

Prof. Dr. Jörg Keller, Prof. Dr. Wolfram Schiffmann

Kurs 01608

Computersysteme I

LESEPROBE

Fakultät für
**Mathematik und
Informatik**

Das Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere das Recht der Vervielfältigung und Verbreitung sowie der Übersetzung und des Nachdrucks bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder ein anderes Verfahren) ohne schriftliche Genehmigung der FernUniversität reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Leseprobe zum Kurs 1608/09

Computersysteme I+II

Dieser Kurs wird im Sommersemester 2017 erstmalig in überarbeiteter Form angeboten.

Inhaltsverzeichnis

1 Technische Informatik	3
1.1 Gebiete der Informatik	4
1.2 Anwendungsbereiche der Technischen Informatik	5
1.3 Teilbereiche der Technischen Informatik	5
1.3.1 Hardware-Entwurf	6
1.3.2 Rechnerarchitektur	6
1.3.3 Bewertung von Computersystemen	6
1.3.4 Echtzeit- und eingebettete Systeme	6
1.3.5 Betriebssysteme	7
1.3.6 Kommunikationsnetze und verteilte Systeme	7
1.4 Einordnung des Moduls Computersysteme	7
1.5 Weitere Kurse zur Technischen Informatik	8
Index	9

Kapitel 1

Technische Informatik

Kapitelinhalt

1.1	Gebiete der Informatik	4
1.2	Anwendungsbereiche der Technischen Informatik .	5
1.3	Teilbereiche der Technischen Informatik	5
1.4	Einordnung des Moduls Computersysteme	7
1.5	Weitere Kurse zur Technischen Informatik	8

1.1 Gebiete der Informatik

Die Informatik beschäftigt sich mit der automatischen Verarbeitung von Informationen mit Computersystemen. Die Bezeichnung *Informatik* wurde von Karl Steinbuch eingeführt und ergibt sich aus der Kombination der Begriffe Information und Automatik.

Wir können im Wesentlichen drei Gebiete der Informatik unterscheiden:

- Theoretische Informatik,
- Praktische und Angewandte Informatik,
- Technische Informatik.

Die *Theoretische Informatik* widmet sich dabei den formalen und mathematischen Aspekten. Sie beantwortet grundlegende Fragen zu Logik und Deduktion, Formalen Sprachen, Automaten-, Berechenbarkeits- und Komplexitätstheorie sowie zur Algorithmentheorie.

Bei der *Praktischen Informatik* stehen keine abstrakten sondern konkrete Problemstellungen im Vordergrund. Es werden Methoden und Werkzeuge für die Erstellung von Softwarelösungen solcher Probleme entwickelt. Die wichtigsten Teilbereiche bilden die Softwaretechnik, Programmiersprachen, Datenstrukturen, Algorithmen, Datenbanken, Compilerbau und Betriebssysteme.

Zur *Angewandten Informatik* zählen die Teilbereiche Webbasierte Systeme, Mensch-Maschine-Schnittstellen, Multimediatechniken, Künstliche Intelligenz, Wissensbasierte Systeme, Bio-Informatik, Computergrafik, Bildverarbeitung und IT-Sicherheit.

Die *Technische Informatik* vereint Konzepte und Methoden zum Entwurf von Computersystemen, die für verschiedene Anwendungsbereiche optimiert werden. Im Folgenden werden wir – beginnend mit kleinen und wenig performanten Computersystemen – verschiedene Anwendungsbereiche vorstellen.

Wie bei vielen Klassifizierungen gilt auch hier, dass die Gebiete der Informatik nicht disjunkt sind. So gibt es beispielsweise starke Bezüge zwischen der Softwaretechnik und der theoretischen Informatik im Bereich automatischer Ableitung von Programmeigenschaften, die bei einer Code-Umstrukturierung erhalten bleiben sollen. Beim Entwurf eines Mikrochips werden die Schaltetze heute durch Hardware-Beschreibungssprachen spezifiziert, so dass sich bei Chip-Entwurfssystemen beispielsweise auch umfangreiche softwaretechnische Fragestellungen ergeben.

Auch zur Einordnung einzelner Fachgebiete der Informatik finden sich in der Literatur Unterschiede. So findet sich die Bildverarbeitung teils in der Angewandten, teils in der Technischen Informatik.

1.2 Anwendungsbereiche der Technischen Informatik

Eingebettete Systeme sind in großer Zahl in Geräten der Haushalts- und Unterhaltungselektronik sowie in modernen Fahrzeugen zu finden. Mobile Geräte wie Smartphones oder Tablet-Computer verfügen zusätzlich über berührungsgesteuerte Bildschirme (touch screens) und Telekommunikationsschnittstellen. Desktop-Computer (häufig auch Personal Computer oder PC genannt) besitzen meist großformatige und hochauflösende Bildschirme, eine Tastatur, leistungsstarke Mehrkern-Prozessoren sowie größere Haupt- und Sekundärspeicher. Bei Server-Computern verzichtet man auf Bildschirm bzw. Tastatur und beschränkt sich auf den Zugriff über eine Netzwerk-Schnittstelle. Server sind auf maximale Performanz optimiert und verfügen daher meist über mehrere Mehrkern-Prozessoren sowie graphische Co-Prozessoren (Graphical Processing Unit, GPU) zur Beschleunigung von rechenintensiven Programmen. Um die Performanz weiter zu erhöhen, können mehrere Server über ein leistungsstarkes lokales Netzwerk zu einem Cluster (engl. für Anhäufung) zusammengeschaltet werden. Der Zusammenschluss mehrerer Cluster führt schließlich zum Grid- oder Cloud-Computing, bei dem die Nutzer über das Internet auf extrem performante Parallelrechnersysteme zugreifen können.

Je nach Anwendungsbereich stehen verschiedene Entwurfsziele im Vordergrund. So sollen z.B. mobile Systeme möglichst wenig Energie verbrauchen, um die Akkulaufzeiten zu verlängern. Im Gegensatz dazu haben PCs und Server einen deutlich höheren Energieverbrauch, da es hier vor allem auf eine hohe Performanz ankommt.

1.3 Teilbereiche der Technischen Informatik

Die Technische Informatik umfasst die Teilbereiche

- Hardware-Entwurf (engl. digital design),
- Rechnerarchitektur (engl. computer architecture),
- Messen, Modellieren und Bewerten,
- Echtzeit- und eingebettete Systeme,
- Betriebssysteme, und
- Rechnernetze und Verteilte Systeme.

wobei die Betriebssysteme in natürlicher Weise eine Schnittstelle zwischen Praktischer und Technischer Informatik bilden

1.3.1 Hardware-Entwurf

Der *Hardware-Entwurf* befasst sich mit der computergestützten Erstellung von Hardware-Schaltungen zur Realisierung von Schaltnetzen, Schaltwerken und Prozessoren.

Die Bandbreite reicht von Überlegungen zur effizienten Realisierung von oft benötigten Schaltungen, zum Beispiel Rechnerarithmetik, über die Eingabe von Schaltungen mittels Hardware-Beschreibungssprachen bis zu Algorithmen zur automatischen Erstellung von Test-Eingabemustern für Mikrochips zur Erkennung von Fabrikationsfehlern.

1.3.2 Rechnerarchitektur

Die *Rechnerarchitektur* befasst sich mit Organisationsformen von Prozessoren, um verschiedene, teilweise in Konflikt stehende Entwurfskriterien wie Leistungsfähigkeit, Ressourcenverbrauch (Leistungsaufnahme, Anzahl Transistoren) und Zuverlässigkeit zu erfüllen.

Die Untersuchungen reichen von der Organisation von Zwischenspeichern zur Erhöhung des Datendurchsatzes über die Anzahl, Auswahl und Steuerung von Ausführungseinheiten für verschiedene Befehle bis zur Struktur von Instruktionssätzen.

1.3.3 Bewertung von Computersystemen

Bei der *Bewertung* von Computersystemen gibt es viele Ansätze auf der Basis von *Modellen* verschiedener Detailgrade sowie durch Ausführung von Anwendungen auf dem realen System. Hierbei darf die *Messung* der relevanten Parameter zur Laufzeit (wie Leistungsaufnahme, Datendurchsatz oder Speicherzugriffszeit) die Ausführung nicht so beeinflussen, dass die Messung verfälscht wird.

1.3.4 Echtzeit- und eingebettete Systeme

Eingebettete Systeme dienen oft der Steuerung von Geräten. Deshalb dürfen Reaktionen des Systems bestimmte Zeiten nicht überschreiten. Zum Beispiel wäre es fatal, wenn die Weitergabe des Kommandos zum Bewegen der Vorderräder beim Steer-by-Wire in Kraftfahrzeugen manchmal erst mit einer Verzögerung von einer halben Sekunde erfolgen würde. *Echtzeitsysteme* erfüllen solche Eigenschaften.

Weiterhin sind solche Systeme oft mobil und daher von einer Batterie oder Akku abhängig, so dass zum Beispiel eine Abwägung zwischen Leistungsfähigkeit und Leistungsaufnahme erfolgen muss.

1.3.5 Betriebssysteme

Ein *Betriebssystem* stellt eine Abstraktionsschicht zwischen einer konkreten Rechnerhardware und Anwendungsprogrammen zur Verfügung. Es ermöglicht beispielsweise, dass mehrere Anwendungsprogramme quasi-gleichzeitig ausgeführt werden, und jedes dabei so ausgeführt werden kann, als ob es alleine auf dem Rechner ausgeführt würde. Gleichzeitig koordiniert und beschränkt es den Zugriff von Anwendungen auf Hardware-Einheiten wie Festplatten oder Netzwerk-Karten.

1.3.6 Kommunikationsnetze und verteilte Systeme

Kommunikationsnetze sorgen dafür, dass verschiedene Rechner miteinander kommunizieren können. Hierbei gibt es eine Fülle von Technologien unterschiedlichster Bandbreite, Verzögerung, oder Übertragungsmedien. Arbeiten mehrere Rechner, die sich an verschiedenen Orten (und in der Regel unter verschiedener administrativer Kontrolle) befinden, an der Lösung eines Problems, so spricht man von einem verteilten System. Mit dem Aufkommen des Grid- und Cloud-Computing hat dieses Feld an Bedeutung zugenommen.

1.4 Einordnung des Moduls Computersysteme

Das Modul *Computersysteme* besteht aus den Kursen *Computersysteme 1* und *Computersysteme 2*.

Der Kurs *Computersysteme 1* bietet eine Einführung in den Hardware-Entwurf. Behandelt werden Schaltnetze, Schaltwerke, komplexe Schaltwerke und die Grundlagen von Mikroprozessoren.

Der Kurs *Computersysteme 2* bietet eine Einführung in die Rechnerarchitektur. Behandelt werden Befehlssatz-Architektur, Mikroarchitektur mit Pipelining, Superskalarität und Sprungvorhersage sowie die Speicherorganisation einschließlich Caches und Externspeichern.

Vielleicht fragen Sie sich, weshalb diese Themen in Ihrem Studiengang vorkommen, wenn Sie doch, wie viele Studierende, eine Tätigkeit in der Software-Entwicklung oder -Administration anstreben. Die kurze, wenn auch wenig motivierende Antwort ist, dass die Empfehlungen der Gesellschaft für Informatik und der Akkreditierungsagenturen einen entsprechenden Studienabschnitt vorsehen. Diese Empfehlungen basieren weniger auf altmodischem Fachverständnis als auf wohlgedachtem Kalkül. Ein großer Teil der heutigen Software läuft nicht auf PCs oder Servern, sondern auf mobilen oder eingebetteten Geräten. Die Software-Entwicklung für solche Geräte verlangt damit auch Kenntnis von deren Besonderheiten (zum Beispiel reduzierte Speichergrößen), um diese berücksichtigen zu können.

Während dies lange als Spezialgebiet galt, führen die Verbreitung von mobilen Geräten (Beispiel App-Entwicklung für Smartphones) sowie die Vision vom *Internet der Dinge*, bei dem letztlich alle Alltagsgeräte computergesteuert sind und über (drahtlose) Netzwerke kommunizieren, dazu, dass immer mehr Software-Entwickler für solche Geräte programmieren.

Auch unabhängig vom Anwendungsfeld und trotz Unterstützung durch Softwareentwicklung-Werkzeuge bedingt eine Performance-Optimierung von Anwendungen in der Regel Kenntnisse der zugrundeliegenden Architektur, zum Beispiel der Cachegröße und -Organisation.

Last but not least: auch wenn Sie derzeit eine konkrete Tätigkeit im Fokus haben, weiß niemand von uns mit Sicherheit, was er oder sie in 10 oder 20 Jahren beruflich machen wird. Hier ist eine gewisse Breite in der Grundlagenausbildung unabdingbar (womit wir wieder bei den Vorgaben der Akkreditierungsagenturen wären).

In diesem Sinne wünschen wir Ihnen viel Freude bei der Bearbeitung des Moduls *Computersysteme*.

Falls wir Sie jetzt sogar neugierig auf Technische Informatik gemacht haben, finden Sie im folgenden Abschnitt Informationen über weitere Module in diesem Bereich.

1.5 Weitere Kurse zur Technischen Informatik

Der Pflichtkurs *Betriebssysteme und Rechnernetze* bietet eine Einführung in die beiden genannten Bereiche. Zu jedem der Bereiche gibt es darüberhinaus ein Wahlpflichtmodul mit vertiefenden Inhalten. Zusätzlich gibt es ein Wahlpflichtmodul *Verteilte Systeme*.

Das Wahlpflichtmodul *PC-Technologie* stellt die Hardware sowie Betriebssystem- und Hardware-nahe Software-Aspekte in PCs vor, und bietet somit eine zu den vorigen Modulen orthogonale Herangehensweise.

Das Wahlpflichtmodul *Parallele Programmierung und Grid-Computing* bietet neben einer Einführung in die Programmierung von Rechnern mit mehreren Prozessoren und verschiedenen Speicherorganisationen auch eine Einführung ins Grid- und Cloud-Computing. Vertiefende Informationen zu diesen Bereichen bietet das Modul *Advanced Parallel Computing*.

Das Wahlpflichtmodul *Virtuelle Maschinen* befasst sich unter anderem mit den Anforderungen an Instruktionssätze und Kontrollmechanismen von Prozessoren, wenn nicht wie bei einem Betriebssystem mehrere Prozesse quasi gleichzeitig mit der Illusion, den Rechner alleine zu nutzen, ausgeführt werden, sondern mehrere *virtuelle* Rechner, von denen jeder sein eigenes Betriebssystem nutzen kann, auf einem physikalischen Rechner ausgeführt werden.

Das Wahlpflichtmodul *Anwendungsorientierte Mikroprozessoren* stellt die

Architektur von Prozessoren vor, die nicht in PCs eingesetzt werden. Mikrocontroller dienen vorwiegend Steuerungsaufgaben und digitale Signalprozessoren dienen vorwiegend der Verarbeitung digitalisierter Signale, zum Beispiel Audio- und Videodaten.

Index

- Adressfeld, 11
- Adressierungsarten, 11
- Assembler, 12
- Ausführungsphase, 8
- Befehlsregister, 8
- Befehlszähler, 8
- Bootstrap Loader, 13
- Cache, 13, 38
- CISC-Prozessoren, 11
- Computer, Grundlagen, 6
- CPU, 6
- Datenbus, 13
- Drei-Adress-Maschine, 27
- Ein-/Ausgabe, 6, 13
- Ein-Adress-Maschine, 27
- Ein-Wort-Befehle, 11
- Exceptions, 21
- Fehlerbehandlung, 21
- fetch, 8
- Firmware, 34
- Flags, 8
- Hauptspeicher, 12
- Holephase, 8
- Indexregister, 11
- Interrupt, 16, 20
 - Anwendungen, 20
 - Beispiel, 26
 - Betriebssysteme, 20
 - Codemethode, 24
 - Ein-/Ausgabe, 20
 - Prioritäten, 24
 - Service-Routine, 22
 - Verarbeitung, 21
 - maskierbarer, 22
 - nicht maskierbarer, 22
 - polling, 23
- Interrupt Handler, 23
- Interrupt-Controller, 24
- Interrupt-Vektor, 21
- Interrupts
 - Abfragemethode, 23
 - Vektormethode, 23
- Leitwerk, 8, 34
 - Mikroprogrammierung, 34
 - Steuerwort-Speicher, 34
- LIFO-Prinzip, 16
- Logische Operationen, 30
- Makroassembler, 19
- Makros, 19
- Memory-Mapped IO, 15
- Mikroassembler, 34
- Mikroprogramm-Steuerwerk
 - Adresserzeugung, 36
 - Folgeadressen, 36
 - Mikrobefehlsformat, 35
 - Mikrooperation, 36
 - Mikroprogrammzähler, 35
 - reagierendes, 35
- Mikroprozessor, 13
- MMU, 38
- Mnemonic, 12
- Null-Adress-Maschine, 27
- Opcode, 6, 8, 11
- Operationscode, 8
- Prioritäten, 22
- Prioritätsencoder, 24
- Prozessor, 6
- Prozessorregister, 15
- RALU, 29
- Rechenwerk, 7
 - Registerarchitektur, 27

- Stackarchitektur, 27
 - Adressregister, 15, 27
 - Datenpfade, 28
 - Datenregister, 15, 27
 - Status-Flags, 31
 - logische Operationen, 30
- Rekursion, 19
- RETI-Befehl, 22
- RISC-Prozessoren, 11
- Schiebemultiplexer, 29
- Schreib-/Lesespeicher, 39
- Segmentregister, 16
- Shifter, 8, 29
- Software-Interrupt, 20
- Speicher, 6, 12, 38
- Organisation, 39
- Stack, 16
- Stackpointer, 16
- Stackpointer, 16
- Status-Flags, 31
- Statusregister, 8
- Steuerbus, 14
- Systembus, 15
- Systembusschnittstelle, 15
- Trap, 21
- Unterbrechung, 16
- Unterprogramm, 16, 17
- CALL-Befehl, 18
 - RETURN-Befehl, 18
 - Zeitbedarf, 19
- Urladeprogramm, 13
- Verschachtelung, 19
- Zwei-Adress-Maschine, 27