

Rechnerorganisation

Sommersemester 2023 – 10. Vorlesung

Prof. Stefan Roth, Ph.D.

Technische Universität Darmstadt

3. Juli 2023



Inhalt

- 1 Organisatorisches
- 2 Analyse der Rechenleistung
- 3 Gleitkommazahlen
- 4 Leistungsbewertung II
- 5 Betriebssysteme & Ausnahmebehandlung
- 6 Zusammenfassung und Ausblick
- 7 Literatur

Organisatorisches



Organisation und Inhalt

- Evaluation der Veranstaltung Rechnerorganisation
 - ▶ Evaluation der Vorlesung & Übung
 - ▶ Evaluation Vorlesung und Übung müssen getrennt abgegeben werden
 - ▶ TANs (je eine TAN für Vorlesung sowie Übung) gibt es in Moodle
 - ▶ Evaluation möglich bis 13. Juli 2023
 - ▶ Gerne auch direktes Feedback an mich!
- Die Fachprüfung, zu der eine Anmeldung in TUCaN notwendig ist, findet statt als Präsenzklausur am:
 - ▶ Datum: 21.09.2023
 - ▶ Uhrzeit: 9:00 - 11:30 Uhr (Klausurdauer 90 Minuten)
 - ▶ Räume: werden rechtzeitig bekannt gegeben
- Deadline für Programmiertestat 02 ist heute nacht.
 - ▶ Nicht vergessen auch einen Testatstermin mit einem Tutor zu vereinbaren (Deadline auch heute nacht)!
- Theorietestat 04 wird nach der Vorlesung freigeschaltet.

Mikroarchitektur [HH16, S. 385 - 484]

- Einführung in die Mikroarchitektur
- Eintakt-Prozessor
- Mehrtakt-Prozessor
- Pipeline-Prozessor
- Analyse der Rechenleistung
- Ausnahmebehandlung
- Weiterführende Themen

Analyse der Rechenleistung



Analyse der Rechenleistung \Rightarrow Leistungsbewertung

- Analyse der Rechenleistung dient der Leistungsbewertung von Rechnersystemen
- **Hardware (Prozessor, Speicher):** bestimmt die Leistungsfähigkeit eines Rechnersystems
- Aber auch das Betriebssystem spielt eine wichtige Rolle.
- Um die Leistungsfähigkeit eines Rechnersystems beurteilen zu können muss es quantifizierbare Leistungsmaße geben.
- Was sind geeignete Leistungsmaße bzw. Bewertungskriterien eines Rechnersystems und eines Prozessors?
- Einige Maße:
 - ▶ Taktfrequenz
 - ▶ Anzahl der Prozessoren
 - ▶ Größe und Art des Speichers

Bewertungskriterien I

- Perspektive des Käufers eines Prozessors:
 - ▶ Welcher hat die beste Leistung?
 - ▶ die niedrigsten Kosten?
 - ▶ das beste Preis-/Leistungsverhältnis?
- Perspektive des ASIC¹-Designers eines Prozessors
- Es existieren verschiedene Designoptionen:
 - ▶ Welche bringt die beste Leistungsverbesserung?
 - ▶ die geringsten Kosten?
 - ▶ das beste Preis-/Leistungsverhältnis?
- Sowohl Käufer als auch ASIC-Designer brauchen ein Maß (Metrik) zur Leistungsbewertung.

¹Application Specific Integrated Circuit

Bewertungskriterien II

- Welche Leistungsmaße sind denkbar?
 - ▶ Antwortzeiten, Durchsatz, Ausführungszeiten
- Aus Sicht eines Benutzers: Antwortzeit
 - ▶ Wie schnell wird eine einzelne Aufgabe vom Prozessor gelöst?
 - ▶ \Rightarrow Der schnellste Prozessor ist derjenige, der eine bestimmte Aufgabe am schnellsten löst.
- Welche Aufgaben sind repräsentativ? Unterschiedliche Benutzer haben unterschiedliche Aufgaben.
 - ▶ Beim wissenschaftlichen Rechnen sind andere Aufgaben zu lösen, als beim Betrieb eines Datenbankservers.
- Aus Sicht eines Rechenzentrumsbetreibers: Durchsatz
 - ▶ Wie viel Arbeit (Jobs) kann ein Rechner pro Tag/Stunde/Woche/sec/ns erledigen?
 - ▶ \Rightarrow Der schnellste Prozessor ist derjenige, der die meisten Jobs pro Tag bearbeitet.

Bewertungskriterien III

- Kriterium: Ausführungszeit
 - ▶ Die Leistung eines Prozessor wird am besten durch die Ausführungszeit für ein Programm beschrieben.
- Unterschied: Antwortzeit und Ausführungszeit
 - ▶ Antwortzeit: Zeit, die real vergeht, bis eine gegebene Aufgabe vom Rechner gelöst wurde.
 - ▶ zählt alles (Festspeicher, Speicherzugriffe, Eingabe/Ausgabe); bewertet Systemperformanz
 - ▶ aber i. d. R. ungeeignet für Prozessorvergleich
 - ▶ Ausführungszeit: Zeit, die die CPU bei der Lösung der Aufgabe benötigt ohne Ein-/Ausgabe und Zeit für andere Aufgaben
- Ausführungszeit besteht aus:
 - ▶ system CPU time (CPU-Zeit für Betriebssystemaufgaben) und
 - ▶ user CPU time (CPU-Zeit, die zur Ausführung eines Programms benötigt wird)

Bewertungskriterien IV

- Ausführungszeit Unix **time** Befehl – Beispiele

- ▶ 1. Beispiel

- ★ 1 Thread:

- real 33m50.500s

- user 33m48.323s

- sys 0m0.412s

- ▶ 2. Beispiel

- ★ 2 Threads:

- real 14m36.878s

- user 28m52.740s

- sys 0m0.276s

- ▶ 3. Beispiel

- ★ 4 Threads:

- real 7m42.551s

- user 27m59.141s

- sys 0m0.336s

Bewertungskriterien V – Taktsignal

- Anwender denken an Ausführungszeit in gemessener Zeit.
- Prozessorentwickler verwendet ein Maß, das in Bezug dazu steht, wie schnell die Hardware (=Mikroarchitektur) elementare Funktionen durchführen kann.
- Nahezu alle heutigen Computer² beruhen auf einem Taktsignal, die das Auftreten von (synchronisierenden) Schaltereignissen in der Hardware bestimmt ((clock) cycles, (clock) ticks, clock periods)

²Es gibt immer wieder Projekte, in denen asynchrone Schaltwerke realisiert werden.

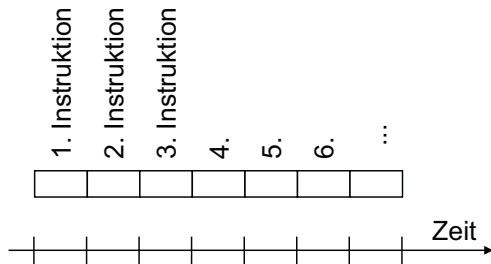
Taktfrequenz – Bedeutung und Grenzen

- Leistungssteigerung wurde lange Zeit durch Erhöhen der Taktfrequenz erreicht. Aktuell liegt der Prozessortakt vieler Mikroprozessoren bei 3.x GHz.
 - ▶ Intel® Pentium 4 EE, 3,8 GHz Takt, 11,5 SPECint2006³
 - ▶ Intel® Montecito 9050, 1,6 GHz Takt, 14,5 SPECint2006
- Bedingt durch die Technologie (CMOS-Technologie) steigt der Leistungsumsatz der Prozessoren mit dem Takt ($P \approx U^2 \cdot f \cdot C_L$).
- Die entstehende Wärme ist nur mit großem Aufwand abzutransportieren.
- Parallelrechner:
 - ▶ Integration mehrerer CPUs auf einem Chip – z. B. Intel® i9-9900K, 3,6 GHz – 5 GHz (im Turbo-Takt), 8 Kerne
 - ▶ Massiv parallele Systeme mit mehreren tausend Prozessoren, Spezialarchitekturen, z. B. Vektorrechner

³Ergebnis eines Leistungstests, vgl. <https://www.spec.org/>

Bewertungskriterien VI – Taktzyklen

- Wie viele Taktzyklen⁴ benötigt ein Programm?
- Man könnte annehmen, dass $\# \text{ Instruktionen} = \# \text{ Takte}$, d. h.



- Annahme ist nur bei Eintakt-Prozessor richtig
- Verschiedene Instruktionen benötigen unterschiedliche Zeit
- Warum ist das so?

⁴Taktzyklus die Zeitspanne zwischen zwei Impulsen eines Oszillators

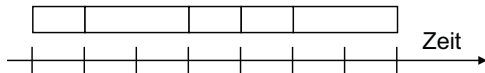
Bewertungskriterien VII – Eintakt-Prozessor \iff Mehrtakt-Prozessor

- Eintakt-Prozessor

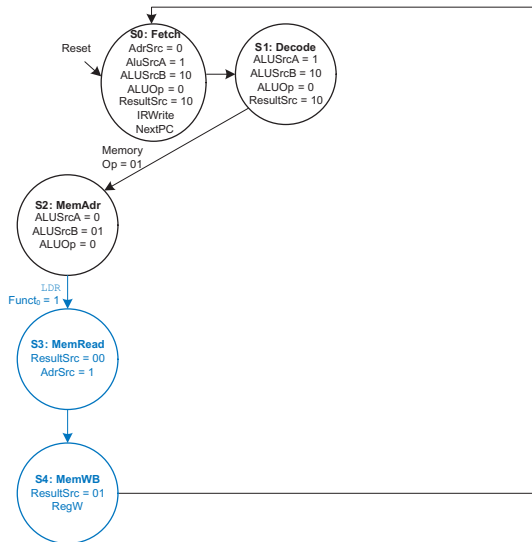
- ▶ Ein Takt pro Instruktion (d. h. alle Instruktionen haben gleiche Taktzykluslänge) ist ineffizient, da die Ausführungszeit dann durch die am längsten dauernde Instruktion bestimmt wird
- ▶ Hardware-Einheiten (z. B. Speicher) müssen mehrfach vorhanden sein

- Mehrtakt-Prozessor

- ▶ pro Instruktion mehrere, aber kürzere Taktzyklen (=höhere Taktfrequenz)
- ▶ Anzahl der Taktzyklen pro Instruktion ergibt die *clock cycles per instruction* (CPI)



Zur Erinnerung: Steuerwerk Mehrtakt-Prozessor



Rechenleistung eines Prozessors

- Ausführungszeit eines Programms
 - ▶ Ausführungszeit = ($\#$ Instruktionen) \cdot (Takte/Instruktion) \cdot (Sekunden/Takt)
- Definitionen:
 - ▶ Takte/Instruktion = CPI (cycles per instruction)
 - ▶ Sekunden/Takt = Taktperiode
 - ▶ $1/\text{CPI} = \text{Instruktionen/Takt} = \text{IPC}$ (instructions per cycle)
- Als einfaches Leistungsmaß wird MIPS⁵ verwendet
- Einhalten zusätzlicher Anforderungen:
 - ▶ Kosten
 - ▶ Energiebedarf
 - ▶ Rechenleistung

⁵Million Instructions Per Second

Die Klassifikation von Flynn I

- Klassifikation von Flynn (vgl. Some Computer Organizations and Their Effectiveness)
- Aufteilung in sogenannte *Instruction Streams* und *Data Streams*
- Instruction Stream:
 - ▶ SI – Single Instruction
 - ▶ MI – Multiple Instruction (mehrere Befehle zu einem Zeitpunkt)
- Data Stream:
 - ▶ SD – Single Data
 - ▶ MD – Multiple Data
- Ergibt dann vier Kombinationen:
 - ▶ SISD – Von-Neumann-Rechner
 - ▶ SIMD – Feldrechner, Vektorrechner
 - ▶ MIMD – Multiprozessorsysteme
 - ▶ MISD – ?

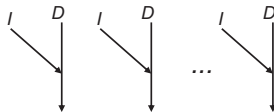
Die Klassifikation von Flynn II

- Veranschaulichung von Flynn

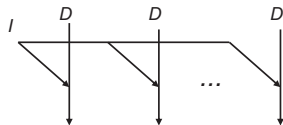
SISD



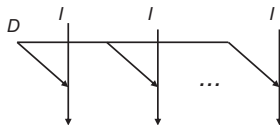
MIMD



SIMD



MISD



Die Klassifikation von Flynn – Diskussion

- sehr grobe Klassifikation
- viele Aspekte werden nicht erfaßt, wie
 - ▶ Pipelining
 - ▶ Wortbreite
 - ▶ Verbindungsstrukturen
 - ▶ Speicherorganisation
 - ★ lokale/globale Speicher, virtueller Speicher, Caches
 - ★ Trennung von Programm- und Datenspeicher
- Aber die Begriffe haben sich etabliert
- \Rightarrow SIMD Befehle, SSE⁶ Einheiten

⁶Streaming SIMD Extensions

Gleitkommazahlen



Zahlendarstellung in Rechnersystemen

- Man unterscheidet zwischen ganzen Zahlen (Integer) und reellen Zahlen (Real).

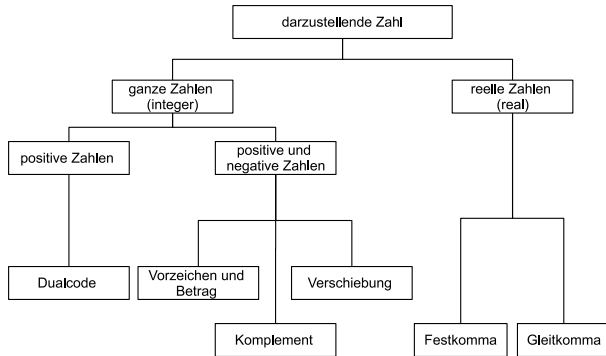


Abbildung: Zahlendarstellung in Rechnersystemen

Darstellung reeller Zahlen

- Bisher sind nur ganze Zahlen betrachtet worden.
- Es gibt zwei verschiedene Darstellungen für reelle Zahlen:
 - ▶ Festkommadarstellung
 - ▶ Gleitkommadarstellung
- Zur Festkommadarstellung (vgl. [HH16, S. 255 ff.]):
 - ▶ Bei der Festkommadarstellung lässt man bei internen Zahlenabbildung das Komma weg und merkt sich, wo es stehen müsste.
 - ▶ Zum Programmbeginn wird die Kommastelle der entsprechenden Variablen definiert und bleibt dann während des Programms fest.

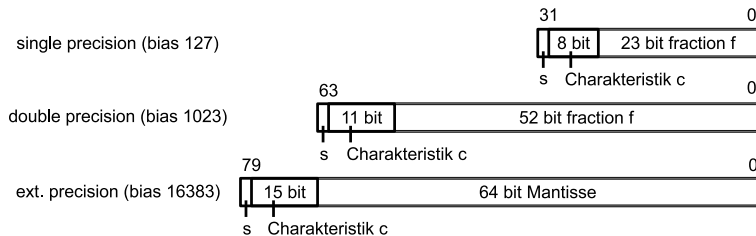
Gleitkommadarstellung nach ANSI/IEEE 754

- Bei der Gleitkommadarstellung (auch halblogarithmische Darstellung genannt) ist die Kommastelle Bestandteil der Zahl und kann dadurch auch im Laufe der Berechnungen verändert werden.
- Durch eine standardisierte Schreibweise bzw. Darstellung braucht man dabei das Komma nicht explizit anzugeben.
- Die Floatingpoint-Darstellung ist standardisiert nach ANSI/IEEE 754 (US-American National Standard Institute/Institute of Electrical und Electronics Engineers).
- Dieser Standard hat sich weltweit durchgesetzt und bildet die Grundlage sowohl für einen weltweiten Datenaustausch wie auch für die Rechenwerke (ALUs⁷) in den Prozessoren.
- Der ursprüngliche Standard wurde 1985 verabschiedet. Im Jahre 2008 hat es eine Revision gegeben.

⁷Arithmetic Logic Units

Gleitkommadarstellung nach ANSI/IEEE 754

- Im IEEE 754 Standard sind single, double und extended precision (Genauigkeit) Formate definiert.
- Folgende Abbildung zeigt die Formate:



Gleitkommadarstellung nach ANSI/IEEE 754

- Besonderheiten: Die Null kann (durch die Normalisierung) nicht dargestellt werden.
- Unendliche Zahlen (oder keine Zahlen) sollen auch *dargestellt* werden.

Normalisiert	\pm	$0 < \text{Charakteristik} < \text{max}$	Beliebiges Bitmuster
Nicht normalisiert	\pm	0	Beliebiges Bitmuster $\neq 0$
Null	\pm	0	0
Unendlich	\pm	111...1 (max)	0
Keine Zahl ⁸	\pm	111...1 (max)	Beliebiges Bitmuster $\neq 0$

⁸NaN: not a number, z. B. Division durch Null, Ergebnis der Quadratwurzel einer negativen Zahl

Leistungsbewertung II



Leistungsbewertung II

- Nach dem kurzen Einschub zu den Gleitkommazahlen
- Zusammenfassung: Leistungsmaße von Prozessoren sind
 1. Taktfrequenz
 2. CPI Rate (Clock Cycles per Instruction)
 3. MIPS Rate (Million Instructions per Second)
 4. MFLOPS (Million Floating-point Operations per Second)
- Zur Erinnerung: Die Kriterien stellen ein Leistungsmaß für den Prozessor dar, lassen aber nur bedingt Aussagen über die Leistungsfähigkeit des Rechnersystems zu. Die Leistung des Systems hängt sowohl von der umgebenden Hardware, als auch von dem komplexen Zusammenspiel von Hardware und Software, insbesondere dem Betriebssystem, ab.

Leistungsbewertung II – Benchmarks

- Benchmarks sind repräsentative Programme, die auf den zu vergleichenden Rechnern ausgeführt werden
- Es gilt der Rechner als der Leistungsfähigste, bei dem die Benchmark-Programme am schnellsten ausgeführt werden.
- Aber: In der Praxis wurden häufig Compiler oder auch Instruktionssätze so optimiert, dass gängige Benchmarks besonders schnell abliefen
- Vier Typen von Benchmarks
 - ▶ **Reale Programme:** Verwendung häufiger Anwendungen wie C-Compiler, Textprozessoren (\LaTeX), SPICE usw.
 - ▶ **Kernels:** Kurze kritische Auszüge aus realen Programmen, die isoliert zur Ausführung gebracht werden
 - ▶ **Toy Benchmarks:** Kleine, einfach zu schreibende Programme, die leicht zur Ausführung gebracht werden können (z. B. Quicksort)
 - ▶ **Synthetische Benchmarks:** Speziell entwickelte Programme, die die Leistungsfähigkeit einzelner Instruktionen oder einzelner Komponenten der zu vergleichenden CPUs aufdecken sollen (z. B. Dhrystone)

Leistungsbewertung II – Benchmarks

- Standard-Benchmark Suites: **SPEC**
- Standard Performance Evaluation Corporation www.spec.org
- gegründet 1989 durch mehrere Rechnerhersteller (Apollo/HP, DEC, MIPS, Sun Microsystems)
- Beschaffen eines Satzes von zunächst realen Programmen, die auf allen Rechnern lauffähig sind und in ihrer Gesamtheit zur Messung der Prozessorleistung herangezogen werden.
- Die SPEC-Benchmark-Suite durchlief mehrere Generationen, in denen die einzelnen Programme jeweils den aktuellen Eigenschaften der Rechner/Prozessoren und der Systemsoftware angepasst wurden.
- Anpassungen z. B. x87 \Rightarrow MMX, SSE

Leistungsbewertung II – Benchmarks

- SPEC Benchmarks
- CPU89
 - ▶ 10 Programme liefern eine einzelne Zahl (SPECmarks)
- CPU92
 - ▶ SPECInt92 (6 integer Programme) and SPECfp92 (14 floating point Programme)
- CPU95
 - ▶ neue Programme: SPECint95 (8 Integer Programme) und SPECfp95 (10 Floating Point Programme)
- CPU2000
 - ▶ CINT2000 11 integer (gzip, gcc, crafty, perl, ...)
 - ▶ CFP2000 14 floating-point (swim, mesa, art, ...)
 - ▶ Alle Ergebnisse gemessen relativ zu Basisrechner Sun 300 MHz 256Mb-RAM Ultra5_10; diese hat Score 100
- CPU2006
 - ▶ CINT2006 Benchmark für Integer-Performanz
 - ▶ CFP2006 Benchmark für Gleitpunkt-Performanz

Leistungsbewertung II – Benchmarks

- SPEC Benchmarks sind nicht nur für Prozessoren erhältlich
 - ▶ Grafikkarten
 - ▶ MPI⁹/OpenMP
 - ▶ Mailserver
 - ▶ Java (für virtuelle Maschinen)
 - ▶ ...
- Die Benchmarks sind kostenpflichtig
- Allerdings einige freie Tests verfügbar (z. B. **gzip** und **bzip**).
 - ▶ z. B. von Intel (Link)
 - ▶ LINPACK: ursprünglich nur numerische Programmbibliothek zum Lösen von linearen Gleichungssystemen
 - ▶ LINPACK wird zur Leistungsmessung von Supercomputern eingesetzt (Lichtenberg II)
 - ▶ aber auch verfügbar für: Linux, Windows, Mac OS

⁹Message Passing Interface

Leistungsbewertung II – Benchmarks

- Ausführung auf Notebook (vier Kerne, 2 GHz)
- Ergebnis
 - ▶ CPU frequency: 1.995 GHz
Number of CPUs: 4
Number of threads: 4
 - ▶ Performance Summary (GFlops)

Size	Average	Maximal
15000	21.7752	21.7752
14000	25.4207	25.6804
13000	25.3818	25.4019
12000	25.0435	25.4617
11000	25.4458	25.6519
10000	25.4523	25.4710
8000	24.2956	24.3837
6000	23.0570	24.0496
1000	16.5292	17.3094

Leistungsbewertung II – Benchmarks

- Außerdem verfügbar unter Linux: **cat /proc/cpuinfo**
- Angaben von
 - ▶ Taktfrequenz
 - ▶ Cache-Größe
 - ▶ Floating Point Unit (vorhanden)
 - ▶ BogoMips
- BogoMips¹⁰
 - ▶ „unwissenschaftliches“ Maß für die CPU-Geschwindigkeit
 - ▶ Wert wird beim Booten ermittelt, um interne Warteschleifen zu kalibrieren
- Mit BogoMips können keine Leistungsvergleiche zwischen Prozessoren durchgeführt werden.

¹⁰Bogus – scheinbar, MIPS – Millionen Instruktionen pro Sekunde

Betriebssysteme & Ausnahmebehandlung



Betriebssysteme I – Historische Entwicklung

- Anfänge der Informationsverarbeitung um 1940:
 - ▶ Rechner werden von Ingenieur:innen und Wissenschaftler:innen programmiert, die sie auch konstruiert haben
 - ▶ Probleme lagen mehr an der Hardware denn an der Programmierung
 - ▶ hohe Hardwarekosten, geringe Softwarekosten
 - ▶ Benutzer schreiben eigenständig Programme für viele Bereiche
 - ▶ Kosten für Programmerstellung stiegen stark an \Rightarrow teure Hardware soll gut ausgenutzt werden
- Diese Entwicklung bewirkte:
 - ▶ Entwicklung von Standardprogrammen für die **Steuerung von Rechnerkomponenten**
 - ▶ Entwicklung von Standardprogrammen für die **Auftragsabarbeitung**
- Ausgangspunkt für die Erstellung von Systemsoftware:
 - ▶ monitor system, supervisory system, executive system,
 - ▶ operating system \Rightarrow Betriebssystem (BS)

- **Definition eines Betriebssystems nach DIN 44300:**

Die Programme eines digitalen Rechensystems, die zusammen mit den Eigenschaften der Rechenanlage die Grundlage der möglichen Betriebsarten des digitalen Rechensystems bilden und insbesondere die Abwicklung von Programmen steuern und überwachen.

- **Definition eines Betriebssystems nach ANSI:**

Software, which controls the execution of computer programs and which may provide scheduling, debugging, input/output control, accounting, compilation, storage assignment, data management and related services.

Aufgaben eines Betriebssystems I

- „Veredelung“ der Hardware:
 - ▶ Geräteüberwachung und -steuerung
 - ▶ einfache, einheitliche I/O-Schnittstelle inkl. Fehlerbehandlung
 - ▶ Unterbrechungsbehandlung
 - ▶ Benutzerschnittstelle
- Organisation und Steuerung des Ablaufs:
 - ▶ Programmabläufe
 - ▶ I/O parallel zu Programmabläufen
 - ▶ Betriebsmittelverwaltung: Speicher, Prozessoren, Geräte
 - ▶ Protokollierung, Accounting
- Langfristige Datenhaltung:
 - ▶ Hintergrundspeicherverwaltung
 - ▶ Dateiverwaltung, Dateien
 - ▶ Schutz von Dateien
 - ▶ Dateisicherung, Archivierung

Aufgaben eines Betriebssystems II

- Einhaltung von Qualitätsanforderungen, z. B.
 - ▶ Leistung (kurze Antwortzeiten, hoher Durchsatz)
 - ▶ Verfügbarkeit
 - ▶ Sicherheit
- zunehmende Anforderungen an das Rechensystem, z. B.
 - ▶ durch Benutzer: will vieles gleichzeitig, bequem und möglichst schnell
 - ▶ durch Eigentümer: will hohe Auslastung und Verfügbarkeit
 - ▶ u.a. dadurch \Rightarrow zunehmende Komplexität des Betriebssystems
- Somit ermöglicht ein BS den Benutzern einer Rechenanlage eine
 - ▶ ggf. gemeinsame Nutzung
 - ▶ bei einfacher Handhabung
 - ▶ und es verwaltet und steuert die verschiedenen Komponenten eines Rechensystems mit dem Ziel optimaler Nutzung

Betriebssysteme – Eine andere Sicht

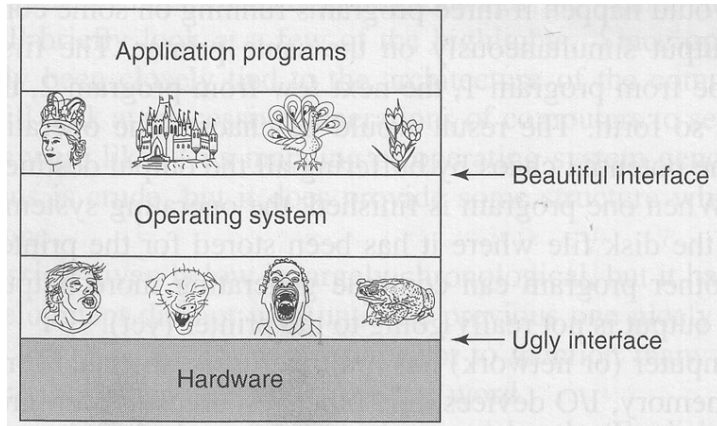
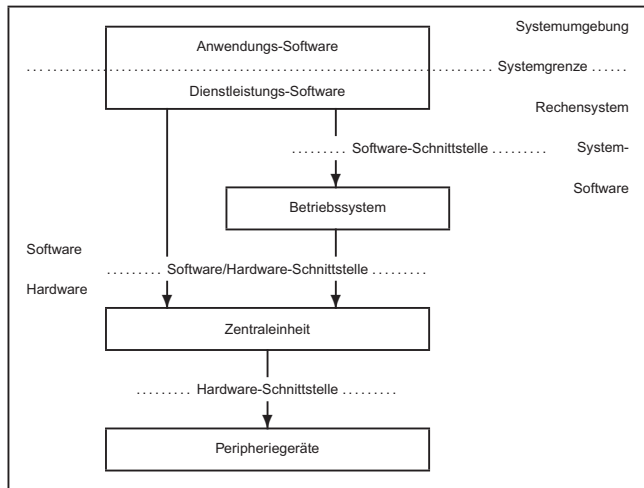


Abbildung: Quelle: [Tan09, S. 35]

Ein verfeinertes Schichtenmodell



Arbeitsweise eines Prozessors I

Betriebszustände eines Prozessors:

- Prozessor kann sich in verschiedenen **Betriebszuständen** befinden
- Betriebszustand teilweise definiert durch:
 - ▶ Maschinenzustand
 - ▶ Privilegierungszustand
- **Maschinenzustand**:
 - ▶ Prozessor kann aus- oder angeschaltet sein
 - ▶ Prozessor kann gerade im **Urladevorgang** sein
 - ▶ Zustände: stop, laden, tätig

Arbeitsweise eines Prozessors II

Betriebszustände eines Prozessors:

- Privilegierungszustand:
 - ▶ Zustände definieren die Menge der jeweils erlaubten, ausführbaren Maschinenbefehle
 - ▶ oft existieren
 - ★ **Anwenderzustand**: nicht privilegiert; nur eingeschränkter Befehlsvorrat gültig
 - ★ **Systemzustand**: privilegiert; voller Befehlsvorrat gültig
 - ▶ Befehl heißt **privilegiert**, wenn ausschließlich im Systemzustand ausführbar
- Hardware muss gestatten, Zustandswechsel per Software herbeizuführen

Konzept der Unterbrechungen I

- Entwicklung des Unterbrechungskonzeptes auf Wunsch der NASA 1956 mit UNIVAC 1103 von Remington Rand
- IBM folgte 1958 mit IBM 709
- ohne Unterbrechungen ist eine mögliche Arbeitsweise:
 - ① Prozessor stösst Tätigkeit eines Gerätes an
 - ② Gerät arbeitet relativ selbständig während Prozessor auf Beendigung wartet
 - ③ Prozessor arbeitet weiter
- zeitlich parallele Tätigkeit von Komponenten nicht oder nicht effizient realisierbar
- aufgrund relativ geringer Verarbeitungsgeschwindigkeit der Peripheriegeräte ist Prozessor schlecht ausgelastet

Konzept der Unterbrechungen II

- mit Unterbrechungen ist die Arbeitsweise:
 - ① Prozessor stösst Tätigkeit eines Gerätes an
 - ② Gerät arbeitet relativ selbständig und parallel zum Prozessor, der andere Arbeiten wahrnimmt
 - ③ am Ende der Tätigkeit meldet sich das Gerät beim Prozessor mit einer Unterbrechung
- besseres Leistungsverhalten
- Prozessor und Peripheriegeräte können zeitlich überlappend arbeiten
- Unterbrechungskonzept verbessert Synchronisation zwischen Prozessor und Gerät

Konzept der Unterbrechungen III

- Unterbrechung \Rightarrow Interrupt
- Signal, das
 - ▶ Befehlszyklus des Prozessors abändert bzw. unterbricht und
 - ▶ Befehlszyklus an spezifizierter Stelle fortführt
- Unterbrechung wird durch Eintreten eines speziellen **Ereignisses** ausgelöst
- **Ereignisursache** in Software oder Hardware
- grobe Unterbrechungsklassifikation nach Ereignisursache
 - ▶ Programmbezogene Unterbrechungen
 - ▶ Systembezogene Unterbrechungen
 - ▶ Maschinenfehler

Konzept der Unterbrechungen IV

Programmbezogene Unterbrechungen:

- der in Ausführung befindliche Befehl löst Unterbrechung aus
 - ▶ Unterbrechung trifft Verursacher
 - ▶ **synchrone** oder **interne** Unterbrechungen
 - ▶ exceptions
- Unterbrechungsursachen sind u. a.
 - ▶ arithmetische Fehler (bspw. Division durch Null, Überlauf)
 - ▶ Adressenfehler (bspw. Zugriff auf nicht installierten Speicher, fehlende Rechte)
 - ▶ falsche Befehle (bspw. privilegierter Befehl im Anwenderzustand)
 - ▶ Spezialbefehle zum Einleiten von Systemaufrufen

Konzept der Unterbrechungen V

Systembezogene Unterbrechungen:

- Unterbrechungsursache liegt außerhalb der Prozessoraktivität
- Unterbrechungsquelle ist ein relativ selbständig arbeitendes Gerät
- Unterbrechung trifft das zufällig laufende Programm \Rightarrow **asynchrone** oder **externe** Unterbrechung
- Beispiele:
 - ▶ Zeitgeberunterbrechungen
 - ▶ I/O-Unterbrechungen
 - ▶ Prozessoranrufe

Konzept der Unterbrechungen VI

Besondere Anwendung von Unterbrechungen:

- Spezialbefehle \Rightarrow Systemaufrufe (*system calls*)
- ermöglichen einen Wechsel von nieder- in höherprivilegierten Zustand
- dadurch wird u. a. ermöglicht, Benutzern Dienste des Betriebssystems verfügbar zu machen
- Befehle dieser Art
 - ▶ SWI \Rightarrow Software (SW) Interrupt (I) bei ARM[®]-Architekturen
 - ▶ INT bei Intel x86
 - ▶ TRAP bei Motorola 680x0
 - ▶ SUPERVISOR_CALL (SVC) bei IBM- und Siemens-Systemen

Zusammenfassung und Ausblick



Zusammenfassung und Ausblick

Zusammenfassung

- Analyse der Rechenleistung
- Ausnahmebehandlung

Ausblick

- Speicher
- Caches & Lokalität

- Die Herausforderungen bei der Analyse der Rechenleistung eines Rechnersystems habe ich verstanden ✓
- Die Aufgaben eines Betriebssystems sind mir (im groben) klar geworden und den Nutzen von Interrupts habe ich verstanden ✓
- ...

Literatur



- [BO10] Bryant, Randal E. und David R. O'Hallaron: *Computer Systems - A Programmer's Perspective*.
Prentice Hall, 2010.
- [HH16] Harris, David Money und Sarah L. Harris: *Digital Design and Computer Architecture, ARM® Edition*.
Morgan Kaufmann, 2016.
- [Tan09] Tanenbaum, Andrew S.: *Moderne Betriebssysteme*.
Pearson, 2009.