichkeit des Hardware-Zugriffs ohne genaue Kenntnis der Hardware zur Verfügung stellen (Hardware-Abstraktion),
- c) Prozesse (Programme) auf Benutzerwunsch starten und beenden,
- d) Informationen und Eingriffsmöglichkeiten für Benutzer zur Verfügung stellen.

Manche Betriebssysteme erfüllen nicht alle diese Aufgaben, so hat z.B. MS-DOS nur eingeschränkte Möglichkeiten der Prozessor-Zuteilung. Ebenso gibt es Systeme, die überhaupt kein Betriebssystem brauchen, da sie entweder nur ein Programm dauerhaft betreiben (z.B. *embedded systems*) oder ein Mini-Betriebssystem bereits im Programm enthalten ist (z.B. frühe Versionen des MS-Flugsimulators).

1.1.2 Warum soll man etwas über Betriebssysteme lernen?

Erstes Argument: Windows wird vermutlich nicht das letzte Betriebssystem der Geschichte sein.

- 1965–1976: Vorherrschende Systeme: herstellerspezifisch (meist IBM)
- 1976–1984: Vorherrschende Systeme: CP/M (Digital Research) und Apple-DOS
- 1984–1995: Vorherrschende Systeme: PC-DOS (IBM) bzw. MS-DOS
- 1995–2000: Vorherrschende Systeme: Windows 95/98
- 2000–2024: Vorherrschende Systeme: Windows XP/7/8/10/11, Android, iOS

Zweites Argument: PCs werden vermutlich nicht die letzten Massen-Computer der Geschichte sein.

- ab 1940: Einzelrechner in Universitäten und Forschungsinstituten
- ab 1955: Kommerzielle Großrechner für Großbetriebe
- ab 1965: Kleine Großrechner für mittelständische Betriebe
- ab 1970: Mini-Computer (CPU auf 1 Platine, Computer in 1 Rack)
- ab 1975: Mikro-Computer (CPU auf 1 Chip, Computer in 1 Einschub)
- ab 1980: Mikro-Controller (Komplettrechner auf 1 Chip) für *embedded systems*

- ab 2000: Smartphones (CPU bedient Datenübertragung und Anwender-Software)
- ab 2010: Smartwatches, Wearables, ...

Parallel zum Aufkommen neuer Rechnertypen wurden andere Typen verdrängt (in der Tabelle nicht ersichtlich), die im Vergleich zu groß und zu teuer waren, deren (ebenfalls gesteigerte) Leistungsfähigkeit aber nicht mehr gebraucht wurde.

PCs können verdrängt werden durch

- Neue PCs mit modernerer, daher inkompatibler Hardware (Wegfall von ISA, PCI und diversen Peripherie-Schnittstellen weist in diese Richtung),
- System-on-a-Chip (SoC) zu geringem Preis mit On-Chip-Festplatten-Ersatz,
- massenhaften Einsatz billiger *embedded systems* für ursprüngliche PC-Aufgaben (Routing, DSL-Einwahl, Monitor-Ansteuerung, Druckerserver, Bilderrahmen).

Es lassen sich auch Argumente finden, warum man nichts über Betriebssysteme zu lernen braucht:

- Betriebssystem-Kurse verkommen leicht zu einfachen Produktschulungen. Diese sind aber Aufgabe der Betriebssystem-Lieferanten.
- Durch neue Versionen ändert sich der Umgang mit Betriebssystemen laufend. Das Betriebssystem-Wissen veraltet zu schnell.
- Viele Betriebssysteme (z.B. Windows) lassen keinen Einblick in Interna zu; insofern kann man kein tieferes Wissen erlangen, sondern wird bewusst dumm gehalten (und fährt oft gut damit).
- Tipps- und Tricks-Literatur existiert in Massen und hilft bei Problemen schnell weiter.
- Allgemeine Theorie über Betriebssysteme ist oft abstrakt und hilft bei praktischen Problemen nicht weiter;
- zudem hinken aktuelle Betriebssysteme hinter dieser Theorie oft um viele Jahre hinterher, so dass sich dieses Wissen oft erst nach vielen Jahren auszahlt (wenn überhaupt).

1.1.3 Warum hängen Betriebssysteme und Netzwerktechnik zusammen?

Pro-Argumente:

- Die Schichten 1–3 des OSI-Modells sind meistens im Betriebssystem eingebaut (implementiert),
- die Verwaltung der Netzwerkeigenschaften des Systems (IP-Adresse, DNS-Name usw.) werden mit Dienstprogrammen des Betriebssystems verwaltet.

Contra-Argumente:

- Die Schicht 7 des OSI-Modells wird in Client- und Serverprogrammen selbsttätig abgearbeitet,
- Netzwerkprotokolle wie z.B. TCP/IP sind genormt und selbst ihre Implementierung (Verwirklichung) ist überall ähnlich.

1.1.4 Entwicklung von Unix und Linux

- 196X - Multics-Entwicklung
- 1969 - erste UNIX-Version für DEC PDP-7, in Assembler
- 1971 - Übertragung auf DEC PDP-11, in "B"; Email-Dienst
- 1973 - Übertragung nach "C" (Ziel: Portabilität)
- 1975 - Version 6 (I.7)
- 1977 - Mikrorechner-Portierung (Prozessor: ID8/32)
- 1979 - Version 7 (I.7)
- 1979 - Ableger: MS-Xenix für 286er, daraus SCO-UNIX
- 1979 - Ableger: Univ.Cal.Berkeley: BSD 4.1, daraus SUN-OS/Solaris, ..., Ultrix - Neuigkeit: TCP/IP-Kommunikation wird vereinfacht durch sog. Sockets
- 1979 - Ableger: HP-UX, Ultrix, ...
- 1981 - System III (III.0/1)
- 1983 - System IV (V.0)
- 1984 - System V, Version 2 (V.2)
- 1985 - System V, Version 3 (V.3) - Kommunikation über Message Queues, Semaphore, Shared Memory
- 1987 - System V, Version 3.1 (V.3.1)
- 1989 - System V, Version 4 (V.4)
- 1991 - Ausgliederung von UNIX durch AT&T (Lizenz- und Monopolfragen)
- 1987 - Minix (Unix auf Intel-Mikros 8086 und 80286) A.S. Tanenbaum, Univ. Amsterdam
- 1990 - NeXT Computer mit NextStep/Mach als Betriebssystem - Microkernel und graphische Oberfläche
- 1991 - Linux (Unix auf Intel 80386) 0.01 - installiert unter Minix (Finnland)
- 1994 - Linux 1.0
- 1997 - Linux 2.0 (modularer Kernel)
- 2000 - Mac OS X, aufbauend auf Mach-Microkernel (und BSD)
- 2008 - Android, aufbauend auf Linux-Kernel
- 2011 - Linux 3.0
- 2015 - Linux 4.0
- 2019 - Linux 5.0
- 2022 - Linux 6.0
- heute - Linux 6.11
- parallel dazu: Entwicklung mehrerer BSD-Linien

1.1.5 Normung

In den neunziger Jahren existierten viele verschiedene UNIX-Systeme auf dem Markt (HP-UX, AIX, Sun-OS/Solaris, Irix, System V, BSD).

- ANSI-C – erfüllten alle
- Art der Parameter-Übergabe durch ANSI-C ans System – war überall gleich
- Kleinster Nenner: Unix-Kern, genormt durch IEEE Posix
- AT&T System V – viele weitere Funktionen durch einen Wettbewerber
- BSD – einige ähnliche Funktionen durch einen anderen Wettbewerber
- X/Open XPG – NLS und weitere Optionen durch andere Wettbewerber
- Normung 90er: ANSI-C < POSIX 1002.1 < POSIX < AT&T System V < X/Open

Heute dagegen hat eine Konzentration auf Linux (und einige andere Unix-Systeme wie BSD, Solaris und MAC-OS X) stattgefunden:

- Normung heute: ANSI-C < POSIX < Linux
- Problem heute: Linux - mehrere Distributionen, jeweils funktional mächtig,
- zum großen Teil besteht Binär-Kompatibilität,
- jedoch unterschiedliche Oberflächen (KDE, GNOME, Unity u.v.a)
- und unterschiedliche Administration (aptitude, YaST usw.).

1.1.6 Linux - Einordnung

Ein Aspekt für die Unterteilung von Betriebssystemen ist die Gleichzeitigkeit; davon werden die technischen Möglichkeiten des Systems beeinflusst:

- Mehrbenutzersystem (Multiuser-): mehrere Benutzer gleichzeitig
- Mehrprozesssystem (Multitasking-): mehrere Prozesse gleichzeitig

System	Multiuser	Singluser
Multitasking	Unix, Linux, Mach, Windows ab XP	Windows 95,98, NT, 2k
Singletasking	—	PC-DOS, Oberon-System, GEM

Ein anderer Aspekt ist die Frage nach den rechtlichen Möglichkeiten im Umgang mit dem Betriebssystem: Kann der Quelltext eingesehen oder sogar verändert werden? Werden Nutzungsgebühren für jeden Computer (Windows) oder sogar für jeden Benutzer (SCO-Unix) erhoben? Ist eine bestimmte Gruppe für Änderungen verantwortlich, oder kann jeder Änderungen einbringen?

- MS-Windows, IBM OS/2, ... : Source nicht-öffentlich, nicht veränderbar
- Unix: Source nicht-öffentlich, aber veränderbar
- Free BSD: Source öffentlich und veränderbar; geschlossene Entwicklergruppe
- Linux: Source öffentlich und veränderbar; offene Entwicklergruppe

Ein dritter Aspekt ist die Verfügbarkeit von Betriebssystemen auf unterschiedlicher Hardware. Betriebssysteme, bei denen dies berücksichtigt wurde, lassen sich erfahrungsgemäß schneller an neue Hardware anpassen. Bei kommerziellen Betriebssystemen dient die Unterstützung spezieller Hardware oft auch firmen-strategischen Zielen.

- DOS - nur Intel/AMD
- Unix - viele
- Linux - viele: Intel/AMD, Alpha, 68k, PowerPC u.v.a.
- Windows NT - Intel/AMD, Alpha
- Windows XP,7,8,10,11 - nur Intel/AMD

1.1.7 Linux - Betriebsarten

Eine weitere Einordnung von Betriebssystemen ist die Einteilung nach den möglichen Betriebsarten:

- Stapelbetrieb (= *batch*) - Aufträge werden in einer festgelegten Reihenfolge abgearbeitet. Nach Abarbeitung holt der Benutzer die Ergebnisse ab. Geeignet für Verarbeitung großer Datenmengen.
- Dialogbetrieb (=interaktiver Betrieb) - Benutzer gibt Auftrag ab und erwartet sofort (innerhalb weniger Sekunden) eine Antwort.
- Echtzeitbetrieb - Hardware-Anfragen werden sofort (innerhalb einer festgelegten und garantierten Antwortzeit, oft im Mikrosekundenbereich) bearbeitet.

Die Einordnung von Linux und Windows innerhalb dieses Aspektes ist in der folgenden Tabelle dargestellt:

Betriebsart	Linux	Windows
Stapelbetrieb	möglich	mit Erweiterungen möglich
Dialogbetrieb	möglich	möglich
Echtzeitbetrieb	ab 6.12 ohne Erweiterungen möglich	mit Erweiterungen möglich

Für Echtzeitbetrieb existieren sowohl eigene Betriebssysteme (RTOS-UH von der Uni Hannover) als auch Erweiterungen zu Linux (diverse) und zu Windows (Fa. Beckhoff).

1.1.8 Linux-Aufbau

Die vier genannten typischen Aufgaben eines Betriebssystems

- Betriebsmittelverwaltung (CPU-Zeit, RAM usw.),
- Hardware-Abstraktion für Programme,
- Programm-Start-Möglichkeit für Benutzer,
- Informationen und Hilfen für Benutzer,

finden sich auch im Aufbau von Linux wieder (siehe Abbildung 1). Der Kernel dient

- zusammen mit den Treibern zur Abschirmung der Hardware-Eigenheiten,
- zur Organisation der Massenspeicher in Form eines Dateisystems (Dateien und Verzeichnisse),
- mit Zugriffsschutzmechanismen (Dateiberechtigungen und Dateieigentum),
- und zur Organisation der Prozessor- und Speicherressourcen.

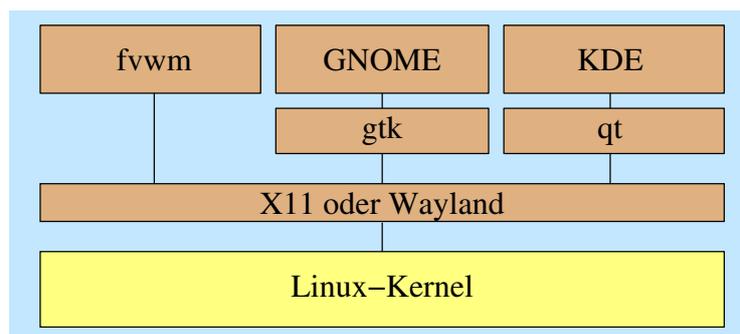


Abbildung 2: Aufbau von Linux mit GUI