

Probeklausur

Technische Grundlagen der Informatik

Universität: Technische Universität Berlin
Kurs/Modul: Technische Grundlagen der Informatik
Bearbeitungszeit: 120 Minuten
Erstellungsdatum: September 19, 2025



Zielorientierte Lerninhalte, kostenlos!
Entdecke zugeschnittene Materialien für deine Kurse:

<https://study.AllWeCanLearn.com>

Technische Grundlagen der Informatik

Aufgabe 1.

(a) Beschreiben Sie die grundlegenden Aufgaben der folgenden Bausteine eines Rechners:

Zentraleinheit (CPU),

Hauptspeicher,

Ein- und Ausgabegeräte.

(b) Erläutern Sie kurz die Rolle des Cache-Speichers zwischen CPU und Hauptspeicher und nennen Sie zwei typische Cache-Ebenen.

(c) Erklären Sie den Unterschied zwischen RAM und ROM und nennen Sie zwei typische Einsatzgebiete.

(d) Was ist ein Bus-System in der Computerarchitektur und nennen Sie zwei gängige Bus-Typen.

Aufgabe 2.

- (a) Nennen Sie die wesentlichen Aufgaben eines Betriebssystems und erläutern Sie kurz, wie sie zusammenwirken.
- (b) Beschreiben Sie den Unterschied zwischen einem Prozess und einem Thread. Welche Vorteile bietet Multithreading?
- (c) Was versteht man unter einem Interrupt? Welche Auswirkungen hat ein Interrupt auf den Ablauf eines Programms?
- (d) Welche Aufgaben erfüllt ein Dateisystem und welche Rolle spielen Dateisystemtreiber im Betriebssystem?

Aufgabe 3.

- (a) Beschreiben Sie grob die Struktur der Netze und verteilter Systeme. Welche Grundidee steht hinter der Kommunikation zwischen Prozessen, die auf verschiedenen Recheneinheiten laufen?
- (b) Erklären Sie das Prinzip des Client-Server-Modells und nennen Sie zwei typische Anwendungsbereiche.
- (c) Welche typischen Probleme verteilter Systeme treten auf und welche allgemeine Strategien gibt es zu deren Bewältigung?
- (d) In einer einfachen verteilten Anwendung soll eine Nachricht von Client zu Server übertragen werden. Beschreiben Sie kurz, wie Zuverlässigkeit und Fehlertoleranz in diesem Kontext erreicht werden können.

Aufgabe 4.

- (a) Skizzieren Sie die grobe Architektur eines Computersystems mit CPU, Cache, Hauptspeicher und Festplatte. Beschreiben Sie in Stichpunkten den Ablauf der Ausführung eines einfachen Programms.
- (b) Erklären Sie den Unterschied zwischen virtuellem Speicher und physischem Speicher. Welche Vorteile ergeben sich daraus?
- (c) Welche grundlegenden Sicherheitsaspekte sind bei Netzwerkanwendungen zu beachten? Nennen Sie zwei zentrale Konzepte.
- (d) Diskutieren Sie kurz, wie Synchronisation und Kommunikation in verteilten Systemen zusammenwirken und welches Problem dabei besonders wichtig zu beachten ist.

Lösungen

Aufgabe 1.

(a) Lösung:

- Zentraleinheit (CPU): Führt die Befehle eines Programms aus, steuert den Ablauf, veranlasst Rechenoperationen und die Bedienung weiterer Bausteine. Typische Bestandteile sind Rechenwerk (ALU), Steuereinheit, Registersatz und der Programmzähler. Die CPU arbeitet gemäß dem Fetch-Decode-Execute-Zyklus und wird durch den Taktgeber synchronisiert.
- Hauptspeicher: Speichert laufende Programme und Daten temporär. Schneller Zugriff wird durch Cache-Ebenen erreicht; der Speicher ist flüchtig (bei Verlust der Stromversorgung gehen Inhalte verloren).
- Ein- und Ausgabegeräte: Dienen der Einbringung von Daten in den Computer (Input) und der Ausgabe von Ergebnissen (Output). Beispiele sind Tastatur, Maus, Monitor, Drucker, Netzwerkkarten, Speicherkarten, Sensoren.

(b) Lösung:

- Cache-Speicher zwischen CPU und Hauptspeicher reduziert die Latenzzeit beim Speicherzugriff, indem häufig gebrauchte Daten/Instruktionen zwischengespeichert werden. Dadurch sinkt die durchschnittliche Zugriffszeit deutlich gegenüber dem direkten Zugriff auf den Hauptspeicher.
- Zwei typische Cache-Ebenen sind:
 - L1-Cache (on-chip, sehr klein, sehr schnell; oft getrennte Instruktion-/Daten-Caches)
 - L2-Cache (größer, etwas langsamer als L1; kann teils extern oder on-chip liegen)

Optional ist ein L3-Cache, der größer und langsamer als L2, aber immer noch schneller als der Hauptspeicher ist.

(c) Lösung:

- RAM (Random Access Memory) ist flüchtiger, read/write-fähig und dient als Arbeits- bzw. Zwischenpuffer für laufende Programme.
- ROM (Read-Only Memory) ist nicht flüchtig oder nicht beschreibbar (in der Praxis oft fest beschreibbare Bereiche oder nur lesbar), dient zur Firmware bzw. grundlegenden Boot-Funktionen.
- Typische Einsatzgebiete:
 - RAM: Arbeitsgedächtnis, Cache-Verwendung, temporäre Datenhaltung während Programmlauf.
 - ROM: Firmware (BIOS/UEFI), Bootloader-Speicher, Mikrocode in eingebetteten Systemen.

(d) Lösung:

- Ein Bus-System in der Computerarchitektur ist eine gemeinsame Kommunikationsstruktur, über die CPU, Speicher und Peripherie Daten, Adressen und Steuersignale austauschen.
- Zwei gängige Bus-Typen (Beispiele):
 - Interner/ Speicherbus (z. B. CPU–Speicher-Verbindung, Adress- und Datenbus innerhalb des Systems)
 - Peripherie-Bus (z. B. PCI/PCIe, USB; extern oder Host-Controller-Bus)

Aufgabe 2.

(a) Lösung:

- Wesentliche Aufgaben eines Betriebssystems (BS) umfassen:
 - Prozess- und Ressourcenmanagement (Planung/Scheduling von Prozessen, Kontextwechsel, Synchronisation, Vermeidung/Deklaration von Deadlocks)
 - Speicherverwaltung (virtueller Speicher, Zuweisung, Paging/Swapping, Schutz der Speicherbereiche)
 - Dateisystemverwaltung (Bereitstellung abstrakter Dateisysteme, Verzeichnisse, Dateispeicher, Dateizugriffskontrollen)
 - Gerätemanagement und Treiber (Ansteuerung von Geräten, Pufferung, Interrupt-Verarbeitung)
 - IPC (Interprozesskommunikation) und Systemaufrufe (Ausschnittstelle zwischen Anwendungen und BS)
 - Sicherheit und Zugriffskontrollen (Benutzerauthentifizierung, Berechtigungen, Schutz gegen Missbrauch)
 - Benutzerschnittstelle (Shell/GUI), Fehlerbehandlung und Systemüberwachung
- Zusammenspiel: Der BS koordiniert Rechenzeit, Speicherraum, Dateizugriffe und I/O, bietet Abstraktionen (Prozess, Datei, Device), reagiert auf Ereignisse (Interrupts) und sorgt für Stabilität und Sicherheit trotz Mehrbenutzer- bzw. Mehrprogramm-Betrieb.

(b) Lösung:

- Ein Prozess besitzt einen eigenen Adressraum; Threads eines Prozesses teilen sich diesen Adressraum und andere Ressourcen des Prozesses.
- Vorteile von Multithreading:
 - Geringerer Kontextwechsel-Overhead im Vergleich zu Prozesswechseln
 - Einfachere Kooperation zwischen Threads, schnellere Reaktion auf Ereignisse
 - Bessere Ausnutzung von Mehrkern-/Mehrprozessorsystemen durch parallele Ausführung von Threads
- Wichtige Hinweise: Synchronisation, Race-Conditions, Deadlocks; geeignete Abstraktionen (Locks, Semaphoren, Barriers, Monitore) erforderlich.

(c) Lösung:

- Interrupts sind asynchrone Signale an die CPU, die die aktuelle Ausführung unterbrechen und in einen Interrupt-Service-Routine (ISR) wechseln lassen.
- Auswirkungen auf den Ablauf eines Programms:
 - Kontextwechsel: aktuelle Instruktionsfolge wird angehalten, Zustand wird gespeichert, ISR wird ausgeführt

- Nach Ablauf des ISR kehrt der Prozessor zum unterbrochenen Zustand zurück (falls vorgesehen)
- I/O-Operationen können so schneller reagieren, aber Synchronisationsprobleme (z. B. reentrante Routinen) müssen vermieden werden

(d) Lösung:

- Aufgaben des Dateisystems: Organisation der Speicherung von Dateien, Verzeichnissen und Metadaten; Zuweisung von Speicherplätzen; Verwaltung von Zugriffsrechten; Bereitstellung konsistenter Benutzerschnittstellen.
- Rolle der Dateisystemtreiber: Implementieren die plattform- und gerätespezifische Anbindung an zugrunde liegende Speichermedien; Übersetzen abstrakter Dateisystemoperationen (open/read/write) in konkrete Block-I/O-Anfragen; Caching/Buffering-Strategien sowie Fehlerbehandlung.

Aufgabe 3.

(a) Lösung:

- Netze und verteilte Systeme bestehen aus mehreren Recheneinheiten, die über Kommunikationskanäle verbunden sind. Typische Strukturmodelle reichen von Schichten/Protokollen (z. B. TCP/IP) bis zu RPC-/Nachrichten-Passing-Paradigmen.
- Grundidee hinter der Prozesskommunikation über verschiedene Recheneinheiten ist die Koordination und den Austausch von Daten/Anweisungen durch Nachrichten oder entfernte Aufrufe.

(b) Lösung:

- Prinzip des Client-Server-Modells: Clients senden Anfragen an Server, der Ressourcen oder Dienste bereitstellt; der Server reagiert daraufhin entsprechend.
- Zwei typische Anwendungsbereiche:
 - Webdienste (HTTP-Server) und Webanwendungen
 - Dateiserver/Datenbank-Server (z. B. SQL-Datenbank, Datei-Repository)

(c) Lösung:

- Typische Probleme verteilter Systeme: Latenz, Fehlertoleranz bei Ausfällen/Partitionen, inkonsistente Zustände, Synchronisation von Uhren, heterogene Umgebungen, Sicherheit und Vertrauen.
- Allgemeine Strategien zur Bewältigung:
 - Replikation von Daten, Caching, Timeouts, Retry-Logik
 - Konsensusprotokolle (z. B. Paxos/Raft) für konsistente Entscheidungen
 - Ereignisgesteuerte Kommunikation, idempotente Operationen, Versions- bzw. Sequenznummern

(d) Lösung:

- Zur Zuverlässigkeit und Fehlertoleranz wird typischerweise Folgendes eingesetzt:
 - Transportprotokoll mit Zuverlässigkeit (z. B. TCP) inklusive Bestätigungen, Zeitüberschreitungen und Wiederholversuchen
 - Anwendungen erreichen Zuverlässigkeit auf Protokollebene durch deduplizierende Mechanismen, idempotente Operationen und eindeutige Message-IDs
 - Zusätzlich können Messaging-Systeme (Queue- oder Publish-Subscribe-Systeme) genutzt werden, um lose Kopplung und Wiederholbarkeit zu unterstützen

Aufgabe 4.

(a) Lösung:

- Grobe Architektur: CPU, Cache, Hauptspeicher, Festplatte.
- Ablauf der Programmausführung (stichpunktartig):
 - Programm wird aus dem Dateisystem geladen und in den Hauptspeicher gebracht
 - CPU holt Anweisungen aus dem Cache; bei Cache-Misses werden Daten aus dem Hauptspeicher nachgeladen
 - Die Steuereinheit decodiert Befehle, die ALU führt Berechnungen durch
 - Speicherzugriffe erfolgen über den Adress- und Datenbus, ggf. mit Paging/virtuellem Speicher
 - I/O-Operationen werden asynchron durch Interrupts gemeldet; Betriebssystem koordiniert Pufferung und Scheduler

(b) Lösung:

- Virtueller Speicher vs. physischer Speicher:
 - Virtueller Speicher ermöglicht abstrakte, prozessspezifische Adressräume, Schutzgrenzen, größere Adressräume als der physische Speicher und einfachere Multiplayer-Speicherverwaltung
 - Physischer Speicher bezeichnet den tatsächlichen physischen RAM
- Vorteile des virtuellen Speichers:
 - Prozessisolierung und Sicherheit (Schutzgrenze zwischen Prozessen)
 - Größerer Adressraum pro Prozess, auch wenn der physische Speicher begrenzt ist (Overcommitment mit Paging/Swapping)
 - Erlaubt Programmwechsel und speicherfreundliche Optimierungen (z. B. Demand Paging)

(c) Lösung:

- Zentrale Sicherheitsaspekte bei Netzwerkanwendungen (mindestens zwei):
 - Authentifizierung und Zugriffskontrolle (Wer darf was tun?)
 - Vertraulichkeit und Integrität der Kommunikation (Verschlüsselung, z. B. TLS; Integritätsprüfungen)

(d) Lösung:

- Synchronisation (Koordination) und Kommunikation in verteilten Systemen beeinflussen sich gegenseitig in mehreren Bereichen (Latenzen, Konsistenz, Fehlertoleranz).

- Besonderes Augenmerk liegt auf dem Problem der Konsistenz und der Reihenfolge von Nachrichten (Ordering) bei verteilten Replikaten; Lösungen nutzen logische/ physische Uhren, Vector Clocks, und Konsensprotokolle (z. B. Paxos/Raft) sowie CAP-Betrachtungen (Konsistenz vs. Verfügbarkeit vs. Partitionstoleranz).