

Betriebssysteme - Grundlagen

by

Dr. Günter Kolousek

Begriffe

- ▶ Betriebssystem (BS), Operating system (OS)
- ▶ Hardware (HW), Software (SW)
- ▶ Betriebsmittel oder Ressource (resource)
- ▶ Ein- und Ausgabegeräte (I/O device, input/output)
- ▶ Benutzer (user), Single-User vs. Multi-User
- ▶ Multitasking, Prozess (process, task) vs. Programm
- ▶ Benutzerschnittstelle (user interface: UI, GUI)

Verarbeitungsarten

- ▶ Interaktive Verarbeitung (online processing, Dialogbetrieb): das ist der Normalfall...
- ▶ Stapelbetrieb (batch processing): Hintergrund vs. Vordergrund. Der Auftrag muss vollständig definiert sein. Während der Abarbeitung des Stapels kann nicht mehr eingegriffen werden.
- ▶ Echtzeitbetrieb: hard real-time vs. soft real-time
- ▶ Transaktionsbetrieb: Auftrag wird entweder ganz oder gar nicht abgearbeitet.

Begriff Betriebssystem

- ▶ Programm zwischen Benutzer und Hardware
- ▶ erlaubt Benutzern Programme auszuführen
- ▶ Verwaltung der Hardware-Ressourcen, z.B.:
 - ▶ CPU
 - ▶ (Haupt)Speicher
 - ▶ Tastatur und Bildschirm
 - ▶ Festplattenspeicher
 - ▶ Kommunikationskanäle

Geschichte der BS

► 1950

- anfangs keine Software
- anfangs nur Konsole, dann Lochkarten, Drucker, Magnetbänder
- dann Assembler, Lader, Binder
- OPEN SHOP: Betrieb durch Operator, Programmierer, Benutzer

► 1960

- batch Betrieb, dann multitasking, dann multiuser
- Platten: Geschwindigkeitsdiskrepanz CPU vs. I/O Geräten
→ CPU Auslastung → Verwendung der Platten als Puffer:
SPOOL (Simultaneous Peripheral Operation On-Line)
- COBOL, FORTRAN
- CLOSED SHOP: Betrieb nur durch Operator
- Beispiele: IBM OS/360, MULTICS

Geschichte der BS – 2

► 1970

- Aufteilung: Großrechner, Mini- und Mikrocomputer, dann Desktop
- Terminals, multiprocessor, fehlertolerante HW
- erste PCs (mit 256 Bytes RAM!)
- Beispiele: IBM MVS, Siemens BS2000, UNIX, DOS
- Netzwerke: ARPANet (Vorläufer des Internet)
- Großrechner: virtuelle Maschinen
 - IBM VM

► 1980

- Cluster von Minicomputer, Workstations
 - DEC-VMS, AIX, Solaris, HP-UX, Novell
- Echtzeitbetriebssysteme
 - QNX, VxWorks
- Netzwerke: TCP/IP

Geschichte der BS – 3

- ▶ 1990
 - ▶ multitasking, multiuser, dann multiprocessor im Desktopbereich
 - ▶ IBM OS/2, Windows NT, Linux, Mac OS
 - ▶ Weitere Verkleinerung: Handheld-Computer
 - ▶ EPOC (Psion PDA), Palm OS (Palm Computing)
 - ▶ Virtualisierung der HW im Serverbereich: VMWare
- ▶ 2000
 - ▶ Fortschritte: GUI, Speichermanagement, Netzwerk,...
 - ▶ Windows 2000,..., Windows 7, Mac OS X
 - ▶ Verbreitung der Virtualisierung
 - ▶ KVM, Hyper-V, VirtualBox
 - ▶ Desktop-Virtualisierung
 - ▶ Citrix, VMWare Horizon View, Remote Desktop Services

Geschichte der BS – 4

- ▶ 2008 ... Container LXC
- ▶ 2007 ... iPhone
- ▶ 2008 ... Android
- ▶ 2010 ... Windows Phone
- ▶ 2013 ... Windows 8.1
- ▶ 2013 ... Docker
- ▶ 2015 ... Windows 10

Hardware

- ▶ CPU (central processing unit, Prozessor)
- ▶ Speicherkontrolleinheit (MMU, memory management unit)
- ▶ Hauptspeicher (RAM), nicht flüchtiger Speicher (ROM, PROM, EPROM, EEPROM, Flash-EEPROM): wird durch Speicherkontrolleinheit angesteuert.
- ▶ Festplattenkontrolleinheit (harddisk controller): IDE, ATA (EIDE), SCSI, SATA
- ▶ Festplatten-, Disketten- und optische Laufwerke
- ▶ Videoadapter, Monitor, Tastatur, Maus, Drucker,...
- ▶ Netzwerkadapter
- ▶ Systembus: verbindet CPU, Speicherkontrolleinheit, Festplattenkontrolleinheit,...

Betriebssystemkomponenten

- ▶ Prozessverwaltung
- ▶ Speicherverwaltung
- ▶ Geräteverwaltung

Prozessverwaltung

- ▶ Prozesse (process, task) anlegen, beenden (Programm vs. Prozess!)
- ▶ CPU Zeit verteilen (scheduling, quasi-parallel vs. echt-parallel). Anforderungen unterschiedlich:
 - ▶ Dialogbetrieb vs. Stapelbetrieb (Interaktion vs. rechenintensiv),
 - ▶ Einzelbenutzer vs. Mehrbenutzersystem, Echtzeitbetrieb (realtime)
- ▶ Prozesse synchronisieren

Prozessverwaltung – 2

- ▶ Deadlocks (Verklemmungen) vermeiden, erkennen, auflösen (Beispiel Straßenkreuzung)
- ▶ Interprozesskommunikation
- ▶ Zugriffsschutz (Hardware: MMU)
- ▶ CPU Modus
 - ▶ user mode (Benutzerprozesse)
 - ▶ privileged mode oder kernel mode (Betriebssystem, direkter Zugriff auf Hardware und Speicher)

Speicherverwaltung

- ▶ Speicherarten
 - ▶ flüchtig (volatile):
 - ▶ Register, Cache (L1, L2, L3) (Größenordnung ns)
 - ▶ Hauptspeicher (Größenordnung μ s)
 - ▶ nicht flüchtig (persistent, non-volatile)
 - ▶ ROM, PROM,...
 - ▶ Festplatte (Größenordnung 10ms)
 - ▶ optische Platte (Größenordnung 100ms)
 - ▶ Magnetband (Größenordnung 100s)
- ▶ Hauptspeicher- und Sekundärspeicherverwaltung

Speicherverwaltung – 2

- ▶ Hauptspeicherverwaltung
 - ▶ Umsetzung der virtuellen Adresse auf eine physikalische Adresse durch die MMU.
 - ▶ belegte und freie Speicherbereiche verwalten
 - ▶ Zuteilung und Entzug von Speicherbereichen an Prozesse
- ▶ Sekundärspeicherverwaltung
 - ▶ ähnlich Hauptspeicherverwaltung: Zugriffszeiten, variabel
 - ▶ Disk Scheduling (mehrere Anfragen sammeln, Reihenfolge)
 - ▶ Pufferung
 - ▶ Speicherplatzzuteilung (Auswirkung auf Zugriffszeit)

Dateiverwaltung

- ▶ Erstellen, Manipulieren, Löschen von Dateien und Verzeichnissen
- ▶ Abbildung von logischer auf physikalische Struktur
- ▶ u.U. automatisches Sichern und Transaktionen (Operation wird entweder komplett oder überhaupt nicht durchgeführt)

Geräterverwaltung

- ▶ Erkennen, initialisieren, adressieren, abfragen,...
- ▶ Gerätetreiber (device driver)
- ▶ Controller
- ▶ meist "memory mapped" (Register, I/O ports)
- ▶ verschiedene Modi
 - ▶ programmed (I/O)
 - ▶ interrupt driven (I/O)
 - ▶ DMA (direct memory access)
 - ▶ Übertragung von großen Datenmengen
 - ▶ Daten vom Puffer der E/A-Kontrolleinheit (z.B. Festplattenkontrolleinheit) zum Hauptspeicher
 - ▶ *nicht* von der CPU sondern von der E/A-Kontrolleinheit
 - ▶ CPU initialisiert lediglich die Register der E/A-Kontrolleinheit
 - ▶ dann überträgt die E/A-Kontrolleinheit selbständig

Interrupt driven I/O

- ▶ Interrupt (SW oder HW):
 - ▶ CPU stoppt gerade aktuelle Arbeit und speichert Zustand
 - ▶ startet Interrupt-Routine und danach wieder zurück.
- ▶ Ein SW Interrupt wird auch trap genannt.

Interrupt driven I/O – 2

Prinzipieller Ablauf:

1. Benutzerprozess ruft Systemfunktion auf
2. Betriebssystem lädt/programmiert Register vom Controller
3. Controller startet I/O Operation und legt Daten im internen Puffer bzw. Register ab.
4. CPU wird mittels Interrupt verständigt.
5. CPU liest Puffer bzw. Register vom Controller aus.
6. Systemfunktion wird verlassen, Ergebnis dem Benutzerprozess zurückgegeben.

→ synchrone Operation

Synchron vs. asynchron

- ▶ lang andauernde Vorgänge → asynchrone Operation besser
- ▶ Synchroner Operation
 - ▶ Aufrufer wartet bis Operation beendet ist.
- ▶ Asynchrone Operation
 - ▶ Direkt nach Aufruf der Operation → Kontrolle an den Aufrufer
 - ▶ Zugriff auf das Ergebnis entweder durch pollen (periodisches abfragen) oder Callback

Arten von Kernel

- ▶ Monolithischer Kernel
 - ▶ Speicherverwaltung, Prozessverwaltung, Geräteverwaltung inkl. Gerätetreiber (kernel mode)!
 - ▶ Vorteil: Geschwindigkeit.
- ▶ Microkernel
 - ▶ nur Speicherverwaltung und Prozessverwaltung sowie Funktionen zur IPC und Synchronisation. Alles andere jeweils als eigener Prozes. Beispiele: Symbian OS, QNX Neutrino
 - ▶ Vorteil: Sicherheit, klare Trennung der Funktionen, Treiber im user mode!
- ▶ Hybridkernel
 - ▶ Kombination von monolithischen Kernel und Microkernel.
 - ▶ Teile der Gerätetreiber im Kernel. Beispiele: Windows NT, Mac OSX

Systemaufruf (monolithisch)

1. Prozess ruft Systemaufruf (system call) auf
2. Jede Systemfunktion hat eindeutigen Identifier (Zahl). Diese und die Parameter → Hauptspeicher
3. Es wird ein Softwareinterrupt (trap) ausgelöst.
4. Prozessor unterbricht die Anwendung und startet Interruptroutine (trap handler, ISR). CPU vom user mode in den kernel mode!
5. Über den Identifier kann die entsprechende Funktion ausgeführt werden. Dazu werden zuerst die Parameter in den Kernelspeicher kopiert.
6. Nach Beendigung kopiert die Funktion das Ergebnis in den Speicherbereich der Anwendung.
7. ISR ist beendet und Prozessor wird vom kernel mode wieder in den user mode geschalten.

Struktur einer Festplatte

- ▶ Aufteilung in Spuren und Sektoren
- ▶ MBR
 - ▶ betriebssystemübergreifend für PCs definiert
 - ▶ die ersten 512 Bytes auf der Festplatte
 - ▶ 446 Byte (!) für Programmcode
 - ▶ 64 für die Partitionstabelle (mit bis zu 4 Einträgen)
 - ▶ magische Zahl 0xAA55 am Ende
- ▶ Partitionen
- ▶ Bootsektor
 - ▶ kann Programmcode zum Starten eines Betriebssystems enthalten
 - ▶ die ersten 512 Bytes einer Partition
 - ▶ magische Zahl 0xAA55 am Ende

Partitionen einer PC-Festplatte

- ▶ physikalische Aufteilung
- ▶ zusammenhängender Bereich angegeben durch Sektoren
- ▶ verschiedene Arten einer Partition
 - ▶ primäre Partition (primary partition)
 - ▶ erweiterte (extended)
 - ▶ logische (logical)
- ▶ max. 4 primäre sind möglich
- ▶ eine erweiterte kann viele logische enthalten, aber direkt keine Daten
- ▶ mehr als 4 Partitionen benötigt → eine primäre durch eine erweiterte ersetzen

Partitionen einer PC-Festplatte – 2

- ▶ Primärer bzw. logischen Partition wird noch ein Typ zugeordnet (numerischer Code, der den Typ des Dateisystems angibt, z.B. 07 für NTFS, 82 für Linux swap oder 83 für Linux native).
- ▶ unter Linux wird jeder Partition ein Name zugeordnet, z.B.: `/dev/sda0`
 - ▶ im GRUB, z.B.: `(hd0,0)`
- ▶ Linux: meist auch eine swap partition (für virtuellen Speicher). Größe des Hauptspeichers + Größe der swap-Partition ergibt Gesamtgröße des zur Verfügung stehenden virtuellen Speichers.

Systemstart eines PCs (BIOS)

1. (Hardware-)Interrupt → CPU beginnt ab Adresse 0x0000: BIOS (basic input output system)
2. Initialisierung der Hardware (CPU-Register, Controller, Speicher inkl. Speichertest,...)
3. Information aus dem CMOS (Halbleitertechnologie mit geringem Stromverbrauch, historischer Begriff im PC Kontext): Datum, Zeit, Peripheriegeräte.
4. Laden des MBR (master boot record) und ausführen.
5. Daraufhin wird meist ein Boot-Manager geladen (wie z.B. GRUB oder der Windows Bootmanager), der u.U. auch die Auswahl eines Betriebssystems zulässt und dieser das Betriebssystem in den Speicher lädt und "ausführt".

Dateisystem

- ▶ Dateisystem (file system): Abbildung der logischen auf die physikalische Struktur
- ▶ formatieren → Dateisystem auf Partition
- ▶ Beispiele
 - ▶ fat32, ntfs
 - ▶ reiserfs, ext2, ext3, ext4,...

Dateisystem – 2

- ▶ Aus Benutzersicht stellt ein Dateisystem zur Verfügung:
 - ▶ Datei (file)
 - ▶ absoluter Pfad
 - ▶ Eigentümer (Benutzer ID)
 - ▶ Gruppe (Gruppen ID)
 - ▶ Dateimodus (file mode): Berechtigungen
 - ▶ atime (file access time)
 - ▶ mtime (file modify time): Änderungszeitpunkt des Inhalts
 - ▶ ctime (file change time): Änderungszeitpunkt des Inhalts oder der Änderungen der Berechtigungen oder des Besitzers
 - ▶ Zugriffskontroll-Listen (access control lists, ACL)
 - ▶ Verzeichnis (directory) → hierarchische Dateisysteme
 - ▶ Weiters: Funktionen, um auf diese zuzugreifen bzw, diese zu verändern.

Kommandointerpreter

- ▶ Schnittstelle zum Benutzer (zum Programmierer mittels Systemaufrufen)
- ▶ Starten und beenden von Prozessen
- ▶ Navigation im Dateisystem
- ▶ Dateimanipulation