

Betriebssysteme

Stoffübersicht Betriebssysteme

- Einleitung
 - Prozessverwaltung
 - Memoryverwaltung
 - Dateisysteme
 - Ein- Ausgabesteuerung
-
- Theorie + Systemprogrammierung (UNIX + C)

Einleitung

- Aufgabe von Betriebssystemen
- Historische Entwicklung von Betriebssystemen
- Unterschiedliche Arten von Betriebssystemen
- Komponenten und Konzepte von Betriebssystemen
 - Prozesse und Threads
 - Speichermanagement
 - Dateisysteme
 - Konzepte der Ein- und Ausgabe

Aufgaben von Betriebssystemen

Aufgaben von Betriebssystemen

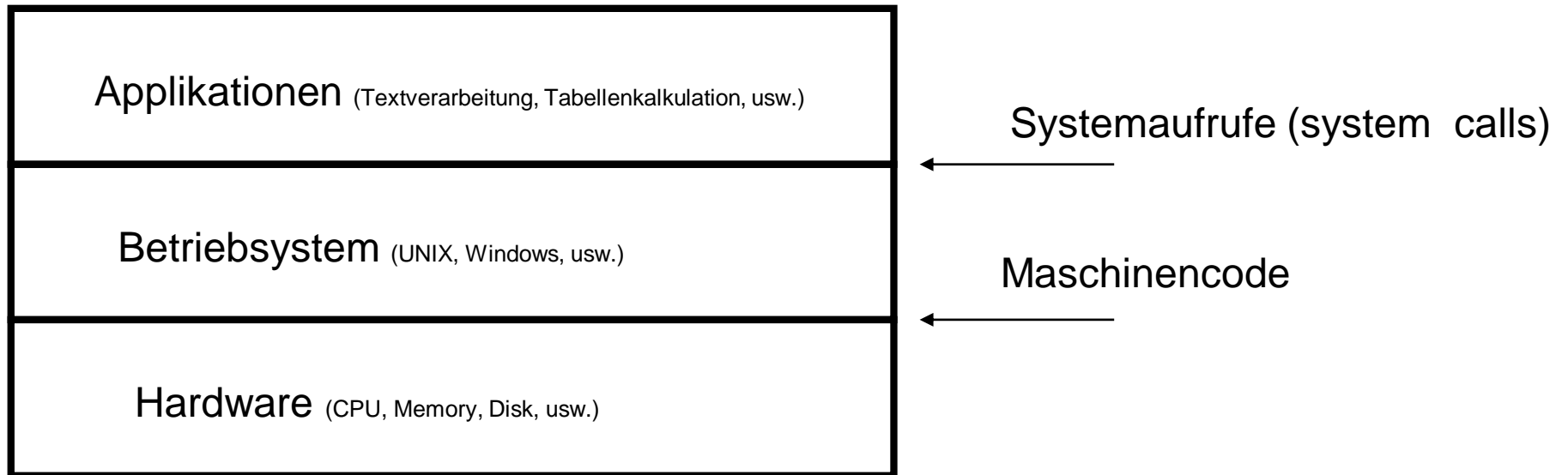
- Ein Betriebssystem ist ein Computerprogramm (Software) mit folgenden grundsätzlichen Funktionen:
 - Bereitstellen einer **Abstraktionsschicht** der Betriebssystem-Komponenten für Programmierer
 - „**Erweiterte Maschine**“
 - Verwaltung und Schutz von **Systemressourcen**
 - beispielsweise in einer Mehrbenutzerumgebung
 - Bereitstellung einer Umgebung zur **Ausführung** von Anwendungsprogrammen

Bereitstellung einer Abstraktionsschicht

- **Abstraktion** von detailliertem Verhalten der zugrunde liegenden Komponenten auf möglichst hohem Niveau
- „Verstecken“ von
 - realer Hardware-Eigenschaften vor dem Programmierer
 - Systemdiensten und Systemfunktionen
- **Ziel: Leichte Programmierbarkeit/Portierbarkeit und Sicherheit**
 - **Beispiel:** Datenspeicherung auf einem Datenträger
 - Entlastung des Programmierers von komplizierten Details wie Einschalten des Motors, Bewegung des Plattenarms etc.
 - => **Hardware-Abstraktion**
 - Einfache Programmierschnittstelle zum Schreiben und Lesen einer Datei (open(), read(), write()) unabhängig von dem verwendeten Dateisystem (FAT, NTFS, ext3, etc)
 - => **Abstraktion von Systemdiensten und Systemfunktionen**

Schichtenmodell

Benutzer



Bereitstellung einer Abstraktionsschicht

- Das Betriebssystem stellt dem Programmierer eine abstrakte Programmierschnittstelle zur Verfügung
- Das Betriebssystem bietet einen Satz von Kommandos (**Systemaufrufe** / **system calls**):
 - zum Zugriff auf E/A-Geräte
 - z.B. Lesen und Schreiben einer Datei
 - zur Prozesskontrolle
 - z.B. Starten und Stoppen eines Anwendungsprogramms
 - zur Speicherverwaltung
 - Überwachung und Kontrolle von Ressourcen

Verwaltung von Systemressourcen

- Verwaltung aller Bestandteile eines komplexen Systems
- Bestandteile sind z.B.
 - Prozessoren und Rechenzeit, Speicher, Uhren, Platten, Terminals, Magnetbandgeräte, Netzwerkschnittstellen, Drucker etc.
- Bestandteile eines Rechnersystems bezeichnet man auch als **Betriebsmittel**.
- Betriebssystemaufgabe:
 - Geordnete und kontrollierte **Zuteilung von Betriebsmitteln** an konkurrierende Prozesse / Benutzer.
 - Schutz des Systems und der individuell zugeordneten Betriebsmittel
 - **Beispiel**: Ein defektes Programm sollte nicht das Verhalten anderer Programme oder das ganze System beeinträchtigen können.
 - **Beispiel**: Schutz von Dateien auf Mehrbenutzersystemen

Betriebssystem als Ressourcenmanager

- Aufgaben beim Verwalten von Systemressourcen:
 - „**Gerechte**“ Zuteilung von gemeinsam genutzten Betriebsmitteln
 - **Auflösung von Konflikten** bei der Betriebsmittelanforderung
 - **Schutz** verschiedener Benutzer gegeneinander
 - z.B. Zugriffskontrolle bei Dateien
 - **Effiziente** Verwaltung von Betriebsmitteln
 - **Protokollieren** der Ressourcennutzung
 - **Abrechnung** der Betriebsmittelnutzung
 - **Fehlererkennung, Fehlerbehandlung**
 - Hardware (Gerätefehler) und Software (Programmfehler)
 - Grundsätzliche **Zugriffskontrolle** zum System
- Ressourcenverwaltung in zwei Dimensionen:
 - **Zeit**: Verschiedene Benutzer erhalten Betriebsmittel nacheinander.
 - **Raum**: Verschiedene Benutzer erhalten verschiedene Teile einer Ressource.

Erweiterbarkeit und Entwicklungsfähigkeit

- **Änderungen** des Betriebssystems erforderlich durch
 - Neue Hardware
 - Neue Protokolle etc.
 - Neue Anforderungen der Benutzer
 - Korrekturen (z.B. Schließen von Sicherheitslöchern)
- ▶ Betriebssystem sollte
 - **modular** und **klar strukturiert** aufgebaut sein
 - gut **dokumentiert** sein.

Historische Entwicklung

Historische Entwicklung

- **Verschiedene Entwicklungsstadien:**
 - Serielle Systeme
 - Einfache Stapelverarbeitungssysteme
 - Mehrprogrammfähige Stapelverarbeitungssysteme
 - Timesharing-Systeme
 - (Systeme mit Graphischen Benutzeroberflächen)
 - Netzwerkbetriebssysteme
 - Verteilte Betriebssysteme

Historische Entwicklung

- **Serielle Systeme (1945-1955)**
 - Betrieb von Rechnern ohne Betriebssystem
 - Programmierung von Rechnern durch
 - Einschieben von Steckkarten
 - Später: Einlesen von Lochkarten
 - Zuteilung von Rechenzeit durch Reservierung mit Hilfe Papieraushang

Historische Entwicklung

- **Einfache Stapelverarbeitungssysteme, 1955 für IBM**
 - Unterscheidung zwischen
 - **Programmentwickler** und **Operateure**, die Rechner („Mainframes“) betrieben
 - Sammlung von Programmen (Jobs) auf Lochkarten
 - Einlesen der gesammelten Jobs durch kleine Rechner (z.B. IBM 1401), Speichern auf Band
 - Abarbeitung des Bandes durch Hauptrechner (z.B. IBM 7094):
 - Lese ersten Job ein
 - Führe ersten Job aus
 - Ausgaben auf zweites Band
 - Dann weiter mit zweitem Job etc.
 - Serielle Abarbeitung der Jobs gesteuert durch kleines Softwareprogramm, genannt **Monitor**.
 - Ausgabe der Ergebnisse auf Band durch kleinen Rechner (z.B. IBM 1401)
 - **Fortschritt**: Eingabe von Lochkarte, Rechnen, Ausgabe auf Drucker **parallel**.

Historische Entwicklung

- **Mehrprogrammfähige Stapelverarbeitungssysteme, ab 1965**
 - Verarbeitung durch einzigen Rechner (IBM 360 + Nachfolger)
 - **Spooling** (**S**imultaneous **P**eripheral **O**peration **O**n **L**ine):
 - Einlesen von Jobs auf Lochkarten, Speichern auf Platte
 - Nach Beenden des aktuellen Jobs: Laden eines neuen Jobs von Platte + Ausführung
 - Mehrprogrammfähigkeit bzw. Multiprogramming

Historische Entwicklung

- Mehrprogrammfähigkeit bzw. Multiprogramming
 - **Beobachtung:**
 - Viel Rechenzeit wird verschwendet durch Warten der CPU auf Beendigung von Ein- / Ausgabeoperationen.
 - Bis zu 90% der Zeit verschwendet bei Hochleistungs-Datenverarbeitung.
 - **Idee:**
 - Führe aus Effizienzgründen Jobs nicht streng sequentiell aus.
 - Aufteilung des Speichers in mehrere Bereiche
 - Eigene Partition pro aktiven Job
 - Wartezeiten auf Beendigung von Ein- / Ausgabeoperationen genutzt durch Rechenzeit für andere Jobs.

Historische Entwicklung

- **Timesharing-Systeme**

- **Nachteil** mehrprogrammfähiger Stapelverarbeitungssysteme:
- kein interaktives Arbeiten mehrerer Benutzer möglich
 - Abhilfe durch Timesharing-Systeme
 - Variante des Multiprogrammings
 - Online-Zugang zum System für alle Benutzer
 - Mehrprogrammbetrieb mit **schnellem Umschalten** von Benutzer zu Benutzer.
- **Idee:**
 - Interaktives Arbeiten eines Benutzers erfordert nicht die komplette Rechenzeit eines Rechners.
 - Bei **schnellem Umschalten** bemerkt der Einzelnutzer nicht, dass er die Maschine nicht für sich allein hat.

Historische Entwicklung

- **Systeme mit Graphischen Benutzeroberflächen**
 - **GUI** (Graphical User Interface)
 - Geht zurück auf D. Engelbart, Stanford Research Institute, 60er Jahre
 - Elemente: Fenster, Icons, Menüs, Maus
 - Übernommen durch
 - Apple Macintosh
 - Später durch **Microsoft Windows**
 - 1985-1995: Graphische Umgebung, aufsetzend auf MS-DOS
 - Windows95, ..., WindowsXP, Windows Vista, Windows 7
 - Betriebssystem und GUI stark miteinander verschränkt
 - **UNIX / LINUX:**
 - GUI als Aufsatz auf Betriebssystem
 - **X-Window-System** (M.I.T.): Grundlegende Funktionen zur Fensterverwaltung
 - Komplette GUI-Umgebungen basierend auf X-Windows: z.B. **Motif**

Historische Entwicklung

- **Netzwerkbetriebssysteme und verteilte Betriebssysteme**
 - **Netzwerkbetriebssysteme:**
 - Benutzer kennt mehrere vernetzte Rechner
 - Einloggen auf entfernten Rechnern möglich
 - Datenaustausch möglich
 - Auf Einzelrechnern: Lokales Betriebssystem, lokale Benutzer
 - Netzwerkbetriebssystem = „normales Betriebssystem mit zusätzlichen Fähigkeiten“
 - **Verteilte Betriebssysteme:**
 - Mehrere vernetzte Rechner
 - Erscheinen Benutzern wie Einplatzsystem
 - Datenspeicherung und Programmausführung verteilt auf mehreren Rechnern
 - Verwaltung **transparent** und **effizient** (?) durch Betriebssystem
 - **Probleme:** **Nachrichtenverzögerungen**, **Datenkonsistenz**

Betriebssystemvarianten

Betriebssystemvarianten

- Mainframe-Betriebssysteme
 - Betriebssysteme für Großrechner
 - Sehr hohe Ein-/Ausgabebandbreite
 - Viele Prozesse gleichzeitig mit hohem Bedarf an schneller E/A
 - 3 Arten der Prozessverwaltung
 - Batch-Verfahren: Erledigung umfangreicher Aufgaben ohne Benutzerinteraktion
 - Transaktionsverfahren: Große Anzahl kleiner Aufgaben von verschiedenen Nutzern
 - Zeitaufteilungsverfahren: Quasi-parallele Durchführung vieler Aufgaben durch mehrere Benutzer
 - Bsp.: IBM OS/390

Betriebssystemvarianten

- Server-Betriebssysteme
 - Betriebssysteme für sehr große PCs, Workstations oder auch Mainframes
 - Viele Benutzer über Netzwerk bedient
 - Zuteilung von Hard- und Softwareressourcen
 - Bsp.: Unix, Linux, Solaris, Windows, usw.
- PC-Betriebssysteme
 - Betriebssysteme für Personalcomputer
 - Meist nur 1 Benutzer (oder wenige über Netzwerk)
 - Mehrere Programme pro Benutzer „quasi-parallel“
 - Bsp.: Windows , Linux

Betriebssystemvarianten

- Echtzeit-Betriebssysteme

- Betriebssysteme zur Steuerung maschineller Fertigungsanlagen etc.
- Einhalten harter Zeitbedingungen, gefordert ist
 - nicht „im Durchschnitt schnell“, sondern
 - Abschließen von Operationen in fest vorgegebenen Zeitintervallen (in jedem Fall)
- Bsp.: VxWorks, QNX, OSEK-OS

- Betriebssysteme für Eingebettete Systeme

- Eingebettete Systeme = „Computer, die man nicht unmittelbar sieht“
- z.B. in Fernseher, Mikrowelle, Mobiltelefon, KfZ
- Meist Echtzeitanforderungen
- Wenig Ressourcen:
 - Kleiner Arbeitsspeicher
 - Geringer Stromverbrauch
- Bsp.: WindowsCE, Symbian, Linux

Zusammenfassung

- Ein Betriebssystem ist ein Softwareprogramm.
- Liefert eine abstrakte Schnittstelle zum Rechner.
- Verwaltet und sichert Systemressourcen.
- Historische Entwicklung in mehreren Stadien.
- Verschiedene Arten von Betriebssystemen aufgrund verschiedener Anforderungen in unterschiedlichen Anwendungsgebieten.
- Moderne Betriebssysteme sind in der Regel Timesharing-Systeme mit vielen zusätzlichen Eigenschaften.