

# Betriebssysteme und Anwendersoftware (Fragenausarbeitung)

## KAPITEL 1 - Informatik -

---

### FRAGE 1.1 : Woraus entstand die Informatik?

#### Antwort:

Früher war noch kein Platz für diese neue Wissenschaft, da die Algorithmen aus der Mathematik bekannt waren und die ersten programmierbaren Maschinen nur für eine sehr eingeschränkte Problemklasse verwendet wurden. Elektrotechnik und Mathematik Kenntnisse waren ausreichend. Die Programmierer waren gleichzeitig Benutzer und Erbauer der Maschinen. Erst später, als die sog. „Stored Program Machines“ entwickelt wurden, die universell einsetzbar waren und auch unmittelbare praktische Auswirkungen hatten, ergab sich der Bedarf nach Informatik.

---

### FRAGE 1.2 : Womit beschäftigt sie sich?

#### Antwort:

1. Mit Struktur, Wirkungsweise, Fähigkeiten,.. von Informationsverarbeitungssystemen
2. Mit Strukturen, Eigenschaften und Beschreibungsmöglichkeiten von Informationen und Informationsverarbeitungsprozessen.
3. Mit Strukturierung und Mathematisierung von Anwendungsgebieten sowie mit der Simulation und Modellbildung.

Sie beschäftigt sich also insbesondere mit Computerstrukturen, Programmstrukturen und Datenstrukturen.

---

### FRAGE 1.3 : Was braucht ein Rechenprozess, damit er ablaufen kann?

#### Antwort:

Es sind prinzipiell drei Dinge erforderlich: Computer, Daten und Programm



---

## FRAGE 1.4: Was sind „Fixed Program Machines“ und „Stored Program Machines“ ?

### Antwort:

„Fixed Program Machines“:

- Das Programm ist nicht in der Maschine gespeichert, sondern stellt eine Art Schaltung dar
- Die Algorithmen sind aus der Mathematik bekannt. (keine Informatikkenntnisse erforderlich)
- Konstrukteure waren gleichzeitig Benutzer und Programmierer
- Programme nur für eingeschränkte Problemklasse; keine unmittelbaren Auswirkungen auf die Praxis

„Stored Program Machines“:

- Programm wie Daten in Maschine gespeichert (von Neumann); wird unter Zuhilfenahme eines anderen Programms eingelesen (Betriebssystem)
- Hardware nur Teil der gesamten Problemlösung. Geeignete Algorithmen müssen erst entwickelt werden. (durch Informatik)
- Maschinenhersteller, Benutzer und Programmierer sind verschiedene Menschen => Dokumentationen notwendig
- Programme haben praktische Auswirkungen auf das Leben.

---

## FRAGE 1.5: Wozu braucht man einen Algorithmus?

### Antwort:

Da der Computer über keinerlei Intelligenz verfügt, muss die Problemlösung durch einen Algorithmus angegeben werden. Dieser Algorithmus besteht aus einzelnen, primitiven und für den Rechner verständlichen Anweisungen.

Man muss sich also vor der Programmierung auf allgemeiner Stufe ein „mechanisches“ Verfahren zur Lösung des Problems überlegen.

---

## FRAGE 1.6: Eigenschaften des Algorithmus?

### Antwort:

- Verfahren muss durch endlichen Text beschreibbar sein. (also nicht:  $1 + \frac{1}{2} + \frac{1}{3} \dots$ )
  - Läuft in wohldefinierten, einzelnen Schritten mit eindeutiger Reihenfolge ab.
  - Besitzt Parameter (Eingabegrößen) und kommt nach endlich vielen Rechenschritten zu einem eindeutigen Ergebnis.
-

---

## FRAGE 1.7: Unvollständigkeitssatz von Gödel?

### Antwort:

Früher dachte man, dass jedes mathematisch exakt formulierte Problem auch algorithmisch lösbar ist. In den 30er Jahren beschäftigte man sich im Zuge der mathematischen Logik intensiver mit dem Begriff des Algorithmus. Es ergab sich die überraschende Erkenntnis, dass sich bestimmte Probleme prinzipiell nicht algorithmisch lösen lassen.

Der Unvollständigkeitssatz von Gödel war der erste Unlösbarkeitsbeweis. Die Aussage dieses Satzes ist, dass es für jedes formale System zur Beschreibung der Arithmetik wahre Aussagen über Zahlen gibt, die sich innerhalb des Systems nicht beweisen lassen.

---

## KAPITEL 2 - Algorithmen und Spezifikationen -

---

### FRAGE 2.1: Wie lautet die Spezifikationsregel?

#### Antwort:

Vor der Entwicklung eines Algorithmus ist eine funktionale Spezifikation zu erstellen.

Bestandteile einer funktionalen Spezifikation:

- Menge der gültigen Eingabewerte (Definitionsbereich)
- Menge der möglichen Ausgabewerte (Wertebereich)
- Funktionaler Zusammenhang zwischen Definitions- und Wertebereich

Spezifikation notwendig, da durch Abstraktionsschritte der Bezug zu real existierenden Gegenständen verloren geht. =>genaue Angabe der zulässigen Eingabewerte und möglichen Ausgabegrößen erforderlich (siehe Parkplatzproblem)

---

### FRAGE 2.2: Was ist ein Datentyp?

#### Antwort:

Ein Datentyp ist eine Menge (Trägermenge) zusammen mit einer bestimmten Menge von Funktionen (Operationen des Datentyps), in deren Definitions- oder Wertebereich diese Menge auftritt.

Bezeichnung:  $D = \langle M; f_1, \dots, f_n \rangle$

---

### FRAGE 2.3: Was ist für eine Spezifikation erforderlich?

#### Antwort:

Funktionaler Zusammenhang zwischen Definitions- und Wertebereich ist implizit gegeben durch Vor- und Nachbedingung. Bei Angabe einer Spezifikation sind folgende grundlegende Punkte notwendig: Eingabe, Vorbedingung, Ausgabe und Nachbedingung.

Die Nachbedingung darf man nicht in die Vorbedingung aufnehmen, da ein Programm „robust“ sein soll. D.h. es soll erst mit der Ausführung beginnen, wenn die Vorbedingungen überprüft worden sind. (Um Nachbedingungen überprüfen zu können, muss die Berechnung bereits beendet sein)

Eine Spezifikation besteht nun aus folgenden Komponenten:

- Deklarationen: (Vereinbaren der problemspez. Objekte)
  1. Konstanten: Vereinbaren Konstantenbezeichner für problemspez. Parameter

2. Typen: Vereinbaren problemspez. Datentypen (legen Typenbezeichner für die Trägermengen und Funktionsbezeichner für die Operationen fest)
  3. Funktionen: Bezeichner für weitere problemspez. Funktionen
  4. Prädikate: Bezeichner für spez. Funktionen, die Wahrheitswert zurückgeben
- Eingabe: Vereinbart Variablenbezeichner für Eingabeparameter der Spezifikation und ordnet diesen Parametern bestimmte Gegenstandsbereiche (Typen) zu.
  - Vorbedingung: Menge von Prädikaten, welche die Eingabe erfüllen soll.
  - Ausgabe: siehe Eingabe, jedoch für Ausgabeparameter
  - Nachbedingung: Menge von Prädikaten, welche die Leistung des zu entwickelnden Algorithmus beschreibt.
- 

**FRAGE 2.4: Was ist ein Algorithmus und welche Bedingungen müssen für ihn erfüllt sein?**

**Antwort:**

Ein Algorithmus ist eine Menge von Regeln für ein Verfahren, um aus gewissen Eingabegrößen bestimmte Ausgabegrößen herzuleiten.

Bedingungen für Algorithmus:

- Finitheit der Beschreibung: muss durch endlichen Text beschreibbar sein.
  - Effektivität: Jeder Schritt des Algorithmus muss auch tatsächlich ausführbar sein.
  - Terminiertheit: Verfahren kommt nach endlich vielen Schritten zu einem Ende.
  - Determiniertheit: Ablauf ist zu jedem Punkt eindeutig vorgeschrieben.
- 

**FRAGE 2.5: Unterschied zwischen Effektivität und Effizienz?**

**Antwort:**

Effizienz: wirtschaftlich vernünftige Machbarkeit

Effektivität: prinzipielle Machbarkeit

---

**FRAGE 2.6: Was ist eine abstrakte Maschine?**

**Antwort:**

Ein Mechanismus mit folgenden Eigenschaften:

- Kann Operationen auf Datenobjekte ausführen
- Arbeitet mit endlicher Anzahl von Typen

- Kann begrenzte Anzahl von Objekten speichern
- Ist in der Lage, einen Algorithmus auszuführen
- Nimmt Datenobjekte als Eingabe an und kann auch solche Objekte als Ausgabe liefern.

Beispiele: Computer, klassische Rechenmaschine, Spielautomat, Sachbearbeiter,...

---

---

## FRAGE 2.7: Methoden zum Algorithmenentwurf?

### Antwort:

Algorithmenentwurf entspricht der Konstruktion einer Folge von abstrakten Maschinen.

Die Spezifikation beschreibt die sogenannte Quellmaschine. (sie hat die Fähigkeit, bestimmte Eingaben, welche die Vorbedingungen erfüllen in Ausgaben, welche die Nachbedingungen erfüllen, zu transformieren)

Eine reale Maschine auf welcher der Algorithmus ablaufen soll wird als Zielmaschine bezeichnet.

Unter Verwendung von Zwischenmaschinen wird die Arbeit des Algorithmenentwurfes in überschaubare Teile zerlegt. Man unterscheidet generell zwei Arten des Entwurfes:

- **Top-Down-Methode:** Man geht von der Quellmaschine aus und entwickelt abstrakte Zwischenmaschinen  $A_1, A_2, \dots$  die immer näher in Richtung der Zielmaschine gehen, bis diese erreicht ist.  
Man beginnt also mit der funktionalen Spezifikation und teilt das Problem in Unterprobleme auf, für die man dann wieder funktionale Spezifikationen aufstellt.  
Danach wird beschrieben, wie sich diese Teillösungen zur Gesamtlösung kombinieren lassen. Abschließend müssen nach dem gleichen Schema noch Unteralgorithmen für die nichtelementaren Unterprobleme entworfen werden. (Vorsicht: Fähigkeiten der Zielmaschine beachten)
- **Bottom-Up-Methode:** Hier geht man von der Zielmaschine aus und entwickelt abstrakte Maschinen  $A_1, A_2, \dots$ , von denen man vermutet, dass sie zur Lösung des Problems beitragen könnten, bis man bei der Quellmaschine angekommen ist.  
Man setzt also nach und nach kompliziertere abstrakte Maschinen auf die Zielmaschine auf. (Vorsicht: Quellmaschine im Auge behalten)

---

## FRAGE 2.8: Wie lautet die Regel vom Testen?

### Antwort:

Durch Testen kann man nur die Anwesenheit von Fehlern nachweisen, nicht aber deren Abwesenheit.

Nur ein erschöpfender Test (Ausprobieren aller möglichen Kombinationen von Eingabewerten) kann die Korrektheit eines Programms beweisen. Da aber der Definitionsbereich meist unüberschaubar groß ist => fast immer unmöglich.

---

## FRAGE 2.9: Unterschied zwischen Korrektheit und partieller Korrektheit?

## Antwort:

**Korrektheit:** Ein Algorithmus heißt korrekt, falls für jede Eingabe, welche die Vorbedingung erfüllt, die Ausführung des Programms zu einem Ende kommt und die Ausgabe die Nachbedingung erfüllt.

In der Praxis ist es aber günstiger, die Frage der Terminiertheit und der Einhaltung der Nachbedingungen voneinander zu trennen. Man kommt dadurch zum Begriff der partiellen Korrektheit.

Man zerlegt das Verfahren in einzelnen Anweisungen und fügt vor und nach jeder Anweisung geeignete Vor- und Nachbedingungen ein. Kann man zeigen, dass alle dieser einzelnen Anweisungen bezüglich der gestellten Bedingungen partiell korrekt sind, so kann man auf die Korrektheit des gesamten Verfahren schließen. (nicht aber auch die Terminiertheit!)

Partielle Korrektheit bezieht sich also auf einzelne Anweisungen und entspricht der Definition der Korrektheit, wenn man sich nur den Algorithmus mit Vor- und Nachbedingungen als einzelne Anweisung vorstellt.

---

## KAPITEL 3 - Die Entwicklung von Software -

---

FRAGE 3.1: Erläutern Sie das Phasenmodell der Software-Entwicklung.

Antwort:

Den gesamten Vorgang von der Problemanalyse bis zur Wartung der Programme nennt man Software-Entwicklung. Der Entwicklungsprozeß besteht aus einer Reihe von Einzeltätigkeiten. Diese zeitlichen Abschnitte nennt man Phasen. Im Phasenmodell sind die Hauptaktivitäten der Software-Entwicklung dargestellt.

Es besteht aus folgenden Punkten:

- Analyse: Das zu lösende Problem muss möglichst genau beschrieben werden. Daraus resultiert das sogenannte Pflichtenheft. (verbindliche Grundlage für die Abnahme der erstellten Software)
- Planung: Zeitplan erstellen, Hilfsmittel abschätzen, Arbeitseinteilung vornehmen
- Entwurf: Modell des Gesamtsystems entwickeln und schrittweise konkretisieren (Top-Down-Methode). Aufteilung des Systems in einzelne Module mit genauer Spezifikation (Schnittstellen, Funktion). Letzter Schritt mündet in die Implementierung.
- Implementierung: Kontrollstrukturen und Datentypen festlegen und Quellprogramm erstellen. Ergebnis: dokumentiertes Programm.
- Installation, Abnahme: Übertragung des Software-Produktes auf das Zielsystem. Abnahme durch Auftraggeber, wenn das Pflichtenheft erfüllt ist.
- Verwendung
- Wartung: Fehler werden korrigiert; je nach Wunsch werden Veränderungen und Erweiterungen vorgenommen; ev. Austausch bestimmter Programmteile oder Algorithmen durch effizientere. Nach Wartungsarbeiten: alle Test wiederholen!

---

FRAGE 3.2: Was ist ein Datenflussplan/Flussdiagramm/Struktogramm?

Antwort:

Alle diese Begriffe sind Hilfsmittel für die praktische Implementierung von Software.

Datenflussplan: skizziert Ein/Ausgabevorgänge beim Computer und verwendet Sinnbilder für die Bearbeitung und die Datenträger, sowie Flusslinien.

Flussdiagramm (od. Programmlaufplan): besteht aus Sinnbilder für Operationen, Ein/Ausgabe und Ablauflinien. Es werden alle möglichen Ausführungswege sichtbar gemacht.

Struktogramm: verwendet keine Ablauflinien! Ansonsten hat es aber gleichen Zweck wie Flussdiagramm.

---

### FRAGE 3.3: Was ist eine Programmiersprache?

#### Antwort:

Eine Programmiersprache ist die Gesamtheit aller zulässigen Anweisungsarten und Informationshinweise, mit denen der Programmierer den Computer befähigt, anstehende Aufgaben zu lösen.

---

### FRAGE 3.4: Was ist eine Maschinensprache?

#### Antwort:

Ist eine Programmiersprache, die aus Befehlen besteht, die der Computer unmittelbar versteht. Die Befehlsstruktur ist deshalb von der Architektur des Rechners abhängig. Die Befehle bestehen aus Operationsteil und Adressteil und werden in binärer Form geschrieben. Der Programmierer muss die Speicheraufteilung selbst vornehmen.

---

### FRAGE 3.5: Was ist eine Symbolsprache und was sind die Nachteile der Programmierung in dieser?

#### Antwort:

Sie gehört zu der Gruppe der Maschinensprachen und verwendet symbolische Operationsteile sowie dezimale Absolutadressen oder sogar symbolische Adressen (Labels).

Bei der Symbolsprache mit mnemotechnischem Operationsteil werden „sprechende“ Buchstabenkombinationen als symbolischen Operationsteil verwendet. Die Adressen werden noch dezimal angegeben => Speichereinteilung und Zuordnung der Daten zu den Adressen muss der Programmierer vornehmen.

Nachteile der Symbolsprachen mit mnemotechnischen Operationsteil:

- Adressteil schwer merkbar (besteht nur aus Ziffern)
- Mangelhafte Übersichtlichkeit, schlechte Lesbarkeit, großer Schreibaufwand, erhöhte Fehlerwahrscheinlichkeit,...
- Durch Befehlseinschübe ändert sich die gesamte Speicheraufteilung!
- Lange Befehlsliste macht das Programmieren sehr gedächtnisaufwendig.

- Programme sind nicht portierbar.
-

---

### FRAGE 3.6: Was ist ein Speicherverteilungsplan?

#### Antwort:

Er legt die Zuordnungen der Variablen zu den Adressen der Speicherzellen fest und gibt auch die Position des Programms im Speicher an. Er ist ein wichtiges Hilfsmittel zur Erstellung von Programmen in einer symbolischen Maschinensprache. (nicht für Assembler! ; diese Sprache gehört zur zweiten Generation der Programmiersprachen und verwendet bereits symbolische Adressen-Labels)

---

### FRAGE 3.7: Was ist ein Assembler?

#### Antwort:

Bei der Assemblersprache wird der Adressteil des Befehlswortes durch alphanumerische Symbole dargestellt (symbolische Adressierung). Auch die Sprungziele werden durch symbolische Adressen (Labels) gekennzeichnet. => anwendungsfreundlicher als reine Maschinensprache (übersichtlicher, Befehle leichter zu merken, Speicheraufteilung erfolgt automatisch, Adressen leicht zu merken, Einfügen von Befehlen leicht möglich,...)

---

### FRAGE 3.8: Unterschied zwischen Makro und Unterprogramm?

#### Antwort:

Unter Verwendung von Makros gelangt man von einer 1:1-Sprache (ein Befehl im Quelltext entspricht einem Maschinenbefehl) zur 1:n-Sprache. Ein Makro ersetzt eine ganze Kette von Maschinenbefehlen. Ein Makro kann im Programm definiert werden oder als Systemmakro aus Makrobibliotheken importiert werden. Vor dem Übersetzen wird der Makroumwandler tätig. Er ersetzt alle Makros durch die entsprechende Befehlskette.

Im Gegensatz zum Makro kann ein Unterprogramm gewisse Eingabe- und Ausgabeparameter haben (Schnittstellen der Module). Ein Makro ist nur eine Befehlsfolge, wogegen ein Unterprogramm ein eigenes Unterproblem des Algorithmus behandelt.

Unterprogramm kann auch mehrmals aufgerufen werden (eigener Programmteil).

---

### FRAGE 3.9: Problemnahe und problemorientierte Programmiersprachen?

#### Antwort:

Bei problemorientierten Programmiersprachen (höhere Programmiersprachen) steht die Programmierung eines bestimmten Computers im Vordergrund, sondern die Programmierung einer Lösung eines bestimmten Problems im Vordergrund. Man muss sich keine

Gedanken über die Möglichkeiten der jeweiligen Maschinensprache machen, sondern kann sich auf das Problem konzentrieren. Solche Sprache haben einen hohen Grad der Verdichtung und verwenden eine fast umgangssprachliche Form. Für jeden Maschinentyp ist ein eigener Compiler erforderlich.

---

### FRAGE 3.10: Welche Aufgaben hat der Compiler?

#### Antwort:

Er übersetzt das entstandene Quellprogramm in ein maschinenorientiertes Objektprogramm. Diese Übersetzung kann auch in zwei Schritten erfolgen (zuerst in Assemblerprogramm und danach durch Assembler in Maschinenprogramm)

Er hat folgende Aufgaben:

- Kontrolle auf Syntaxfehler
  - Teilweise Kontrolle auf Semantikfehler
  - Übersetzen des Quellprogramms in provisorisches Zielprogramm (eventuell in einer Zwischensprache)
  - Erstellen einer optimalen Form des Zielprogramms
- 

### FRAGE 3.11: Was ist eine Zwischensprache?

#### Antwort:

### FRAGE 3.12: Was sind Syntax- und Semantikfehler?

#### Antwort:

Ein Syntaxfehler eines Programms steht gewissermaßen für einen grammatikalischen Fehler im Quelltext. (z.B.: Befehl falsch geschrieben, Strichpunkt vergessen,...)  
Ein Semantikfehler kennzeichnet einen logischen Prinzipfehler.

---

### FRAGE 3.13: Unterschied zwischen Compiler und Interpreter?

#### Antwort:

Beide Begriffe bezeichnen Übersetzungsprogramme. Der wesentliche Unterschied liegt aber darin, dass der Interpreter ein Programm, welches nicht in

Maschinensprache formuliert wurde, ausführen kann, ohne es vorher zur Gänze übersetzt zu haben. Das Übersetzen in Maschinencode erfolgt zur Laufzeit. (also Befehl für Befehl) Dies bedeutet natürlich eine langsamere Ausführungsgeschwindigkeit der Programme.  
Der Compiler übersetzt Quelltext vollständig in Maschinencode und erzeugt ein ausführbares Programm.

---

### FRAGE 3.14: Anwendung von Programmiersprachen?

#### Antwort:

Ein Programm richtet die universellen Möglichkeiten des Computers auf die Lösung eines Problems aus. Man erkennt in den letzten Jahren zwei widersprüchliche Tendenzen: einerseits werden problemorientierte Programmiersprachen laufend weiterentwickelt um in sämtliche Problembereiche (technisch-wissenschaftlich/kaufmännisch) universell einsetzbar zu sein. Andererseits entstehen Programmiersprachen für spezielle Anwendungsbereiche (z.B.: Datenbankdialog, Simulationen und Produktionssteuerung)

---

### FRAGE 3.15 Unterschied von prozeduralen und nichtprozeduralen Hochsprachen?

#### Antwort:

Prozedurale Programmiersprachen verwenden zur Problemlösung feste Strukturen (Schleifen, Verzweigungen, Prozeduren, Funktionen, usw.). Es steht

---

### FRAGE 3.15: Unterschied von prozeduralen und nichtprozeduralen Hochsprachen?

#### Antwort:

Prozedurale Programmiersprachen verwenden zur Problemlösung feste Strukturen (Schleifen, Verzweigungen, Sprunganweisungen, usw.).

Die Programmiersprachen der 4. Generation lösen sich weitgehend von den datenverarbeitungstechnischen Prozeduren=> nichtprozedurale Sprachen.

Objektorientierte Programmierung kann als Ablöse des prozeduralen Programmierens angesehen werden. Man versteht darunter eine Technik, die sich auf Daten (Objekte) und die Schnittstellen zwischen ihnen konzentriert.

---

### FRAGE 3.16: Einteilung der Programmiersprachen in Generationen?

#### Antwort:

Gen	Sprache	Beispiel	Typische Merkmale
-----	---------	----------	-------------------

1	Maschinensprachen	Maschinencode	Maschineninterne Operationsabläufe/Speicheraufteilungen
2	Assemblersprachen	Für alle Computer	Symbolische Adressierung, maschinenabhängige OP-Symbole
3	Problemorientierte Hochsprachen	ALGO, BASIC, COBOL, Fortran, PL/1,...	Universell für Problem und Computer
3+	Prozedurale Hochsprachen	ADA, C, FORTH, MODULA, PASCAL, ...	Zuweisungsorientiert, komplexe Datentypen
4	Nichtprozedurale Hochsprachen	C++, Java, LISP, PROLOG, ...	Problemlösend, transaktionsorientiert

Sprachen der 4. Generation erlauben Transaktionen (durch Dialogaufruf Daten suchen, ändern, ersetzen, löschen) und schrittweise Problemlösungen (prototyping) Sie bieten eine interaktive Benutzerschicht.

### FRAGE 3.17: Welche konstruktiven Maßnahmen zur Qualitätssicherung kennen Sie?

#### Antwort:

Unter Qualitätssicherung versteht man alle geplanten und durchgeführten Tätigkeiten, die erreichen sollen, dass ein Software-Produkt den vorgegebenen Qualitätsanforderungen entspricht.

Konstruktive Maßnahmen sind Konzepte, Methoden, Sprachen und Werkzeuge, die dafür sorgen, dass das entstehende Software-Produkt a priori bestimmte Eigenschaften besitzt.

#### Beispiele:

- Programmiersprache, die strukturierte Steueranweisungen den Sprüngen bevorzugt (z.B.: Pascal) => i.a. wird dadurch Programm weniger fehleranfällig
- Sprache, die Definition von globalen Variablen nur unter speziellen Umständen ermöglicht => bessere Lesbarkeit; leichteres Wiederverwenden bestehender Module
- Werkzeug, das Schachtelungstiefe von Steuerstrukturen begrenzt
- Programmteile, die abstrakt präzise beschreibbar sind, sollen durch ein Programm automatisch erzeugt werden.

Prinzip der maximalen konstruktiven Qualitätssicherung besagt: „Vorbeugen ist besser als heilen“ oder „Fehler, die nicht gemacht werden, brauchen auch nicht behoben zu werden“

### FRAGE 3.18: Welche analytischen Maßnahmen zur Qualitätssicherung kennen Sie?

#### Antwort:

Analytische Maßnahmen der Qualitätssicherung dienen dazu, um Fehler und Defekte zu lokalisieren. Es handelt sich also um diagnostische Maßnahmen

Dazu zählen:

- Programmverifikation: Ein Programm wird mittels mathematischer Methoden untersucht, ob es seine Spezifikation erfüllt. Problem: für große Programme=> sehr komplexe mathematische Beweise!
- Code-Inspektionen und Walkthroughs: Sind Überprüfungsmethoden ohne Computer (human testing) Im Rahmen einer Sitzung versucht ein Projektteam visuell beim gemeinsamen Lesen des Codes, Fehler zu finden. (es werden ca. 30-70% der logischen Entwicklungs- und Codierungsfehler gefunden)  
Bei der Code-Inspektion erklärt der Programmierer Schritt für Schritt jede einzelne Anweisung. Anschließend wird das Programm mit einer Checkliste üblicher Fehler untersucht.  
Beim Walkthrough spielen die Mitglieder des Teams selber Computer. Sie führen am Papier Testfälle durch und besprechen die Ergebnisse.
- Konventionelles Testen

---

### FRAGE 3.19: Was sind die allgemeinen Prinzipien der Qualitätssicherung?

Antwort:

- Prinzip der frühzeitigen Fehlerentdeckung: Kosten der Fehlerbeseitigung steigen nahezu exponentiell mit dem Zeitpunkt des Fehlererkennens. Durch frühzeitiges Erkennen bieten sich einige Vorteile: Vermeidung von Fehlern in späteren Phasen, höhere Wahrscheinlichkeit einer vollständigen Fehlerkorrektur und Senkung der Entwicklungskosten!
- Prinzip der entwicklungsbegleitenden, integrierten Qualitätssicherung: Ist unbedingt zu beachten, da ansonsten Fehler auf keinen Fall bereits frühzeitig erkannt werden können.
- Prinzip der externen Qualitätskontrolle: Programmierer darf selbst nicht testen
- Prinzip der werkzeugunterstützten Qualitätssicherung: Wirtschaftlichkeit von analytischen u. konstruktiven Maßnahmen der Qualitätssicherung kann durch Werkzeugunterstützung erhöht werden. Auch gute Vorbereitung auf bevorstehende Walkthroughs.
- Prinzip der produktabhängigen Qualitätssicherungsmaßnahmen: Maßnahmen hängen von den Qualitätsanforderungen ab. (vgl.: Software für Autopilot bzw. Software für Wirtschaftsstatistiken). Es muss ein Kompromiss zwischen Nutzen und Kosten der Maßnahmen gefunden werden.

---

### FRAGE 3.20: Welche Arten des Testens bei der Programmentwicklung kennen Sie?

Antwort:

Unter Testen versteht man die Überprüfung des beobachteten Ein/Ausgabeverhaltens von Programme oder Programmteilen. Kein vollständiges Testen möglich! (zu großer Definitionsbereich) Beachte auch Regel vom Testen.

Ein erfolgreicher Test zeigt, welche Eingabedaten das getestete Programm Ausgabedaten liefert, die nicht seiner Spezifikation entsprechen.

Je nach dem Zeitpunkt des Testens bzgl. der Phasen der Entwicklung unterscheidet man:

- **Modultest:** Bereits in der frühen Entwurfsphase werden einzelne Programmmodule überprüft. Dazu müssen sog. Treiber- und Dummymodule (Stub-Module) geschrieben werden. Das Treibermodul versorgt das zu testende Modul mit den notwendigen Eingabedaten. Die Dummy-Module ersetzen ein bereits geplantes Modul, welches Daten von dem zu testenden Modul erhält.
- **Integrationstest:** Sind bereits alle Module einzeln getestet worden, so steht nun der Integrationstest bevor, der das Zusammenwirken aller Module testet. Dabei werden insbesondere die Schnittstellen der Module untersucht (Fehlerquelle Nr.1)
- **Installationstest:** Das ganze Programm wird auf seine Spezifikation hin untersucht. Der Software-Entwickler teilt das Ergebnis dem Auftraggeber mit.
- **Abnahmetest:** Wird von unabhängigen Teams des Auftraggebers durchgeführt und besteht aus mehreren Systemtests an der gelieferten Software.

Generell sind zwei Vorgehensweisen beim Testen möglich:

- **Top-Down-Testen:** Man beginnt mit dem Hauptmodul (oberstes Modul in der Kommunikationshierarchie / wird von keinem anderen Modul aufgerufen). Alle Module, die an das Hauptmodul Daten geben, werden durch Treibermodule ersetzt und alle Module, die Daten vom Hauptmodul erhalten werden durch Dummy-Module ersetzt. Danach kann das Hauptmodul getestet werden. Man geht anschließend in der Hierarchie einen Schritt nach unten und wiederholt diesen Test für das aktuelle Modul. Dabei gilt es zu beachten: Ein-/Ausgabemodule und kritische Module so früh als möglich testen.  
Vorteile: Programmgerüst, das Ein-/Ausg. durchführt liegt sehr schnell vor  
Nachteile: sehr schwierig, für ein Modul bestimmte Testfälle durch Dummy-Module zu kreieren.
- **Bottom-Up-Testen:** Man beginnt mit dem Testen bei den Terminalmodulen (Module, die selbst keine anderen Module aufrufen) Man erstellt für das betreffende Modul ein! (vergl. Baumstruktur) Treibermodul.  
Vorteile: Fehlerlokalisierung wird erleichtert, da man den entdeckten Fehler sofort einem bestimmten Modul zuordnen kann; kombinatorische Schwierigkeiten beim Testen leicht in den Griff zu bekommen.  
Nachteile: während des Tests ist noch kein Programmskelett vorhanden.  
Lauffähiges Programm erst nach dem Testen des letzten Moduls (=Hauptmodul!  
=> kritische Phase)

Testen nimmt etwa 50% der Zeit und über 50% der Kosten einer Software-Entwicklung in Anspruch.

---

## KAPITEL 4 - Betriebssysteme -

---

**FRAGE 4.1:** Erklären Sie anhand der Benutzung eines früheren Computers die Motivation zur Schaffung eines Betriebssystems.

**Antwort:**

Die typische Computerbenutzung gegen Ende der 50er Jahre war für den Anwender sehr umständlich, da dieser den gesamten Arbeitsablauf selbst steuern und zum richtigen Zeitpunkt veranlassen musste.

Wollte der Benutzer ein neu programmiertes Anwendungsprogramm assemblieren und testen, so hatte er in etwa folgende Schritte zu machen: Anlage eine Woche im Voraus reservieren, Assembler mittels Lochstreifen einlesen, das zu übersetzende Anwendungsprogramm. Über Lochstreifen einlesen, Assembler starten und übersetztes Programm als Lochstreifen entnehmen, Standardproz. aus Bibliothek entnehmen, Anwendungsprogramm und Lochstreifen mit Standardprozess. Einlesen, Lochstreifen mit Eingabedaten einlegen, Programm starten, Ergebnisse als Lochstreifen entnehmen, Ausdruckmöglichkeit auf Fernschreibgerät, auftretende Fehler sowie Zeitraum der Nutzung in Logbuch eintragen.

Durch diese enge Einbeziehung des Menschen in den technischen Ablauf ergaben sich folgende Probleme:

- Große Leerzeiten der Maschine
- Hohe Fehleranfälligkeit
- Benutzer muss während der gesamten Auftragsbearbeitung anwesend sein.

Es wurde deshalb daran gearbeitet, diese Abläufe durch den Computer automatisch durchführen zu lassen. So entstanden die ersten Betriebssysteme.

---

**FRAGE 4.2:** Wozu braucht man ein Betriebssystem?

**Antwort:**

- Steuert den Ablauf der Auftragsbearbeitung für den Benutzer.
  - Plant die Reihenfolge der Auftragsbearbeitung für mehrere Benutzer.
  - Lädt Programme in den Hauptspeicher und startet sie.
  - Stellt Standarddienste für den Transport von Daten zwischen Programm und Geräten bereit
  - Koordiniert und synchronisiert den Zugriff auf gemeinsame Betriebsmittel. (wenn mehrere Programme gleichzeitig laufen)
  - Analysiert verbrauchte Rechenleistung und registriert Hardwareprobleme.
-

### FRAGE 4.3: Definieren Sie den Begriff Betriebssystem.

**Antwort:**

Das Betriebssystem ist die Menge aller Programme, die zusammen mit den Eigenschaften des Computers die Basis aller möglichen Betriebsarten bilden. Insbesondere steuern und überwachen diese Programme die Abwicklung anderer Programme.

### FRAGE 4.4: Was ist ein Prozess?

**Antwort:**

Der Ablauf eines Programms wird Prozess genannt, wenn dieser Ablauf durch das Betriebssystem verwaltet wird.

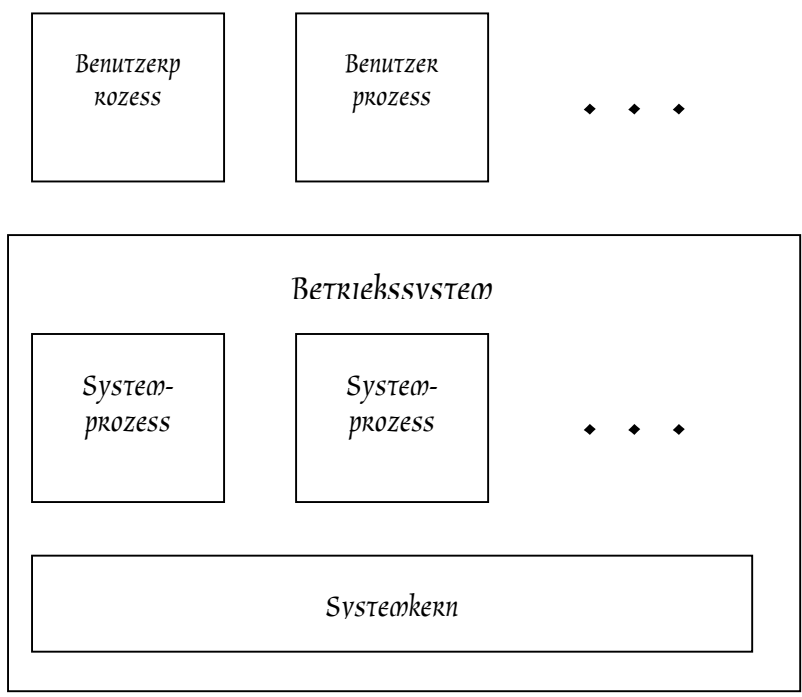
Prozesse können logisch parallel ablaufen (in Mehrprozessorsystemen auch physikalisch)

Nun einige Begriffsfestlegungen:

- Subprozesse: Abläufe, die nicht Verwaltungseinheit des Betriebssystems sind und innerhalb von Prozessen ablaufen.
- Benutzerprozesse: Sind Prozesse, die Aufträge des Benutzers abwickeln
- Systemprozesse: sind Prozesse, die ausgewählte Dienstleistungen des Betriebssystems erbringen.

Systemprozesse können als Komponenten des Betriebssystems betrachtet werden. Alle nicht als Prozesse realisierten Komponenten des Betriebssystems werden als Betriebssystemkern bezeichnet

Es ergibt sich der prinzipielle Aufbau eines Betriebssystems:



---

---

### FRAGE 4.5: Erklären Sie den Begriff Objekt; was sind Objekte?

#### Antwort:

Objekte sind Hardwarekomponenten oder Datenstrukturen, die durch das Betriebssystem verwaltet werden.

#### Begriffsbildungen:

- **Objektklasse/Typ:** wird jedem Objekt zugeordnet und bestimmt die möglichen Operationen auf die zugehörigen Objekte.
- **Zugriffsdienste/Dienste:** bezeichnet diese Operationen auf Objekte
- **Objektmanager/Verwalter:** Komponenten des Betriebssystems, die für einen bestimmten Objekttyp zuständig sind. Sie haben folgende Aufgaben:
  - Führen Buch über Zustand der Objekte des entspr. Objekttyps
  - Stellen Dienste dafür bereit
  - Lösen Zugriffskonflikte und optimieren die Nutzung solcher Objekte

Ein Objektverwalter erbringt seine Dienste unter Zugriff auf Dienste anderer Objektverwalter. Dies ist dann der Fall, wenn ein Objekt einer Objektklasse mit Hilfe einfacherer Objektklassen realisiert wird. (Hierarchische Struktur von Objekten)  
Einfache Objekt: z.B.: durch Hardware realisierte; Komplexe Objekte: z.B.: Dateien!  
Verwalter für die Geräte eines Computersystems werden auch Treiber genannt.  
Betriebsmittel sind die zum Ablauf eines Prozesses benötigten Objekte.

---

### FRAGE 4.6: Was ist eine virtuelle Maschine?

#### Antwort:

Es ist die Aufgabe eines Betriebssystems, eine komfortable und für die Programmierung bequeme Umgebung zu schaffen. Das Betriebssystem stellt also das Bindeglied zwischen Hardware, Programmiersystem und Anwendungsprogrammen dar.

Dem Benutzer wird eine virtuelle Maschine bereitgestellt.

Definition (Virtuelle Maschine): Ist eine Einheit, die sich funktionell wie eine Maschine verhält, deren Befehlsatz jedoch nicht direkt durch die Hardware, sondern durch ein überlagertes Software-System interpretiert wird.

---

### FRAGE 4.7: Was ist das Programmiersystem eines Betriebssystems?

#### Antwort:

Man fasst die anwendungsorientierten Module (Kommando-Interpreter, Compiler, Laufzeitsysteme, Testhilfen für höhere Sprachen usw.) unter dem Begriff Programmiersystem zusammen.

Diese Module betrachtet man nicht als Bestandteil des Betriebssystemkerns.

---

---

## FRAGE 4.8: Was sind Datenobjekte?

### Antwort:

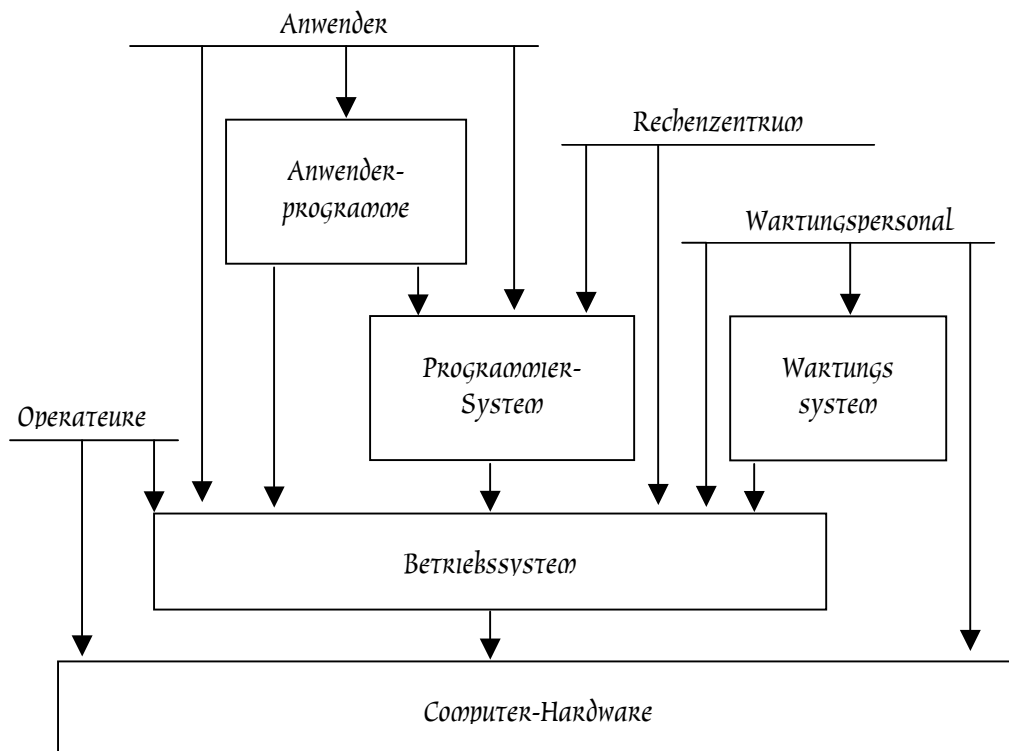
Datenobjekte werden vom Betriebssystem dem Benutzer zur Verfügung gestellt, um eine komfortable Umgebung zu schaffen. Neben den hardwareorientierten Datenobjekte werden auch z.B.: Datenobjekte wie die Datei angeboten. Beim Entwurf eines Betriebssystems ist zu entscheiden, welche Datenobjekte ein Betriebssystem enthalten soll. (Datenobjekte können auch vom Anwenderprogramm oder vom Programmiersystem realisiert werden)

---

## FRAGE 4.9: Schichtenmodell eines Computersystems?

### Antwort:

Man erkennt die Schnittstellen und Zugriffsmöglichkeiten auf das Computersystem durch unterschiedliche Benutzergruppen:



---

## FRAGE 4.10: Welche Aufgaben hat ein Betriebssystem in Hinblick auf Prozess- und Betriebsmittelverwaltung?

### Antwort:

Ein Betriebssystem verwaltet Prozesse. Diese Prozesse benötigen zur Durchführung ihrer Aufgaben Betriebsmittel. Prozesse konkurrieren also um die Betriebsmittel. Das Betriebssystem hat aus der Sicht der Prozesse folgende Aufgaben:

- Erzeugen und Löschen von Prozessen
- Synchronisation des Zugriffs auf die Betriebsmittel
- Zuteilung der Betriebsmittel
- Bereitstellung von Kommunikationsmechanismen zwischen Prozessen.

Weiters werden vom Betriebssystem die Wartezeiten der Prozesse beim Zugriff auf ein Betriebsmittel gemessen und die Auslastung der Komponenten eines Systems (für Tuning)

Auch die einzelnen verbrauchten Rechenleistungen bei mehreren Benutzern werden gemessen und aufgezeichnet. (Basis für Kostenermittlung und Abrechnung)

Die verbrauchte Computerleistung in Betriebssystemen wird oft als Overhead bezeichnet. Typische Werte liegen zwischen 10 und 30% der Gesamtleistung des Computersystems.

---

## FRAGE 4.11: Nennen Sie typische Betriebsziele eines Betriebssystems.

### Antwort:

Das Betriebssystem verteilt die erforderlichen Betriebsmittel nach einer Zuteilungsstrategie auf die einzelnen Prozesse. Bestimmte Betriebsmittel kann ein Prozess nicht beliebig lange behalten (z.B.: Prozessor). Deshalb gibt es auch sogenannte Verdrängungsstrategien. Das Zusammenwirken aller im Betriebssystem vorhandenen Strategien bestimmt die Bearbeitungsfolge und den Fortschritt der Aufträge. Diese Strategien müssen so gewählt werden, dass eine optimale Anpassung an die Betriebsziele des Betreibers des Computers erreicht wird.

Typische Betriebsziele sind:

- Hohe Auslastung der Komponenten
- Gute Reaktionen (akzeptierbare Reaktionszeiten richten sich nach der verlangten Aufgabe)

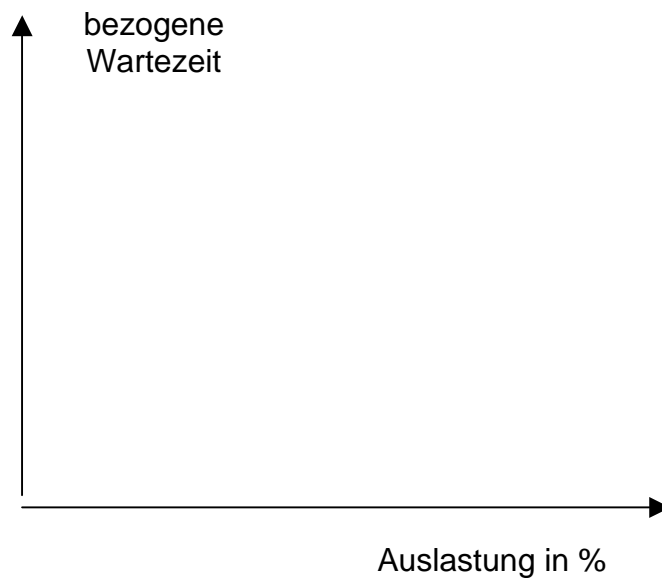
Betriebsziele stehen in Konflikt zueinander!:

- Bevorzugung kleiner Aufträge verursacht eine Behinderung großer Aufträge
  - Gute Auslastung der Komp. Bedeutet schlechte Reaktionszeiten
-

---

**FRAGE 4.12: Erläutern Sie das M/M/1-Modell.****Antwort:**

Ist ein Modell für die Darstellung des Zusammenhanges zwischen Auslastung und Reaktionszeit für ein einfaches Computersystem. Dabei kommen die Aufträge in zufälligen Zeitintervallen in eine Warteschlange und werden nach und nach vom Prozessor bearbeitet (Bearbeitungszeit auch zufällig!). Man spricht dann von einem M/M/1-Modell, wenn die Bearbeitungszeit und die Eintreffintervalle der Aufträge negativ exponentiell verteilt sind.



Die bezogene Wartezeit ist die Zeit zwischen Eintreffen und Beginn der Bearbeitung geteilt durch die mittlere Bearbeitungszeit.

Die Wartezeit geht gegen unendlich, wenn die Auslastung gegen 100% geht. Bereits bei 50% Auslastung ist die bezogene Wartezeit so groß wie die mittlere Bearbeitungszeit.

---

**FRAGE 4.13: Nennen sie einige Grundkonzepte von Betriebssystemen.****Antwort:**

In einem Betriebssystem treten charakteristische Probleme auf. Zur Lösung dieser Probleme entwickelten sich einige Grundkonzepte, die bei der Implementierung des Betriebssystems zu beachten sind:

- Hardware ändert sich im Laufe der Zeit und auch von System zu System => da Betriebssystem direkte Schnittstelle zur Hardware hat ist durch verschiedene

Techniken sicherzustellen, dass das Betriebssystem leicht anzupassen ist. (Interface-Module, Konfigurationstabellen, Virtualisierung)

- Langlebige Datenbestände der Benutzer müssen gegen unberechtigten Zugriff und Verlust geschützt werden.
- Betriebssystem muss eine Hierarchie von Informationsspeichern verwalten (meist dreistufig: Magnetbänder, Plattenspeicher und Hauptspeicher)
- Da das Betriebssystem viele parallel ablaufende Vorgänge steuert, muss es diese Vorgänge bzgl. Des Zugriffs auf Betriebsmittel synchronisiert werden. Auf Anforderungen und Meldungen dieser Prozesse muss schnell reagiert werden (aber abgestuft je nach Wichtigkeit). Abläufe müssen voneinander geschützt werden.
- Es existieren Abbildungen zw. den Eigenschaften virtueller und den der realen Geräte. (obersten Schichten des BS arbeiten mit virtuellen Geräten) Diese Abbildungen sind in Konfigurationstabellen festgelegt. => Benutzer kann leicht ein neues Gerät einer Geräteklasse in das System integrieren.
- Betriebssystem virtualisiert die Hardwarekomponenten, insbesondere die Geräte. Geräte mit ähnlichen Eig. Werden zu einer Klasse zusammengefasst. Virtuelle Geräte heißen auch logische Geräte. Eigenschaften der virtuellen Geräte können Obermenge oder Teilmenge der realen Geräte sein!
- Betriebssystem unterstützt ein breites Spektrum unterschiedlicher Benutzer. Um individuellen Anforderungen der Benutzer gerecht werden zu können, müssen ausreichend parametrisierte Auftragsbearbeitungsstrategien vorgesehen werden.
- Da Betriebssysteme eine sehr lange Lebensdauer haben=> konsequente Modularisierung erforderlich. Systemdienste werden über Makros aufgerufen. Bei Änderung der Systemschnittstelle sollte man die Makros ändern und nicht die Parameterversorgung der bestehenden Makros oder deren Aufruf => Anwenderprogramm muss nicht verändert werden, falls eine neue Version des BS verwendet wird.
- Betriebssystem wird möglichst hardwareunabhängig programmiert. Die verbleibenden hardwareabhängigen Stellen werden in Interface-Module zusammengefasst. Dieses Vorgehen ist dann geeignet, wenn eine Programmänderung und eine anschließende Systemgenerierung zur Anpassung an die Konfiguration akzeptabel ist. Dies wird in der Regel bei Portierung des BS auf neue Computersysteme oder beim Anschluss eines völlig neuen Gerätes der Fall sein.

#### FRAGE 4.14: Was versteht man unter Virtualisieren (Konzepte)?

##### Antwort:

Die reale Computerstruktur wird benutzt, um eine gedachte (virtuelle), benutzerfreundliche Struktur nachzubilden und nach außen zu simulieren.

Verschiedene techn. und organisatorische Konzepte sind dabei entstanden:

- symbolische statt absolute Adressen bei Assemblersprachen
- symbolische Anweisungen bei höheren Programmiersprachen
- logische statt physikalische Datenstrukturen
- virtuelle Bildschirme (Hintergrundbildschirme), die dynamisch mit dem realen Vordergrundbildschirm fortgeschrieben werden.
- virtuelle, physikalisch nicht direkt existierende Leitungsverbindungen zwischen Computern.
- die relative und indizierte Adressierung anstelle der absoluten.
- Firmware statt technischer Schaltwerke

- Systemprogramme zur Ergänzung der Hardware.
  - Nachbilden eines Computersystems mit einem anderen (Emulation)
  - Betriebsarten wie Dialogbetrieb zur Teilung der Betriebsmittel.
- 

#### FRAGE 4.15: Was sind virtueller Speicher, Drucker, Bildschirm?

##### Antwort:

Virtuelle Bildschirme: Hintergrundbildschirme, die dynamisch mit dem realen Vordergrundbildschirm fortgeschrieben werden.

Virtueller Speicher: virtuelle Erweiterung des realen Hauptspeichers mittels Peripheriespeicher.

---

#### FRAGE 4.16: Nennen sie die Konzepte des virtuellen Speichers.

##### Antwort:

- Virtuelle Erw. des realen Hauptspeichers mittels Peripheriespeicher
- Strukturierung eines Kassetten-Massenspeichers als virtueller Plattenspeicher bei der Zusammenarbeit mit realen Plattenspeichern.
- Nachbilden eines Diskettenlaufwerkes im Hauptspeicher (RAM-Disk)

Der erste Fall wird am häufigsten angewendet und bezieht sich auf einen realen Hauptspeicher sowie ein oder mehreren schnellen Peripheriespeichern (z.B.: Plattenspeicher)

---

#### FRAGE 4.17: Was sind Probleme bei virtuellen Datensystemen?

##### Antwort:

Der reale Hauptspeicher ist aufgeteilt in einen Bereich für residente Programme (des Betriebssystems) und einen Bereich, der Programmbausteine und Daten aus dem virtuellen Speicher aufnimmt. Die restlichen Systemprogramme, Anwenderprogramme und Daten befinden sich im virtuellen Speicher.

Dabei sind zwei Probleme zu lösen:

- realer und virtueller Hauptspeicher müssen auf die verschiedenen Aufträge (Jobs) aufgeteilt werden.
  - reale und virtuelle Adressen müssen einander eindeutig zugeordnet werden.
- 

#### FRAGE 4.18: Was ist Paging?

##### Antwort:

Unter Paging (Seitenverfahren) versteht man die Aufteilung des realen- und virtuellen Speichers in gleich große Bereiche (Seiten). Der sog. Seitenrahmen dient als Behälter für die vom virtuellen Speicher übernommenen Seiten. Eine Seite enthält typisch  $2^9$ ,  $2^{10}$ ,  $2^{11}$  oder  $2^{12}$  Bytes. Eine feste Anzahl von Seiten (16 oder 32) des virtuellen Speichers wird zu einem Segment zusammengefasst.

Eine Seitentabelle ordnet den virtuellen Seitenadressen die physischen Adressen des realen Hauptspeichers zu. Eine Markierung in der Tabelle verweist darauf, ob sich die betreffende Seite momentan im realen Hauptspeicher befindet. Da sich diese Zuordnungen laufend ändern, ist die Tabelle ständig zu aktualisieren.

Die Wirkungsweise des virtuellen Speichersystems hängt sehr stark von der Geschwindigkeit des Info-Transportes zwischen realen- und virtuellen Speicher zusammen.

---

## FRAGE 4.19: Welche Strategien zum Seitenwechsel kennen Sie?

### Antwort:

- **Einlagerungsstrategie:**  
Bezieht sich die aktuelle Anweisung eines Programms auf eine Adresse im virtuellen Speicher, so wird vorher nachgesehen, wo sich die entsprechende Seite befindet. Ist diese Seite in der Seitentabelle nicht markiert, so befindet sie sich nicht im realen Hauptspeicher. Ein Interrupt unterbricht das laufende Programm und lädt diese Seite in den realen Hauptspeicher. Um diesen Vorgang zu beschleunigen, werden bereits neue Seiten (bevor sie gebraucht werden) in den Hauptspeicher geladen. (Das Betriebssystem muss demnach aus verwendeten Seiten der Vergangenheit abschätzen welche Seiten in der Zukunft gebraucht werden. Diese Seiten werden eingelagert, falls sich freie oder ersetzbare Seiten im realen Hauptspeicher befinden.
- **Ersetzungsstrategie:** Da die Kapazität des realen Hauptspeichers begrenzt ist, müssen laufend nicht benötigte Seitenblöcke freizusetzen, um benötigt Seiten vom virtuellen Speicher übernehmen zu können.

Dabei sind zwei Fälle zu unterscheiden:

1. Seiten, die seit ihrer letzten Einlagerung nicht geändert wurden können direkt überschrieben werden.
2. Seiten die geändert wurden müssen vorher ausgelagert werden.

Die beste Strategie wäre, dass diejenige Seite ausgelagert wird, die erst am spätesten gebraucht wird. Diese Strategie lässt sich aber nicht implementieren.

Praktische Ersetzungsstrategien sind:

- **FIFO (First-In-First-Out):** Es wird die Seite ersetzt, die am längsten im realen Speicher ist. Somit muss der Zeitpunkt beim Einlagern gespeichert werden. Nachteil: Die Häufigkeit der Verwendung einer Seite wird nicht berücksichtigt.
- **LRU (Least-Recently-Used):** Es wird die Seite ersetzt, die am längsten nicht verwendet wurde. Nachteil: Speicherung des letzten Zugriffszeitpunktes sehr schwierig.

- LFU (Least-Frequently-Used): Es wird die Seite ersetzt, die am wenigsten oft verwendet wurde. Zähler wird bei jedem Zugriff inkrementiert. Nachteil: gerade erst eingelagerte Seite kann gleich wieder ersetzt werden (falsch laut LRU)

Zusätzliche Beschleunigung des virtuellen Speichersystems durch Speicherung von häufig verwendeten Teilen der Seitentabelle in sehr schnellen Registern (Assoziativregister).

---

---

#### FRAGE 4.20: Vorteile der virtuellen Speicherverwaltung.

##### Antwort:

- Einsetzbare Speicherkapazität beträgt ein Vielfaches der realen Hauptspeicherkapazität. (bei 32 Adressbit => maximal 4Gbyte)
  - Programmierer braucht die technischen Speichergrenzen kaum zu beachten. Er arbeitet mit standardisierten Speichereinheiten (Seiten)
  - Der Peripheriespeicher wird wirtschaftlich genutzt.
- 

#### FRAGE 4.21: Nennen Sie Unterschiede zwischen Einprozess- und Mehrprozesssystemen.

##### Antwort:

- Einprozesssystem:  
Kann immer nur einen Ablauf(Prozess) im System unterstützen.  
Es gibt daher keine Konkurrenz um Betriebsmittel und keine Probleme mit Zugriffsschutz. Solche Systeme sind nur für einen Benutzer bedingt geeignet. Dieser kann dann z.B.: nicht gleichzeitig einen Text schreiben und etwas ausdrucken lassen. Der Aufbau des BS ist aber viel einfacher, da Prozessverwaltung, Interprozesskommunikation, Einrichtungen zur Vermeidung von deadlocks usw. entfallen
- Mehrprozesssystem: Ein Mehrbenutzersystem ist stets ein Mehrprozesssystem, da mehrere Programme parallel bearbeitet werden müssen. Beim Einbenutzersystem setzt man sich zum Ziel, das Betriebssystem durch diese Einschränkung wesentlich zu vereinfachen. Es zeigt sich aber, dass sich nur wenige Bereiche für die Vereinfachung des Betriebssystems auf professionellen Arbeitsplätzen durch den Übergang auf ein Einbenutzerbetriebssystem eignen.  
Ein professioneller Anwender erwartet sich:
  - Computersystem muss in ein Computernetz integriert werden
  - Benutzer möchte während laufender Vorgänge eine andere Arbeit durchführen.
  - Benutzer arbeitet an mehreren Aufgaben parallel.

All diese Tätigkeiten erfordern ein Mehrprozesssystem.  
Die Vereinfachungen könnten sein: Verzicht auf Zugriffsschutz in dem Computersystem, verzichten auf Strategien zur Festlegung der Bearbeitungsreihenfolge der Aufträge und Zuteilung der Betriebsmittel. Es zeigt sich aber, dass diese Vereinfachungen Probleme mit sich bringen und deshalb nicht immer sinnvoll sind.  
Es ergeben sich deshalb nur geringe Vereinfachungen vom Mehrbenutzer zum Einbenutzer-BS. Es ist daher für viele Anwendungen nicht falsch, einem einzelnen Benutzer ein Mehrbenutzerbetriebssystem zur Verfügung zu stellen.

---

---

## FRAGE 4.22: Was sind stapelverarbeitende Systeme?

### Antwort:

Stapelverarbeitung war die nahezu ausschließliche Betriebsform für universelle Computer in den 60er Jahren. Heute ist diese Betriebsform bei Hochleistungscomputern die wichtigste.  
Bei der Stapelverarbeitung wird eine Folge von Stapelaufträgen sequentiell oder parallel bearbeitet. Ein Stapelauftrag ist ein vollständig definierter Auftrag eines Benutzers mit allen Steueranweisungen, Programmen und Daten. Er kann ohne Eingriffe des Benutzers bearbeitet werden.  
Der Ausdruck der Stapelverarbeitung stammt aus der Zeit der Lochkarten.( Programme, Daten und Steueranweisungen waren auf Lochstreifen vorhanden und der Auftrag bildete einen Stapel von Lochstreifen)  
Ein einfaches bandorientiertes Stapelverarbeitungssystem rechnet eine Folge von Stapelaufträgen sequentiell gemäß der Reihenfolge in der Eingabewarteschlange. In der Bearbeitung kann also keine andere Auftragsbearbeitung eingeschoben werden. Die Aufträge werden in der Reihenfolge des Eintreffens bearbeitet (FIFO). Auf das Eintreffen neuer Aufträge kann nur nach Ende eines Auftrags reagiert werden. Man nennt diese Strategie der Auftragsbearbeitung eine nichtpräemptive Strategie. Ein weiterer Nachteil ist die schlechte Ausnutzung der Maschine (große Wartezeiten zwischen den Prozessen wegen der Datenübertragung zwischen Hauptspeicher und langsamen Geräten.(z.B.: Drucker, Ausgabemagnetband, ...)  
Eine andere Möglichkeit ist die Verwendung von Stabelverarbeitungssystemen mit Mehrprogrammbetrieb. Diese Systeme verarbeiten im Zeitmultiplexbetrieb mehrere Aufträge parallel. Damit lässt sich eine bessere Reaktionszeit für kurze oder wichtige Aufträge erreichen.

---

## FRAGE 4.23: Was ist Spooling?

### Antwort:

Durch Spooling (Simultaneous Pheripheral Operation On-Line) kann der Durchsatz eines Stapelverarbeitenden Systems durch parallele Ausführung folgender Tätigkeiten erhöht werden:

- Eingabe-Spool-Prozess sorgt für die Aufbereitung der Daten für den Bearbeitungsprozess (Eingabeaufträge werden über Eingabegerät gelesen und in Eingabedatei geschrieben)
  - Ausgabe-Spool\_Prozess gibt Inhalt der Ausgabedatei auf Magnetband oder Drucker aus.
  - Im Bearbeitungsprozess werden die Stapelaufträge ausgeführt. Dieser Prozess greift auf die Virtualisierung der Eingabe und Ausgabegeräte über die anderen beiden Spool-Prozesse zurück. Er muss sich deshalb nicht mehr über die genaue Steuerung des Druckers kümmern, sondern schreibt die Ergebnisse einfach in die Ausgabedatei, die dann vom Ausgabe-Spool-Prozess übernommen wird.
-

---

**FRAGE 4.24: Erklären Sie den Begriff der Stapelfernverarbeitung.****Antwort:**

Hier können die Eingabestapel über ein Netz auf die Eingabedatei überspielt werden. Entsprechend können die Ergebnisse über einen abgesetzten Drucker ausgegeben werden.

Das Betriebssystem benötigt deshalb Erweiterungen wie: Kommunikationsprotokolle, Netzbetrieb, Gruppierung von Geräten zu logischen Stationen usw.

---

**FRAGE 4.25: Was sind Dialogsysteme (Time-Sharing-Systems)?****Antwort:**

Dialogsysteme erlauben eine interaktive Arbeit mehrerer Benutzer über direkt am Computer angeschlossener Endgeräte. (Entwicklung Ende der 70er Jahre)  
Dialogsysteme gehören zur Klasse der Mehrprozess- und Mehrbenutzersysteme. Im Dialogbetrieb bedient der Computer die Benutzer quasi gleichzeitig. Da es Betriebsmittel gibt, die nicht für alle Benutzer ausreichen (z.B.: Prozessor und Hauptspeicher), muss es entsprechende Zuteilungsstrategien geben. Der Hintergrundspeicher (Plattenspeicher) mit den Dateien der Benutzer ist ein nichtentziehbares Betriebsmittel. Man hat daher bei der Zuteilung von Speicherbereichen auf dem Plattenspeicher auf Systemverklemmungen (deadlocks) zu achten. Zur Vereinfachung dieses Problems gibt es folgende Ansätze:

- Für jeden Benutzerauftrag ist die max. Speicherkapazität am Hintergrundspeicher bekannt.
- Zahl der zugelassenen Benutzer ist so begrenzt, dass die Summe ihrer Maximalanforderungen das System nicht überfordert.

Eine sogenannte „Interaktion“ beginnt mit dem Ende der Eingabe des Benutzers und endet mit der Ausgabe des Computers aufgrund dieser Eingabe. Je nach Aufgabe benötigen die Interaktionen verschieden viel Zeit. Diese Zeit ist für das Betriebssystem nicht vorhersehbar. Es tritt deshalb ein Problem auf, da sich der Benutzer eine schnelle Ausführung kurzer Interaktionen wünscht und das Betriebssystem nicht erkennen kann ob es sich um eine kurze Interaktion handelt, die es zwischen einer langen Interaktion einschieben kann. Es muss also „auf Verdacht“ mit der Bearbeitung beginnen.

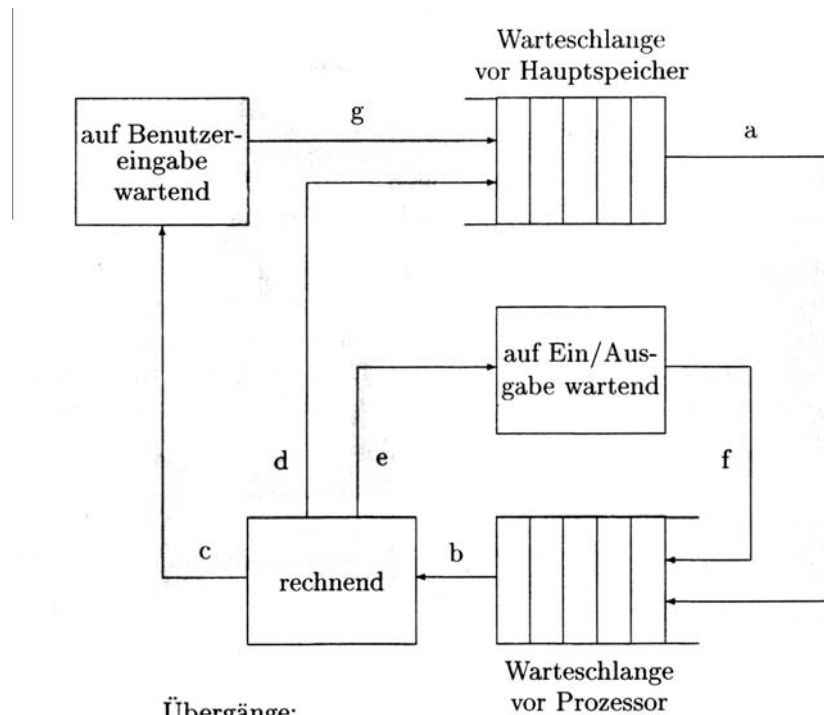
---

**FRAGE 4.26: Was ist Round Robin?****Antwort:**

Das vorher besprochene Problem der Erreichung einer guten Reaktionszeit bei der Zuteilung der Betriebsmittel (Prozessor, Hauptspeicher) an durch Interaktionen hervorgerufenen Prozesse kann durch die RR (Robin Round) – Strategie sehr gut erreicht werden. Diese einfache Prozesszuteilungsstrategie funktioniert folgendermaßen:

Die aktiven Prozesse (sind diejenigen, die eine Interaktion ausführen) werden in eine Warteschlange vor dem Prozessor eingereiht. Jeder Prozess erhält für eine gewisse Anzahl von Zeiteinheiten  $Q$  dieses Betriebsmittel. Wird er innerhalb dieser Zeit nicht beendet, so wird er wieder in die Warteschlange eingereiht. Für sehr kurze Zeitintervalle  $Q$  ergibt sich eine gute Reaktionszeit.

Eine implementierbare RR-Strategie funktioniert folgendermaßen:



Übergänge:

- a Zuteilung des Hauptspeichers
- b Zuteilung des Prozessors
- c Benutzereingabe angefordert
- d Zeitscheibe Q abgelaufen
- e Ein/Ausgabe wird veranlasst
- f Ein/Ausgabe ist beendet
- g Benutzereingabe ist erfolgt

Da nicht alle Prozesse im Hauptspeicher Platz haben, werden jene, die auf Eingabe des Benutzers warten auf den Hintergrundspeicher verdrängt. Sie werden über eine Warteschlange in den Hauptspeicher geladen und kommen erneut in eine Warteschlange vor dem Prozessor. Bei diesem Übergang wird die Rechenzeit  $T$  gespeichert. Der Grad des Mehrprogrammbetriebs bezeichnet die maximale Anzahl der im Hauptspeicher befindlichen Prozesse.

Wenn ein Prozess nun im Prozessor bearbeitet wird, können drei Fälle auftreten:

- Die aufgenommene Zeitscheibe  $T$  ist größer gleich der maximalen Zeitscheibe  $Q$ . Der Prozess wird wieder auf den Hintergrundspeicher verdrängt. Er wird wieder in die Warteschlange vor dem Hauptspeicher eingereiht. (Pfeil d)
- Der Prozess verlangt eine Eingabe vom Benutzer=> wird auf Speicher verdrängt und bleibt solange inaktiv, bis die Eingabe des Benutzers beendet ist. Danach kommt er in die Warteschlange vor dem Hauptspeicher. (Pfeil e)
- Prozess veranlasst einen Ein/Ausgabe-Vorgang. Er wartet dann auf das Ende des Ein/Ausgabe-Transportes. Anschließend wird er wieder in die Warteschlange vor dem Prozessor eingereiht.

Festlegung von  $Q$  nicht einfach: Wird  $Q$  zu groß gewählt, so ergibt sich eine schlechte Reaktionszeit für kurze Interaktionen. Ist  $Q$  zu klein, so wird das System unnötig mit Verdrängen und Laden der Prozesse belastet.

---

---

### FRAGE 4.27: Was ist eine Transaktion?

#### Antwort:

Transaktionen sind in der Regel kurze, fest definierte Interaktionen.  
Z.B.: Flugreservierungen

---

### FRAGE 4.28: Was ist Teilhaberbetrieb?

#### Antwort:

Der Teilhaberbetrieb ist eine spezielle Form des Dialogbetriebs (normal: Teilnehmerbetrieb) bei dem keine freie Programmierung möglich ist. Alle angeschlossenen Benutzer führen nur fest definierte und in der Regel kurze Interaktionen (Transaktionen) durch.  
Alle Transaktionen greifen auf den gleichen globalen Datenbestand zurück. Die Transaktionen der einzelnen Terminals werden von dem Anwenderprogramm in einer festen Reihenfolge nacheinander ausgeführt.  
Da keine Kritischen Betriebsmittelforderungen entstehen, könnte ein Betriebssystem für Teilhaberbetrieb sehr einfach aufgebaut sein. Es ist aber üblich, kommerziell verfügbare Betriebssysteme für universelle Computer zu verwenden.

---

### FRAGE 4.29: Was sind Mehrprozessorsysteme?

#### Antwort:

Bestehen aus zwei bis sechzehn Prozessoren, die auf einen gemeinsamen Hauptspeicher zugreifen. Solche Systeme gehören zur Gruppe der „eng gekoppelten“ Systeme, da die Prozessoren räumlich nicht wesentlich getrennt sind. Es können nun Prozesse echt parallel verarbeitet werden. Es erhöht sich aber nur der Durchsatz der Anlage, nicht aber die Bearbeitung eines einzelnen Auftrags. (Ausnahme: Aufteilung der Aufträge in parallel bearbeitbare Prozesse)

---

### FRAGE 4.30: Warum nimmt der Leistungszuwachs bei Mehrprozessorensystemen schnell ab?

#### Antwort:

Je mehr Prozessoren man bereitstellt, desto schwächer wird der Leistungszuwachs pro hinzugefügtem Prozessor.

Dies hat folgende Gründe:

- Durch Erhöhung der Parallelarbeit => mehr Ein/Ausgabe-Wünsche treten je Zeiteinheit auf => Ein/Ausgabeleistung des Systems wird zu Engpass.

- Da alle Prozessoren auf einen gemeinsamen Hauptspeicher zugreifen, entstehen Wartezustände der Prozessoren.
- Da der Betriebssystemkern ein exklusives Betriebsmittel ist, kommt es auch hier zu Wartezuständen.

Es ergibt sich durch diese Behinderungen z.B. beim Übergang von einem Einprozessorsystem auf ein Zweiprocessorsystem nur eine Leistungssteigerung um den Faktor 1,8!

Es ergeben sich also bei der Realisierung eines Mehrprozessorsystems die zwei wichtigsten Ziele: möglichst hohe Parallelität bei möglichst geringer gegenseitiger Behinderung der Prozesse und Prozessoren.

---

### FRAGE 4.31: Was sind verteilte Computersysteme?

#### Antwort:

Ein verteiltes Computersystem besteht aus einem Netz von eigenständigen Computersystemen. Das Betriebssystem koordiniert die Zusammenarbeit der einzelnen Systeme und die Abwicklung der Aufträge auf diesen. Diese Computersysteme sind nicht über einen gemeinsamen Hauptspeicher, sondern über z.B. ein lokales Netz miteinander verknüpft.

Man nennt solche Systeme auch „lose gekoppelte Systeme“.

---

### FRAGE 4.32: Was ist die Zielsetzung von lose gekoppelten Systemen?

#### Antwort:

- Erhöhung der Rechenleistung
- Leichte Erweiterbarkeit bei Bedarf nach neuen Funktionen oder mehr Rechenleistung durch hinzufügen von Computersystemen (ev. mit neuen Dienstleistungen)
- Leichte Installation und Wartung (durch gleiche Systeme und modularem Aufbau)
- Bessere Verfügbarkeit (Ausfallsicherheit durch mehrere Computer sehr hoch)
- Höhere Sicherheit gegen Verlust der Benutzerdaten.
- Verwendung gemeinsamer Ressourcen. (z.B.: teure Spezialgeräte)

Verteilte Systeme erscheinen im Idealfall für den Benutzer unstrukturiert. Das System kann die Objekte ohne Kenntnis des Benutzers beliebig verschieben. Diese Objekte werden unabhängig von ihrem Speicherort benannt.

Die Aufträge werden durch geeignete Lastverteilungsalgorithmen auf die einzelnen Computer aufgeteilt. Es ergibt sich eine kleinere gegenseitige Behinderung der Rechner im Vergleich zum Mehrprozessorsystem, da es wenig gemeinsame Betriebsmittel gibt (z.B.: kein gemeinsamer Hauptspeicher). Flaschenhals ist aber die Kommunikation.

Es gibt auch die Möglichkeit einer redundanten Schaltung sämtlicher Komponenten eines Systems => im Idealfall keine Leistungseinbuße bzw. kein

Info-Verlust bei Ausfall eines Rechners. Optimal ist eine Verwendung eines gemeinsamen Betriebssystems mit einheitlichem Betriebssystemkern.

In verteilten Computersystemen gibt es folgende besondere Kennzeichen:

- Durch langsame Kommunikation =>kein Rechner weiß über den aktuellen Zustand des Systems bescheid => trotzdem muss zuverlässige und produktive Arbeit möglich sein.
  - Da Datenbestände mehrfach vorhanden sind =>nach Ausfall einer Komponente => Datenkonsistenz muss gewährleistet sein.
  - Verwendung eines Netzstandards und Standardprotokolle unbedingt erforderlich (zukünftige Anbindung an andere Netze)
  - Bei auftreten eines Fehlers vor Abschluss der Aufgabe müssen alle Veränderungen rückgängig gemacht werden.
- 

### FRAGE 4.33: Was sind Echtzeitsysteme?

Antwort:

Werden zur Überwachung und Steuerung technischer Prozesse und für die Erfassung von Messdaten eingesetzt. Wichtig ist deshalb die garantierte Einhaltung gewisser Zeitbedingungen. So muss z.B. in gewissen Zeitabständen ein Signal abgetastet werden oder Messdaten abgefragt werden. Oder es müssen bestimmte Tätigkeiten zu festen Zeitpunkten angestoßen werden.

Einige Eigenschaften von Echtzeitsystemen:

- Ausgefeilte Zeitverwaltung.
  - Alle Abläufe u. Wartezustände in dem Betriebssystem und den Anwendungsprogrammen werden zeitlich überwacht.
  - Man ist nicht an Virtualisierung interessiert.
  - Zeitkritische Programme sind Hauptspeicherresident und werden nicht verdrängt.
  - Hauptspeicher wird aus Zeitgründen direkt adressiert.
- 

### FRAGE 4.34: Unterschied Prozessrechner und Universalrechner?

Antwort:

Die Hardware eines Echtzeitsystems wird Prozessrechner genannt.. Sie unterscheidet sich von der Hardware eines Universalrechners durch Vorhandensein folgender Komponenten:

- Echtzeit Uhren
  - Analog-Ein/Ausgänge
  - Digital-Ein/Ausgänge
  - Vielfalt von angeschl. Geräten (Messwertgeber, Messwertempfänger, Sensoren,...)
  - Große Anzahl von Interrupt-Ebenen.
- 

### FRAGE 4.35: Was sind Interruptebenen?

**Antwort:**

Hierarchische Struktur von durch außen aufrufbare (z.B. durch Signale der Sensoren) Unterbrechungsrountinen, die ein Stoppen der aktuellen Befehlsausführung zur Folge haben und die Bearbeitung eines zugeordneten Unterprogramms einleiten. In dieser sogenannten Interruptroutine erfolgt dann die Behandlung dieses Zustandes.

---

---

### FRAGE 4.36: Was ist ein Zielprogramm?

#### Antwort:

Da der Computer nur Maschinencode versteht, der Programmierer aber in einer symbolischen Assemblersprache oder in einer höheren Programmiersprache komfortabel arbeiten will, sind sogenannte Compiler erforderlich. Ein Compiler übersetzt das Quellprogramm in ein für den Computer verständliches Zielprogramm.

---

### FRAGE 4.37: Welche Arten von binären Zielprogrammen gibt es?

#### Antwort:

Man unterscheidet drei Arten von Zielprogrammen:

- absolut-binäre Programme:  
Sie sind bereits mit den absoluten Hauptspeicheradressen versehen und können unmittelbar ausgeführt werden. Der sogenannte Lader(loader) transportiert das Programm aus dem Peripheriespeicher in den Hauptspeicher, wo es dann ausgeführt werden kann. Das Laden von Programmen ist eine wichtige Aufgabe des Betriebssystems.
  - verschiebbar-binäre Programme:  
Sind relativ zum Programmumfang adressiert. Die relativen Adressen müssen vor der Programmausführung durch einen sog. (Verschiebe-)Lader in die absoluten Adressen übergeführt werden. Dies ist wichtig für den Mehrprogrammbetrieb, wo die Programme im Hauptspeicher verschiebbar bleiben müssen.
  - symbolisch-binäre Programme:  
Daranter versteht man Programme, die zwar bereits in binärer Form vorliegen, aber noch durch symbolische Variablen Bezug auf andere symbolisch binäre Programme haben (z.B.: Unterprogrammtechnik). Der sogenannte Bindelader übernimmt die Übersetzung in ein verschiebbar-binäres Programm und die Einbringung des Programms in den Hauptspeicher.
- 

### FRAGE 4.38: Was sind Dienstprogramme des Betriebssystems?

#### Antwort:

Ein Betriebssystem verfügt außer über den Betriebssystemkern noch über ein Programmiersystem. Dieses System besteht aus Compilern und einer Reihe von Dienstprogrammen, die dem Anwender den Umgang mit dem Computer und die Nutzung der Betriebsmittel erleichtern.

Man versteht unter den Dienstprogrammen diverse Arbeitsprogramme des Betriebssystems, die beim normalen Computereinsatz relativ häufig gebraucht werden.

Man kann sie in folgende Gruppen einteilen:

- Kopierprogramme: kopieren Daten zw. Peripheriespeichern
- Mischprogramme: für Zusammenführung mehrerer Dateien zu einer Neuen (Editor)
- Sortierprogramme: stellen Ordnung der Datensätze einer Datei her
- Ausgabeprogramme: erleichtern die Ausgabe (z.B. auf Drucker)
- Testprogramme: sind Diagnosehilfe beim Testen von Programmen (Debugger)
- Hauptspeicherbezogene Dienstprogramme: werden benötigt, um bestimmte Speicherinhalte meist zum Zwecke der Fehlersuche zu setzen oder zu lesen.

---

FRAGE 4.39: Beschreiben Sie einige Betriebssysteme.

Antwort:

### MS/DOS:

Wurde von der Firma MicroSoft speziell für 16-Bit Personal-Computer, die auf der Intel 80x86-Familie basieren, entwickelt. Die Hauptspeicherverwaltung beschränkt sich auf 640Kbyte. DOS ist ein Einbenutzer-/Einprozesssystem mit Spooling. Die erste Version wurde 1981 mit dem IBM Personal-Computer ausgeliefert. Die Version DOS 2.0 folgte 1983. Sie unterstützte 10 Mbyte Plattenspeicher. Ab dieser Version wurde eine hierarchische Dateistruktur eingeführt. Mehrbenutzer und Mehrprozessfähigkeit sowie geschützte Speicherzugriffe usw. wurden nie erreicht.

### OS/2, WINDOWS:

Wurden von MicroSoft entwickelt und sind speziell auf die Prozessorfamilie 80x86 zugeschnittene Einbenutzer/Mehrprozess- Betriebssysteme. Mehrere Anwendungen können parallel laufen, ohne einander zu stören (Protected Mode). Die Grenze der Hauptspeicherverwaltung von 640Kbyte wurde überwunden. Diese Betriebssysteme können 1Gbyte virtuellen Speicher adressieren. Über die Kompatibilitätsbox laufen herkömmliche Anwendungen auf dem Betriebssystem MS/DOS.

### UNIX:

Unix ist das einzige Betriebssystem, das sowohl auf Kleincomputern als auch auf Großcomputern läuft. Früher wurde es fast ausschließlich im universitären Bereich verwendet. Heute hat es bereits eine weite Verbreitung. Unix wurde in den Bell-Laboratorien gegen Ende der 60er Jahre geboren.

Nun folgt ein kurzer Überblick über die Entwicklung von UNIX:

Aufbauend auf dem Dialogbetriebssystem MULTICS wurde das Betriebssystem UNICS entwickelt. Die UNIX Version 5 entstand aus einer vollständigen Neuimplementierung in der Sprache C. UNIX war damals das erste "portable" Betriebssystem. Es wurden äußerst kostengünstige Kopien an akademische Einrichtungen weitergegeben, sodass sich UNIX rasant weiterentwickelte. Es entwickelten sich viele Derivate, die nicht UNIX genannt werden durften (AEGIS, XENIX, MINIX, PRIMIX,...) Die Version UNIX System V.4 versuchte unterschiedliche Entwicklungslinien wieder zusammenzuführen.

Ein besonderes Anliegen vieler UNIX-orientierter Hersteller ist es, die Ausfallsicherheit zu erhöhen. Dies erreicht man durch Bildung von sog. Cluster. (mehrere Computer, die von Außen als einziger Computer wirken)  
Es gibt sehr viele unterschiedliche UNIX-Varianten, die sich generell in drei Hauptentwicklungslinien einteilen lassen:

- AT&T-Linie (aktuelle Version UNIX V.3)
- Berkeley-Linie (aktuelle Version BSD 4.4)
- XENIX-Linie (aktuelle Version XENIX V)

In letzter Zeit laufen große Anstrengungen zur Standardisierung und Vereinheitlichung von UNIX-Betriebssystemen. Drei Dokumente bilden Grundlage für Versionszusammenführung und Standardisierung von UNIX-Systemen:

- Die SVID (System V Interface Definition)
- Der X/OPEN Portability Guide
- Die IEEE-Normentwicklung POSIX

## LINUX:

Ist eine freie Version von UNIX. LINUX wurde 1991 an der Universität Helsinki entwickelt. Der LINUX-Kernel wurde zum freien Download angeboten. Eine Vielzahl freiwilliger Entwickler arbeiten seither an LINUX. Mittlerweile bildet der LINUX-Kernel zusammen mit den GNU-Programmen der Free Software Foundation ein stabiles Betriebssystem, das vor allem in der Software-Entwicklung zum Einsatz kommt.

Leistungsmerkmale: Multiprozessorfähigkeit, Dateisysteme mit Journal, erweiterte Dateizugriffsrechte,...). LINUX ist mittlerweile den UNIX-Systemen an Funktionalität nicht mehr unterlegen. Durch die graphischen Benutzeroberflächen (KDE, Gnome) und immer mehr Anwendungssoftware schafft LINUX immer mehr den Einzug in den Office-Bereich.

LINUX ist das am meisten portierteste Betriebssystem und wird auch in sogenannten Embedded Systems verwendet.

## VMS:

Ist ein Betriebssystem der Firma „Digital Equipment Corporation“, das im Jahr 1978 vermarktet wurde. Es zeichnet sich aus durch eine besondere Robustheit und gute Cluster-Fähigkeit. Im Bezug auf Geschwindigkeit (z.B. Dateizugriffe) ist es aber UNIX unterlegen.

VMS wird deshalb vor allem dort verwendet, wo es auf hohe Datensicherheit ankommt (z.B.: Banken, Spitäler, usw.). Die aktuelle Version heißt OpenVMS und ist nur auf VAX und Alpha-Prozessoren lauffähig.

## MVS:

MVS (Multiple Virtual Storage) ist neben VM das wichtigste Betriebssystem für Großcomputer der Firma IBM. Etwa die Hälfte aller Großcomputer wird von IBM gefertigt. MVS ist speziell für stapelverarbeitende Computersysteme geeignet und wurde im Jahr 1974 entwickelt. Da der Bedarf an stapelverarbeitenden Computersystemen höchster Leistungen immer gegeben sein wird, scheint die Zukunft von MVS auch langfristig gesichert.

**VM:**

Wurde von IBM 1972 entwickelt. Die grundlegende Idee von VM besteht darin, über das sogenannte CP (Control Program) jedem Benutzer eine virtual machine vorzugeben, welche wie ein Computersystem mit nur einem Benutzer bedient werden kann. Jede dieser Maschinen kann ein eigenes (Unter-)Betriebssystem verwenden. Neben CMS (Conversational Monitor System) kann als Unterbetriebssystem auch LINUX, MVS sowie der UNIX-Dialekt AIX verwendet werden.

**BS2000:** Ist das Betriebssystem der SIEMENS-Großcomputer mit Skalarprozessor. Wird primär im kaufmännischen Bereich verwendet, wo der Benutzer im Teilhaberbetrieb Zugriff auf die Rechenanlage hat.

---

## KAPITEL 5 - Datenbanksysteme -

---

**FRAGE 5.1: Was sind die Nachteile einer starren Zuordnung der Daten zu den sie verarbeitenden Programmen?**

**Antwort:**

In den Anfängen der elektronischen Datenverarbeitung wurden Rechner fast ausschließlich zur Verarbeitung numerischer Daten verwendet. Später trat nach und nach ihr Einsatz zur Verarbeitung nicht-numerischer Daten in den Vordergrund. Der Computer wurde auch zum Verwalter großer Datenbestände, deren Speicherbedarf weit über die Kapazität von Arbeitsspeichern hinausgeht.

Mit der Entwicklung von Anwenderprogrammen zur Aktualisierung und Auswertung der Dateien, dem Übergang vom Stapel- zum Dialogbetrieb und der Ermöglichung des Mehrfachzugriffes auf Daten entstanden die sogenannten Dateisysteme.

Bei diesen Dateisystemen besteht eine starre Zuordnung von Anwendungen zu Dateien.

Das bedeutet, dass jedes Anwendungsprogramm eine eigene Datei verwaltet. Somit existieren mehrere Dateien mit dem gleichen Inhalt.

Es ergeben sich dadurch folgende Probleme:

- hohe Redundanz zwischen einzelnen Dateien durch Mehrfachspeicherung. Diese Redundanz wird zentral nicht kontrolliert.
  - Gefahr der Inkonsistenz: Entsteht, wenn ein Programm seine Datei ändert und die anderen Programme diese Änderung nicht in ihren Dateien durchführen.
  - Inflexibilität gegenüber Veränderungen. Ändert sich die Anwendung, so ist ein neues Programm mit einer neuen Datei zu erstellen. Es ergibt sich dadurch auch eine aufwendige Programmwartung (z.B.: wenn sich Dateiformat,... ändert =>Anwendungsprogramme müssen auch verändert werden.)
  - Problem der Einhaltung von Standards, da z.B. Dateiformat durch Anwendungsprogramme festgelegt wird. => Probleme beim Übergang auf neues Betriebssystem oder beim Datenaustausch zwischen unterschiedlichen Datenbanksysteme.
- 

**FRAGE 5.1: Was sind die Aufgaben eines Datenbankverwaltungssystems?**

**Antwort:**

Die in der vorhergehenden Frage besprochenen Probleme legen eine Lösung nahe, nach der die Daten als eigentliche, wesentliche Betriebsmittel betrachtet werden. Alle Anwendungsdaten arbeiten mit einem integrierten Datenbestand. Man spricht von

einem Datenbanksystem. Die Software des Datenbanksystems wird auch als DBMS bezeichnet (Data Base Management System/ Datenbankverwaltungssystem). Sie sorgt dafür, dass die Daten den Anwendungsprogrammen in geeigneter Form zur Verfügung gestellt werden.

Es wird durch das DBMS eine Trennung von Programmen und Daten erreicht. Es übergibt den Anwendungsprogrammen nicht mehr ganze Datensätze, sondern genau die vom Programm benötigten Datenelemente => Datensatzstruktur bzw. interne Datenorganisation sind für Programm unsichtbar.

Durch das DBMS ergeben sich folgende Vorteile:

- Gemeinsame Basis für Anwendungen (Vereinheitlichung)
- Redundanz entfällt; wo sie nützlich ist (Effizienz) wird sich durch DBMS kontrolliert.
- Es entfallen die Konsistenzprobleme der Dateisysteme
- Anwendungsprogrammierung wird einfacher (Programmierer muss den internen Aufbau der Datei nicht berücksichtigen).
- Änderungen im Dateiaufbau führen nicht mehr zwingend zu einer Änderung der Anwendungsprogramme.
- Datenbanksystem kann die Plausibilität von Daten prüfen
- DBS kann zentrale Mechanismen zur Wiederherstellung im Fehlerfall bereitstellen.

---

**FRAGE 5.2: Erklären Sie die Begriffe von physischer und logischer Datenunabhängigkeit.**

**Antwort:**

Unter physischer Datenunabhängigkeit versteht man die Trennung der Daten von den Anwendungsprogrammen durch das Datenbankverwaltungssystem (DBMS). Man versteht darunter die Unabhängigkeit des Programms von der Organisation der Daten. Logische Datenunabhängigkeit bedeutet, dass zwischen einer spezifischen, anwendungsbezogenen Sicht auf die Datenbank und deren logischer Gesamtstruktur unterschieden werden kann. (Programm sieht nicht logische Gesamtstruktur sondern nur einen Teil davon). Es soll gewährleistet sein, dass bei Änderung der globalen Struktur des Datenbanksystems nicht die Anwendungsprogramme geändert werden müssen.

---

**FRAGE 5.3: Was sind die Vorteile eines Datenbanksystemkonzeptes?**

**Antwort:**

- Vermeidung von Redundanz.
- Keine Dateninkonsistenz
- Mehr Flexibilität in der Datenauswertung.
- Gewährleistung der Integrität der Daten: Dies bedeutet Korrektheit und Vollständigkeit. Bei Dateisystemen ist der Anwendungsprogrammierer dafür verantwortlich, geeignete Sicherheitsmechanismen in das Anwendungsprogramm zu implementieren.=> je nach Programm unterschiedl.

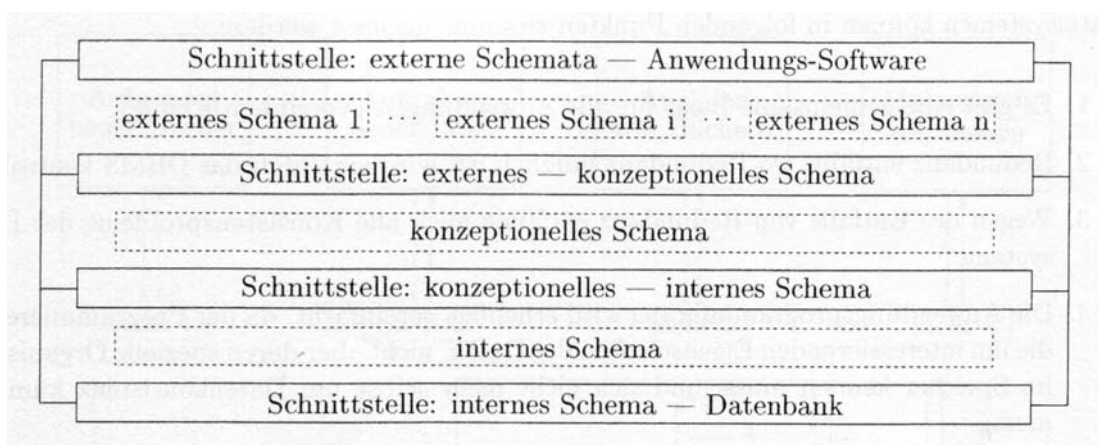
Sicherheitsvorkehrungen möglich=> besser sind für alle Daten vereinheitlichte Maßnahmen des DBMS.

- ... (siehe FRAGE 5.1)
-

## FRAGE 5.4: Erklären sie das 3-Ebenen-Architekturkonzept für Datenbanksysteme.

### Antwort:

Ist ein etabliertes Architekturkonzept für Datenbanksysteme, welches aus drei Schichten besteht und jeder Schicht ein eigenes Schema zuordnet:



- Die interne Schicht liegt dem physikalischen Speicher am nächsten. Sie betrachtet diese Daten aber nicht als Speicherseiten oder Blöcke sondern als Datensätze. Das interne Schema legt also alle Definitionen zu Art und Aufbau der verwendeten Datenstrukturen fest. Diese Schnittstelle hat die Abbildung des logischen auf den physikalischen Adressraum durch geeignete Zugriffsmechanismen zu erledigen.
- Die konzeptionelle Schicht legt die logische Gesamtsicht aller Daten und ihre Beziehungen zueinander im konzeptionellen Schema fest. Diese Abstraktion von der internen Schicht wird durch ein Datenmodell erreicht. Zur Gewährleistung der logischen Datenunabhängigkeit ist dabei wesentlich, dass dieses Schema von Datenstruktur- und Zugriffsaspekten frei ist. Zu dieser Definition steht die Datendefinitionssprache des konkreten Systems zur Verfügung.
- Die externe Sicht (Ausschnitt der Gesamtsicht) beinhaltet alle individuellen Sichten der Anwendungsprogramme. Diese Sichten jeweils in einem eigenen externen Schema beschrieben.

Diese einzelnen Schichten kommunizieren über Schnittstellen, welche vom DBMS zur Verfügung gestellt werden miteinander

## FRAGE 5.5: Was ist eine DML (Data Manipulation Language)?

### Antwort:

Das DBMS stellt auch eine bestimmte Schnittstelle zum Anwender bereit, welche im 3-Schichten-Architekturkonzept zwischen externer Schicht und den Anwendungsprogrammen liegt. Diese Schnittstelle umfasst insbesondere eine Sprache, die sogenannte Datenmanipulationssprache DML über die der Benutzer mit dem DBMS kommunizieren kann. Anwender einer bestehenden Datenbank benötigen diese Sprache um z.B.: Updates an den Daten durchzuführen (ändern, löschen, einfügen,...) oder um Anfrage zu stellen.

Die DML kann folgende Erscheinungsformen haben:

- Sie kann als eigenständige Dialogsprache implementiert sein, sodass der Benutzer direkt im Dialog mit der Datenbank arbeiten kann (z.B.: dBase, SQL)
- Sie ist in eine sog. Wirtssprache (Host Language eingebettet). Die Wirtssprache ist meist eine klassische höhere Programmiersprache (COBOL, Fortran, Java, Pascal, PL/1) die um Sprachelemente erweitert wurde und durch einen speziellen Precompiler in die eigentliche Wirtssprache übersetzt wird.

---

### FRAGE 5.6: Was ist eine DLL (Data Definition Language)?

**Antwort:**

Bevor man mit der Datenbank arbeiten kann, müssen die logischen und physischen Strukturen der Datenbank definiert werden. Hierzu stellt das DBMS eine Datendefinitionssprache (DDL) bereit. Mit dieser Sprache werden alle Einzelheiten des konzeptionellen- und internen Schemas festgelegt. Im Anschluss an diese Festlegungen wird noch die externe Schicht durch die DDL beschrieben.

---

### FRAGE 5.7: Unterschied zwischen DDL und DML?

**Antwort:**

Es gibt zwei Philosophien in Bezug auf die Realisierung durch DDL und DML bei Datenbanksystemen:

- Manchmal wird kein Unterschied zw. DML und DDL gemacht (Befehle können in beliebiger Reihenfolge verwendet werden). Es ist damit möglich, bereits mit der Datenbank zu arbeiten obwohl erst ein Teil der Gesamtkonzeption fertig ist.
- Bei anderen Systemen ist das Arbeiten mit der Datenbank erst nach der vollständigen Festlegung des konzeptionellen Schemas durch die DDL möglich. Nachträgliche Änderungen erfordern dann einen erheblich höheren Aufwand.

---

### FRAGE 5.8: Was ist ein Datenbankadministrator?

**Antwort:**

Die DML kann von allen Anwendern benutzt werden, während die DDL meistens nur dem Datenbankadministrator (DBA) vorbehalten bleibt. Diese Instanz kann aus einer

Gruppe von Personen bestehen. Der DBA ist verantwortlich für die zentrale Kontrolle des Gesamtsystems.

---

### FRAGE 5.9: Was gehört zu den Aufgaben des Datenbankadministrators?

#### Antwort:

Zu den Kompetenzen und Verpflichtungen des DBA kann z.B. gehören:

- Einrichten einer Datenbank
  - Vergeben von Zugriffsrechten
  - Festlegen von Zugriffskontrollen
  - Kontrollieren der Integrität (Vollständigkeit und Korrektheit) der Datenbanken.
  - Festlegen von Sicherungsstrategien
  - Festlegen von Wiederanlaufstrategien.
  - Abstimmen der Konfigurationsparameter des DBMS.
  - Überwachen des Laufzeitverhaltens des Systems.
- 

### FRAGE 5.10: Wie kann die DBA-Institution aufgespalten werden?

#### Antwort:

Kann gemäß dem 3-Ebenen-Architekturkonzept aufgespalten werden:

- In die Anwendungsadministration für die externe Schicht.
  - In die Unternehmensadministration für die konzeptionelle Schicht und
  - In die eigentliche Datenbankadministration für die interne Schicht.
-

## KAPITEL 6 - Verteilte Datenverarbeitung –

---

### FRAGE 6.1: Entwicklung zur verteilten Datenverarbeitung?

#### Antwort:

Früher wurde der Computer als alleinstehendes mathematisches Werkzeug für komplexe Berechnungen verwendet. Heute steht dagegen mehr die Haltung und Manipulation großer Datenbestände im Vordergrund. (z.B.: Lohndaten einer Firma, Kontendaten einer Bank,...) Diese Datenbestände müssen einer großen Zahl von Sachbearbeitern zugänglich sein, die im Dialog darauf zugreifen. Auch Mitarbeiter im Außendienst müssen teilweise auf diese Daten zugreifen können. Bei komplizierten Dialogvorgängen ist eine Benutzerführung bei der Eingabe erwünscht. Problem: Übertragung dieser Daten zu kostspielig und zu langsam. => Verteilung der Rechenleistung auch auf entfernte Computer => Computernetz  
Auch die Kopplung unterschiedlicher Netze wird immer wichtiger.

---

### FRAGE 6.2: Nach welchen Gesichtspunkten werden Rechenleistung und Datenhaltung verteilt?

#### Antwort:

- Aufteilung der Aufgabe auf mehrere Rechner kann dadurch erzwungen werden, dass ein einzelner Computer mit der Aufgabe überfordert wäre.
  - Durch eine Zuordnung der Computer zu den Organisationseinheiten des Unternehmens => klare Abgrenzung der Verantwortlichkeit; keine Überschneidung von Aktivitäten
  - Änderungen im Datenverarbeitungssystem durch modularem Aufbau mit einzelnen, vernetzten Computern und Geräten sehr kostengünstig und einfach zu bewerkstelligen.
  - Speicherung und Verarbeitung der Daten ist billiger als deren Übertragung => Rechenleistung und Datenhaltung sind jeweils „vor Ort“ zu erbringen. Im lokalen Bereich können aber teure Spezialgeräte von mehreren Anwendern gleichzeitig genutzt werden
- 

### FRAGE 6.3: Welche Arten von Verbunden gibt es?

#### Antwort:

- Lastverbund: dient der Erhöhung der Gesamtleistung des Systems durch Abwicklung gleichartiger Aufgaben an mehreren Stellen gleichzeitig. Beim dynamischen Lastverbund werden die Aufgaben je nach Auslastung der Systeme optimal auf diese verteilt.
  - Sicherheitsverbund: Durch eine Vervielfachung des Systems erhält man die Möglichkeit, dass nach Ausfall eines Computers dessen Aufgaben von anderen gleichartigen Computern übernommen werden. (statische oder dynamische Implementierung möglich)
  - Funktionsverbund: Unterschiedliche Aufgaben werden auf spezialisierte Computer verteilt. Beim Programmverbund laufen aufwendige Programme nur in einem Computer ab, während die Ausführung aber von verschiedenen Computern angestoßen werden kann.
  - Datenverbund: Daten werden dort gehalten, wo sie hauptsächlich verwendet werden, um Übertragungskosten zu sparen. Durch Speicherung der gleichen Daten an mehreren Orten erreicht man einen Sicherheitsverbund.
- 

#### FRAGE 6.4: Welche Arbeitsweisen gibt es in der verteilten Datenverarbeitung?

##### Antwort:

- Stapelbetrieb: Wird auch heute noch bei z.B.: komplexen wissenschaftlichen Berechnungen eingesetzt. Dabei wird ein Programm durch Eingabe von Daten gestartet und läuft nach festen Regeln ab, bis ein Ergebnis zurückgeliefert wird. Beim Stapelfernbetrieb werden die Ein/Ausgabedaten über Leitungen zum Supercomputer transportiert.
  - Dialogbetrieb: Beim Dialogbetrieb arbeitet ein Mensch an einem Terminal im Dialog mit dem Computersystem. D.h. er wartet auf Ergebnisse und löst durch Eingaben einen weiteren Schritt aus. In den 70er Jahren wurde Teilnehmerbetrieb zur gemeinsamen Nutzung eines Computers durch mehrere Benutzer eingesetzt. (Rechenleistung noch nicht billiger als Übertragung). Eine sehr wichtige Form des Dialogbetriebs ist der Teilhaberbetrieb. Die Benutzer des Systems haben gleiche Aufgaben und beziehen sich auf einen gemeinsamen Datenbestand. In dieser Betriebsart werden meist Transaktionen (On-Line Transaction Processing OLTP: Vorgang, der als Ganzes abgewickelt werden muss, damit alle betroffenen Daten konsistent bleiben) ausgeführt.
- 

#### FRAGE 6.5: Welche Funktionen/Dienste stellt ein modernes Computernetz zur Verfügung?

##### Antwort:

- Koordinierung von entfernt voneinander ablaufenden Aktivitäten durch Nachrichten. (zwischen Mensch und Programm, zw. zwei Programmen oder

zw. zwei Menschen). Sichere Übertragung dieser Nachrichten muss gewährleistet sein.

- Vermittlungsaufgaben mit Adressierung der Partner im Netz und zugehörige Wegfindung für die Übertragung erforderlich.
- Koordinierungsleistungen beim Transfer von größeren Datenmengen.
- Im Verbundbetrieb ist die Funktion des Jobtransfer notwendig.
- Elektronische Nachrichtensysteme (electronic mail) bei Büroautomatisierung
- Effiziente Archivierung großer Datenmengen
- Umfangreiche Administrationsdienste
- Funktion des Fernprozeduraufrufes (remote procedure call) zur Weitergabe von Teilaufgaben an andere Computer im lokalen Bereich.
- Im Transaktionsbetrieb => Transaktionsmonitor notwendig (sorgt für Konsistenz des gemeinsamen Datenbestandes)

---

## FRAGE 6.6: Was ist die Schichtung?

### Antwort:

Die Datenkommunikation bedingt durch die Vielfältigen zu erbringenden Aufgaben und Funktionen eine sehr komplexe Hard- und Software voraus. Diese Komplexität bekommt man nur dadurch in den Griff, dass man das Gesamtsystem in Teile zerlegt, die untereinander zusammenarbeiten. Die Regeln der Zusammenarbeit eines Moduls mit einem anderen sind mit der Schnittstelle definiert. Das System wird nach Funktionen aufgeteilt wobei die Schnittstelle möglichst einfach gehalten wird.

Ein besonders klares Ordnungsprinzip ist die Schichtung. Hier werden die Funktionen so aufgebaut, dass in jeder Schicht die Funktionen der darunterliegenden Schicht als Dienste in Anspruch genommen werden können und für die darüberliegende Schicht Dienstleistungen erbracht werden.

Diese Modularisierung bringt den Vorteil, dass Veränderungen der Module leicht durchgeführt werden können, ohne das ganze System neu gestalten zu müssen (sofern die Schnittstelle unverändert bleibt). Auch die Entwicklung eines Kommunikationssystems ist wesentlich einfacher, da die Module einzeln entwickelt und getestet werden können.

---

## FRAGE 6.7: Was sind die Nachteile der Modularisierung?

### Antwort:

Da geschichtete Systeme ihre Aufgaben für den allgemeinen Fall erbringen sollen, benötigt der einheitliche Ablauf einen Aufwand, der für Einzelaufgaben unnötig ist. Die Lösung durch Schichtung wird deshalb für kleine Anwendungen nicht so effizient sein wie eine auf den jeweiligen Einsatzfall optimierte Lösung. Dieser Nachteil wird aber in Kauf genommen, da die Vorteile der Modularisierung überwiegen und der Nachteil durch die rapide wachsende Hardwareleistung bei sinkendem Preis vermindert wird.

---

---

## FRAGE 6.8: Erläutern sie das OSI-Referenzmodell

### Antwort:

Das OSI-Referenzmodell für offene Kommunikation wurde von der ISO im Jahre 1984 verabschiedet. Es bildet die Grundlage für Normungen und Festlegungen im Bereich der Kommunikation. Es verwendet das Prinzip der Funktionsschichtung. Alle Funktionen die für die Kommunikation zw. 2 Elementen eines verteilten Systems benötigt werden sind hierarchisch angeordnet.

Das OSI-Referenzmodell definiert die grundlegenden Konzepte des Modells und die Funktionalität der einzelnen Schichten. Die Dienste der Funktionen werden nur allgemein beschrieben. Sie sind in gesonderten Normen zu definieren. Auch die Protokolle (Syntax und Semantik der ausgetauschten Nachrichten) werden im OSI-Referenzmodell nicht definiert. Die Protokolle leiten sich vielmehr von den Diensten ab.

Man spricht von einem „offenem System“, falls es die so definierten Protokolle benutzt.

In den 70er Jahren war es mit großen Schwierigkeiten verbunden, Netze verschiedener Hersteller untereinander mit einem aus der Benutzersicht einheitlichen Dienst zu verbinden.

Bei der Entwicklung des OSI-Modells setzte man sich folgende Ziele:

- Einheitliches Modell der Kommunikation verteilter Systeme erreichen
- Freiheiten der Implementierung des Modells nicht einschränken

Das Ziel war, die offene Kommunikation zwischen allen Teilnehmern an einem heterogenen Netz zu verwirklichen. Systeme, die die nach außen sichtbaren Eigenschaften eines solchen Systems aufwiesen wurden „offene Systeme“ genannt.

Kurze Begriffserläuterungen:

- Endsysteme: Endpunkte der Kommunikationslinien (z.B.:PC)
- Teilnetze: firmeninterne Netze (z.B.: LAN) die über
- Transitnetze: mit anderen Firmen verbunden werden können.(globale Netze)

OSI-Modell muss nicht die beste Lösung sein. (lässt sich nicht beweisen) Je höher die Schicht, desto mächtigere Aufgaben sind durchführbar.

Einteilung in 7 Schichten:

Verarbeitungsschicht	Schicht 7
Darstellungsschicht	Schicht 6
Kommunikationssteuerungsschicht	Schicht 5
Transportschicht	Schicht 4
Vermittlungsschicht	Schicht 3
Sicherungsschicht	Schicht 2
Bitübertragungsschicht	Schicht 1

- Bitübertragungsschicht (physical layer) Ermöglicht den Transport der Bits auf verschiedenen Medien und mit unterschiedlichen Verfahren. Sie verbirgt die physikalischen Eigenschaften der Übertragungsmedien.
- Sicherungsschicht (link layer): Zuständig für die Fehlererkennung und Korrektur.
- Vermittlungsschicht (od. Netzschicht /network layer): Sorgt für die richtige Weiterleitung der Nachrichten zum Zielsystem. Muss auch das Zusammenschalten von Teilnetzen unterschiedlicher Struktur und Technologie bewältigen und einen einheitlichen Vermittlungsdienst anbieten.
- Transportschicht (transport layer): Sorgt für einen optimalen Transport – beste Leistung bei minimalen Kosten – von Nachrichten zwischen Partnern und bietet den Benutzern einen einheitlichen Transportdienst.
- Kommunikationssteuerungsschicht (session layer): Legt die Regeln der Kommunikation fest und überwacht diese.
- Darstellungsschicht (presentation layer): Enthält Funktionen für die Darstellung und die Manipulation strukturierter Daten.
- Verarbeitungsschicht (Anwendungsschicht /application layer): Legt die erforderlichen Protokolle für eine Kommunikation zwischen Anwendungsprozessen fest und stellt die notwendigen Dienste bereit.

---

## FRAGE 6.9: Was ist das Internet?

### Antwort:

Internet steht für INTERnational NETworking. Früher konnte man die Zusammenarbeit zweier Computer über Relais bewerkstelligen. (Relais mussten aber an die jeweilige Hard- und Software angepasst werden)

Das amerikanische Verteidigungsministerium förderte deshalb das Forschungsinstitut ARPA (Advanced Research Projects Agency). Diese Forschungseinrichtung beschäftigte sich mit Studien über einen allgemeinen Transportdienst für Datenübertragung.

1969 wurde in Zusammenarbeit mit dem MIT das sog. ARPA-Netz mit 4 Knoten in Betrieb genommen. (dieses Netz verband mehrere von der ARPA unterstützte Institute des MIT)

Dieses Netz diente als Grundlage für die Erforschung der Datenkommunikation. Die „Datenfernverarbeitung“ war um 1970 bereits im kommerziellen Bereich verbreitet. Es waren dies aber meist Verbindungen zwischen „gleichartigen“ Rechnern, die eine bestimmte Kopplung hatte. Diese Kopplungen mussten je nach Hard- u. Software immer neu entwickelt werden. Manchmal mussten zwei Computer die am selben Platz standen sogar über zwei getrennte Leitungen an das Netz angeschlossen werden. Dies bedeutete aber eine Verschwendung von Entwicklungs- und Netzwerkmitteln.

Es musste eine Architektur für die Datenkommunikation entwickelt werden.

Um 1973 erreichte man mit der herstellerunabhängigen Protokollfamilie TCP/IP (Transport Control Protocol/ Internet Protocol) diese erwünschte Verbindung verschiedenartiger Rechner. Dieses Protokoll entspricht in den untersten Schichten dem OSI-Modell.

Besondere Bekanntheit erreichte das Internet durch Entwicklung des WWW. Es beruht auf dem HTTP-Protokoll, welches Informationsquellen unabhängig vom Ort und

Format der Speicherung mit der Benutzeroberfläche zusammenführt. Es wurde die Sprache HTML entwickelt, die es ermöglicht, Informationen betriebssystemunabhängig zu verfassen.

HTML ist für statische Dokumente konzipiert. Animationen und interaktiv veränderliche Inhalte sind über den Umweg eines externen Programmaufrufes mittels CGI (Common Gateway Interface) oder über direkt ausführbare Java-Applets möglich. Ein Nachteil von HTML ist die schlechte Wartbarkeit von großen Dokumenten (Formatierungsangaben und tatsächliche Info nicht klar trennbar). Dies führte zur Entwicklung der Sprache XML. (eXtensible Markup Language) Für Computer, die am Internet angeschlossen sind und Prozesse laufen haben, die auf Befehle von außen reagieren (z.B. Mailserver, WWW-server,...) besteht ein Sicherheitsrisiko. Möglichkeit der Sicherung: Firewall (reglementiert und protokolliert den Zugriff von außen; man sieht von außen nur die Firewall).

---

### FRAGE 6.10: Beschreibe die Netzwerkarchitektur DNA.

#### Antwort: ^

Bevor sich das Internet durchgesetzt hatte, waren mehrere kommerzielle Netzwerkarchitekturen verbreitet. DNA war eine davon.

DNA (DEC Network Architecture) ist die Netzwerkarchitektur der ehemaligen Firma DIGITAL (1999 von Compaq übernommen). DNA ist funktionsgeschichtet und für die Programm/Programm-Kommunikation zwischen gleichberechtigten Computern konzipiert.

DIGITAL entwickelte zusammen mit XEROX und INTEL das Konzept für Ethernet, dessen Protokolle Bestandteil von DNA sind. Die Software-Produkte zur Unterstützung von DNA wurden unter der Bezeichnung DECnet vertrieben.

---

### FRAGE 6.11: Beschreibe die Netzwerkarchitektur SNA

#### Antwort:

SNA (Systems Network Architecture) wurde 1974 von IBM unter der Bezeichnung „Teleprocessing“ angekündigt.

Bei der Entwicklung von SNA stand der Aspekt der gemeinsamen Nutzung zentraler oder regionaler Betriebsmittel durch einfache, kostengünstige Endgeräte im Vordergrund. Der Aufbau ist deshalb zentralistisch und hierarchisch orientiert und besteht aus Instanzen unterschiedlicher Komplexität und Mächtigkeit, zwischen denen vorwiegend Master/Slave-Beziehungen herrschen.

SNA wurde laufend weiterentwickelt und es wurden auch gleichberechtigte Beziehungen eingebracht (Anforderung der verteilten Datenverarbeitung). IBM bietet auch SNA Anbindungen an TCP/IP und OSI an.

---

### FRAGE 6.12: Beschreibe die Netzwerkarchitektur TRANSDATA..

#### Antwort:

TRANSDATA ist die Netzwerkarchitektur von SIEMENS. Das erste TRANSDATA-Netz wurde 1974 installiert. Zu dieser Zeit stand auch bei SIEMENS die

Datenfernverarbeitung auf den Fernanschluss einfacher Datenstationen an zentrale Computer im Vordergrund. Bald wurde jedoch die Kommunikation zw. Programmen verschiedener Computer unterstützt. Ab 1979 wurde TRANSDATA auf die verteilte Datenverarbeitung mit einem Funktionsverbund von Computern unterschiedlicher Mächtigkeit erweitert.

Der wichtigste Aspekt bei der Entwicklung von TRANSDATA war immer die Fähigkeit, über alle Netztypen kommunizieren zu können. (insbesondere auch über öffentliche Wählnetze)

TRANSDATA hat eine geschichtete Architektur und die Computer sind gleichberechtigt (peer-to-peer) gekoppelt.

---

### FRAGE 6.13: Welche Absprachen sind bei der Übermittlung von Nachrichten notwendig?

#### Antwort:

Durch Verteilung der Arbeit auf mehrere Prozessoren kann die Leistung eines Systems erhöht werden. Man spricht von fest gekoppelten (nutzen gemeinsamen Hauptspeicher) Multiprozessor-Systemen.

Komplexer wird diese Angelegenheit bei lose gekoppelten Systemen (bei unabhängigen Prozessoren, die keinen gemeinsamen Hauptspeicher haben; sowie bei individuellen Betriebssystemen). Die Aufgaben können nur über Nachrichten koordiniert werden.

Hier sind besondere Absprachen bzgl. Übermittlung dieser Nachrichten möglich:

- Übereinstimmung von Spannungspegel und Übertragungsgeschwindigkeit bei Sender und Empfänger.
- Absprachen über die Sicherheitsinformation und über die Art, wie Übertragungsfehler gemeldet und korrigiert werden.
- Vereinbarungen über die Struktur von Adressen.
- Absprachen über die Art, wie die Darstellung von Text gesteuert wird. (Zeilenumbruch, Tabulator,...)

---

### FRAGE 6.14: Was ist ein Protokoll?

#### Antwort:

Ein Protokoll für einen Dienst ist die Summe aller Absprachen für die koordinierte Erbringung dieses Dienstes über mehrere Computer hinweg.

Es regelt den Austausch von Infos zwischen den an diesen Dienst beteiligten Instanzen. Diese Infos werden den Nachrichten höherer Schichten als Nachrichtenkopf oder Header vorangesetzt. Das Protokoll regelt auch die Darstellung (Syntax) und Bedeutung (Semantik) der ausgetauschten Informationen und deren zeitliche Abfolge.

---

## FRAGE 6.15: Was sind livelock und deadlock?

### Antwort:

Bei der Definition des Protokolls muss darauf geachtet werden, dass es nicht zu Blockierungszuständen kommen kann. Bei diesen unterscheidet man zwei Arten:

- deadlock: Keiner der beiden Partner kann weiterarbeiten, da jeder auf eine Aktion des anderen wartet und das Protokoll diesen Blockierungszustand nicht aufheben kann.
  - livelock: Beide Partner sind so miteinander verklemmt, dass sie zwar weiterhin Signale austauschen, aber trotzdem keinen realen Arbeitsfortschritt erzielen.
- 
- 

## FRAGE 6.16: Wie kann man die Zusammenarbeit zweier Instanzen organisieren?

### Antwort:

In einem geschichteten System muss es für jede Schicht mindestens 1 Protokoll geben. Bei Computern, die mehrere Datenstationen unterstützen können, kann die Übertragung nicht mehr als fehlerfrei angenommen werden (da Stationen meist weit entfernt sind)=> es gibt ein entsprechendes Sicherheitsprotokoll. Die zeichenweise Übertragung (vgl. Mensch-Computer über Tastatur) mit Echo ist aufgrund der langsamen Übertragungsgeschwindigkeit nicht mehr akzeptabel.=> es werden Blöcke übertragen. Es müssen also zusätzlich Dateitransferprotokolle abgesprochen werden.

Die Art der Zusammenarbeit zweier Instanzen kann nach zwei verschiedenen Strategien organisiert werden:

- Master/Slave-Prinzip: Wird bei der Kopplung mehrerer Datenstationen an einen Computer verwendet. Dieses hierarchische Prinzip besteht aus Master und Slave-Instanzen. Der Master ist die Leitinstanz und vergibt Senderechte an die Slaves, die von alleine aus nicht aktiv werden können. Die Abläufe der Kommunikation sind übersichtlich und gut beherrschbar. Es ergibt sich jedoch der Nachteil, dass die Kommunikation zwischen zwei Slaves sehr aufwendig ist (nur über Master möglich)
  - Peer-to-peer-Kommunikation: Ist eine gleichberechtigte Kommunikation, bei der jede der beiden Partnerinstanzen von sich aus aktiv werden können. Diese Art der Zusammenarbeit ist wegen möglicher Überschneidungen komplizierter und schwieriger zu koordinieren. Sie ist jedoch flexibler und erleichtert die Zusammenarbeit untereinander.
- 

## FRAGE 6.17: Wie schaut die 3-stufige Struktur der Computernetze in größeren Unternehmen aus?

**Antwort:**

Für größere Unternehmen ist eine 3-stufige Struktur des Computernetzes vorteilhaft:

- Personal- und Kleincomputer: für die Aufgaben einzelner Mitarbeiter oder Projektgruppen.
- Abteilungscomputer: für die Speicherung und Verarbeitung der in der Abteilung gemeinsam genutzten Daten.
- Zentralcomputer: dient zur Haltung und Manipulation von Daten, die das gesamte Unternehmen eines Standortes betreffen. Zentralcomputer kann auch durch einen Supercomputer ersetzt bzw. ergänzt werden. Ein sog. Supercomputer ist ein Rechner mit überdurchschnittlich hoher numerische Rechenleistung., dessen Leistungen von verschiedenen Firmenstandorten genutzt werden können.

Durch diese Gliederung ergibt sich eine hierarchische Struktur. Die Rechenleistung und die Kapazität der Peripheriespeicher steigen mit zunehmender hierarchischer Ordnung.

---

---

## FRAGE 6.18: Unterschied physikalische und logische Struktur eines Computernetzes.

### Antwort:

In einem Computernetz unterscheidet man zwei Aspekte der Strukturierung:

- Physische Struktur: beschreibt die Hardwaregegebenheiten des Netzes
- Logische Struktur: zeigt die Kommunikationsbeziehungen zwischen abstrakten Instanzen.

Physikalische und logische Struktur müssen nicht übereinstimmen.

Das einfachste Bild eines Kommunikationssystems für die verteilte Datenverarbeitung ist das von untereinander verbundenen Netzknoten, an die jeweils Endbenutzer angeschlossen sind. Wie ein Endbenutzer an einen Knoten angeschlossen ist, wird nicht im OSI-Modell beschrieben. Z.B. ist ein Programm als Endbenutzer über eine Programmschnittstelle mit seinem Netzknoten verbunden; eine Datenstation ist über eine Leitung mit dem Computer, der den Netzknoten simuliert, verbunden,...

Aus Kostengründen ist nicht jeder Knoten mit jedem anderen Knoten verbunden => Nachrichten müssen über Zwischenknoten zum Endknoten geleitet werden. Alle Netzknoten sind aber gleichberechtigt (peer-to-peer) gekoppelt. Bei SNA gibt es auch Knoten mit verminderter Funktionalität (Cluster)

---

## FRAGE 6.19: Was ist das client/server-Prinzip?

### Antwort:

Auch in Netzen mit gleichberechtigten Instanzen können bestimmte Betriebsmittel (z.B.: teure Laserdrucker usw.) gemeinsam nach dem client/server-Prinzip verwendet werden.

Der „Kunde“ (client) erteilt einen Auftrag an einen spezialisierten Dienstcomputer (server).

Client und server können untereinander gleichberechtigt kommunizieren und auch die Rollenverteilung ist nicht unbedingt fest zugeordnet. (server kann auch client eines anderen Computers sein)

---

## FRAGE 6.20: Vergleichen Sie die zentrale und dezentrale Netzstruktur.

### Antwort:

Vorteile der zentral orientierten Netzstruktur:

- Erlaubt dynamisch änderbare Zuteilung von Computerleistung und Hauptspeicher auf die einzelnen Stationen. => da nicht alle Anwender

gleichzeitig die volle Leistung benötigen=> Summe der Betriebsmittel kann kleiner bemessen werden=> Gesamtsystem wird billiger

- Zusammenarbeit der Anwender wird durch gleichartige Arbeitsplätze vereinfacht
- Nachrichten gehen nicht verloren wenn eine Station während des Empfangens nicht betriebsbereit ist, da die Daten zentral gespeichert werden.
- Die an einer Stelle gespeicherten Daten können besser gegen unbefugten Zugriff gesichert werden.
- Ausfälle von Übertragungswegen wirken sich nur auf einen Teildes Netzwerkes aus und sind leichter zu diagnostizieren.

Vorteile der dezentralen Netzstruktur (gleichberechtigt gekoppelte Instanzen):

- Höhere Übertragungsgeschw. für den Datenaustausch zwischen Anwendern
- Zuverlässigeres Gesamtsystem, da bei Ausfall eines Computers alle anderen weiterarbeiten können. Bei Ausfall eines Knotens =>nur Inselbetrieb möglich.
- Leistung kann durch Zuschalten von Computern erhöht werden und ist nicht durch einen zentralen Rechner bestimmt.
- Da die Bedienung am jeweiligen dezentral gelegenen Computer erfolgt =>komfortable auf den einzelnen Anwender zugeschnittene Bedienung möglich; es kann auch zeichenweise auf die Eingabe reagiert werden und somit eingabefehler sofort erkannt werden. (Datenstation muss Eingabeblocke zum zentralen Computer übertragen der den die Kontrolle durchführt)

Eine zentral orientierte Netzstruktur verwendet man dann, wenn auf zentrale Datenbestände sehr oft zugegriffen werden soll. Es kann aber auch in einer dezentralen Netzstruktur ein Server als Datenspeicher verwendet werden.

## FRAGE 6.21: Welche Netzwerke gibt es?

### Antwort:

Die optimale Technologie des Übertragungssystems hängt von der räumlichen Ausdehnung des Netzes ab:

- LAN (Local Area Network): Netz für den Nahbereich (verbindet z. B. Rechner eines Stockwerkes oder mehrerer benachbarter Gebäude)
- WAN (Wide Area Network): hier können interkontinentale Verbindungen auftreten
- MAN(Metropolitan Area Network): bezeichnet ein z.B. auf der Basis des Kabelfernsehnetzes realisiertes Netzwerk für eine ganze Stadt. Dieser Begriff stammt aus den USA. Genaue Normen sind noch nicht festgelegt worden.
- GAN (Global Area Network): Kommunikationssystem, welches durch Verwendung von Satellitenstrecken praktisch keiner räumlichen Beschränkung unterliegt. GANs arbeiten mit regionalen WANs zusammen. Die End-to-end-Verzögerung ist relativ hoch.

Für die Datenübertragung über große Distanzen müssen fremde Leitungen eingesetzt werden. In den meisten Ländern hat eine staatliche Fernmeldeverwaltung (wird als PTT bezeichnet) das Monopol auf die Übertragung von Daten über private Grundstücksgrenzen hinweg. In Österreich ist das die Bundespost.

Die PTTs stellen Standleitungen (Leased Lines) und Wählleitungen (Switched Lines) zur Verfügung. Wählleitungen sind Verbindungswege, die nicht fest geschaltet sind und nur auf Aufforderung verwendet werden können. Die Gebühren sind deshalb hier zeit- oder volumenabhängig. Bsp: Telefonnetz

---

## FRAGE 6.22: Übertragungsgeschwindigkeiten von Kabeln?

### Antwort:

Aus Kostengründen kommt für die Datenübertragung nur serielle (Bitweise) Übertragungstechnik zum Einsatz. Die einfachste Möglichkeit ist die Verwendung von verdrehten Kupferleitungen (wie bei Telefonnetzen zw. Teilnehmer und Ortsamt). Diese Leitungen sind aber stör anfällig und eignen sich nur für kurze Entfernungen (abh. von der Geschw. der Übertr.) Bei 100Mbit/s rund 100m und bei 250kbit/s ca. 2km.

Auf Telefonleitungen sind Geschw. von einigen 10kbit/s üblich. (Ausnahme: DSL) Durch Verwendung von Koaxialleitungen erreicht man Ggeschw. von über 100Mbit/s.

Noch schneller geht's mit Glasfaserleitungen (bis über 2Gbit/s) Satellitenübertragung erlaubt Geschw. von einigen Mbit/s.

Funk und Infrarotübertragungen werden nur selten eingesetzt.

Bei LANs werden Koaxialkabeln eingesetzt (34Mbit/s). Für die Kopplung von Großcomputern verwendet man optische Übertragung (bis 622Mbit/s)

Neue Standards bezeichnen sich mit FDDI (Fibre Distributed Data Interface) mit 100Mbit/s und ATM (Asynchronous Transfer Mode) mit 155Mbit/s und 622Mbit/s. Die Übertragungsgeschw. auf normalen Telefonleitungen beträgt bis zu 56kBit/s. Es gibt aber auch spezielle von der Bundespost betriebene Direktverbindungen mit 2-155Mbit/s.

In vielen Ländern wird ISDN(Integrated Services Digital Network) eingesetzt, dass über drei Kanäle ( $2 * 64\text{kbit/s} + 1 * 16\text{kBit/s}$ ) Sprache und Daten gleichzeitig überträgt.

Seit 1999 werden in einigen Ländern Breitbandnetze auf Basis der DSL-Technik(Digital Subscriber Line) von den Telfongesellschaften angeboten. Durch spezielle Codiervverfahren erreicht man damit auf normalen Telfonleitungen Geschw. bis zu 13Mbit/s. Eine Variante dieses Verfahren bezeichnet sich mit ADSL(Asymmetric Digital Subscriber Line). Dieses Verfahren erlaubt ein Empfangen mit 256kbit/s-8Mbit/s und ein Senden mit 64-640kbit/s. ADSL überträgt mit ATM. Datentransport über Fernsehkabelnetze ist für Empfangsraten von bis zu 2Mbit/s geeignet. Durch die Beschränkung des Fernsehkanals auf 24Mbit/s ergibt sich bei vielen Teilnehmern eine viel niedrigere Übertragungsgeschwindigkeit. (bei ADSL keine solche Einschränkung)

Mobilfunknetze ermöglichen mit UMTS(Universal Mobile Telecommunications System) Datenraten von 384kbit/s bis 2Mbit/s. Das GPRS (General Packet Radio Service) gestattet bestenfalls 50kbit/s.

---

## FRAGE 6.23: Beschreiben Sie die Netztopologien in der verteilten Datenverarbeitung.

### Antwort:

Die realisierten Wege der Übertragung von Nachrichten zw. Kommunikationspartnern ist für diese unsichtbar. Für den Netzbetreiber ist diese Struktur aber sehr wichtig, da sie die Kosten und die Funktionsfähigkeit des Systems bestimmt. Man unterscheidet wieder zwischen physikalischer Struktur (d.h. Topologie der verwendeten Leitungen) und der logischen Struktur (d.h. Topologie der Verkehrsbeziehungen).

Verwendet man Wählleitungen, so ist die physikalische Topologie Sache der Fernmeldevermittlung. Das logische Bild der Netzwerktopologie ist dabei das der kompletten Vermaschung. Wählleitungen sind aber nicht immer einsetzbar (niedrige Datenraten und hohe Kosten bei großem Verkehrsaufkommen) => Standleitungen werden für Computernetz erforderlich. Mit Standleitungen ist eine vollständige Vermaschung nicht mehr bezahlbar. Eine teilweise Vermaschung ist bei der Kommunikation zwischen gleichberechtigten Computern vorteilhaft und kann nach unterschiedlichen Strukturen erfolgen:

- Stern: Alle Teilnehmer sind über eine Zentrale Stelle miteinander verbunden. Diese Stelle muss Vermittlungsaufgaben übernehmen. Durch Erweiterung eines Sternnetzes durch Unter-Sternnetze kommt man zum Baum. Stern und Baumnetze sind hierarchische Netzstrukturen, die bei Unternehmen verwendet werden (Abteilungscomputer, Zentralcomputer, Personal- u. Kleincomputer). Funktionsfähigkeit des Netzes hängt von der Verfügbarkeit der Zentrale ab.
- Ring: dient zur Kopplung gleichwertiger Teilnehmer im lokalen Bereich. Die Übertragung erfolgt dabei in einer bestimmten Richtung. Jeder Teilnehmer prüft, ob das Datenpaket für ihn bestimmt ist und gibt es gegebenenfalls weiter. Bei Ausfall eines Teilnehmers wird der Ring lahmgelegt. Gegenmaßnahmen: entweder besitzt jeder Computer eine Einrichtung zur Überbrückung bei Ausfall oder die Leitungen werden über einen gemeinsamen Schaltkasten geführt. Dieser Schaltkasten ist als passives, sehr ausfallsicheres Element realisiert, der die entsprechenden Überbrückungen vornimmt. => logische Struktur: Ring; physikalische Struktur: Stern
- Bus: alle Teilnehmer sind an ein gemeinsames Medium (meist Koaxialkabel) angeschlossen. Eine ausgesendete Nachricht wird von allen Teilnehmern empfangen. Die Zuteilung des Mediums an die einzelnen Teilnehmer muss geregelt werden, da nur jeweils ein Teilnehmer senden kann. Jeder Teilnehmer kann das gesamte System lahm legen indem er ein Dauersignal sendet. Die Reichweite eines Netzes mit Bustopologie ist stark beschränkt. => nur für lokale Kopplungen

Große Computernetze sind meist mit gemischter Topologie aufgebaut. Größere Computer sind dabei vermascht und kleinere Computer sind über eine Bus- oder Ringtopologie verbunden. Eine vieldiskutierte Topologie für LANs besteht aus Kopplungen von lokalen Ring oder Bus-Topologien über LAN-Bridges an WANs.

---

## FRAGE 6.24: Was ist die hierarchische Adressierung?

### Antwort:

Eines der größten Probleme der verteilten Datenverarbeitung ist die Adressierung. Dem Diensterbringer muss die Adresse des gewünschten Teilnehmers mitgeteilt werden. Bei besonders komfortablen Diensten genügt die alleinige Angabe des Namens.

Die Kopplung von Netzen verschiedener Hersteller wird außer durch verschiedene Netztopologien auch durch Inkompatibilität der Adressierungsverfahren verhindert. Es müssen deshalb zur Verbindung von Netzen verschiedener Hersteller oft aufwendige Mechanismen zur Adressbehandlung implementiert werden.

Unter einer hierarchischen Adressierung versteht man die Aufteilung der Adresse in mehrere Teile (Adressierungsebenen), die sukzessive den Standort des Empfängers angeben (vgl. Ortsvorwahl und Landeskennzahl beim öffentlichen Telefonnetz)

Die ISO hat das Adressierungsproblem im Rahmen des OSI-Modells für offene Systeme durch die Definition einer global eindeutigen, hierarchischen Adressierung gelöst. Dadurch wird jedes System, das sich an die OSI-Vereinbarung hält, erreichbar.

---

## FRAGE 6.25: Was ist Routing?

### Antwort:

Die Funktion der Wegewahl (Routing) bezeichnet die Fähigkeit von Systemen, die eingehenden Nachrichten, die nicht für dieses System selbst bestimmt sind, auf den richtigen Wegen weiterzuleiten.

---

## FRAGE 6.26: Welche Arten von Routing gibt es?

### Antwort:

- **Statische Wegewahl:** ist der einfachste Fall. Dabei sind die Wege zwischen den Partnern starr festgelegt. (statische Routingtabellen). Es ergibt sich der Nachteil, dass bei Ausfall eines Abschnittes alle Systeme, deren Weg über diesen defekten Abschnitt verläuft, nicht mehr erreichbar sind. Eine etwas kompliziertere Variante ist das Umschalten der Wege bei Ausfall durch Administrationskommandos.
- **Alternative Wegewahl:** Zwischen zwei Systemen werden mehrere verschiedene Wege verwendet, die nicht gleichberechtigt sind bzgl. Durchsatzrate, Kosten, Sicherheit usw.. Die Auswahl des Weges richtet sich nach den gewünschten Eigenschaften der Übertragung.
- **Adaptive Wegewahl:** Die Wege der Übertragung werden nach bestimmten Gesichtspunkten ständig neu bestimmt. Durch Wegewahlalgorithmen können verschiedene Gesichtspunkte wie z.B.: Lastverteilung, Leitungsgüte, Durchsatz, Sicherheit, Gebührenoptimierung, Leitungsausfall,... berücksichtigt werden.

Die Wegwahl nach diversen Kriterien ist bei komplexen Netzen sehr rechenaufwendig => ein kleines System kann daher nur einfache Wegwahlalgorithmen benutzen.

---