

# — Informatik I —

## Modul 7: Betriebs- und Kommunikationssysteme



Universität  
Zürich <sup>UZH</sup>



## Modul 7: Betriebs- & Kommunikationssysteme

- ❑ **Betriebssysteme**
- ❑ Auftrags- und Speicherverwaltung
- ❑ Einlagerung, Zuweisung, Ersetzung
- ❑ Kommunikationssysteme
- ❑ Internet

# Definition Betriebssystem

Definition des Begriffs "Betriebssystem" (Operating System):  
z.B. nach dem Deutschen Institut für Normung (DIN 44300):

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

- Ziel eines Betriebssystems ist es, die semantische Lücke zwischen Anwendung (wünscht möglichst hohe Operationen) und Hardware (bietet elementare Operationen) zu verkleinern.

## Semantische Lücke

- Applikation und Hardware sind voneinander entfernt.
  - Semantische Lücke!



- Betriebssystem erweitert die Fähigkeiten der Hardware.
- Betriebssystem verkleinert die semantische Lücke.

# Aufgaben eines Betriebssystems

---

## □ Betriebsmittelverwaltung (*resource management*)

- Verwaltet alle peripheren Betriebsmittel des Rechnersystems, wie z.B.
  - Festplatte
  - Floppy Disk
  - CDROM
  - Drucker, ...

## □ Auftragsverwaltung (Prozeßverwaltung, *tasks*)

- Verwaltet Prozesse und Threads

## □ Speicherverwaltung (*memory management*)

- Durch immer größer werdende Programme sowie mehrerer quasi gleichzeitig laufender Programme wird zur Verfügung stehender Arbeitsspeicher zu klein

# Modul 7: Betriebs- & Kommunikationssysteme

---

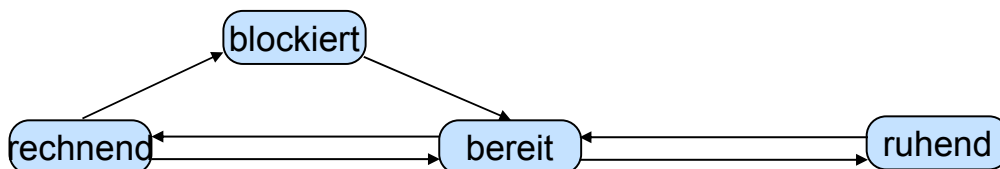
- Betriebssysteme
- **Auftrags- und Speicherverwaltung**
- Einlagerung, Zuweisung, Ersetzung
- Kommunikationssysteme
- Internet

# Auftragsverwaltung

- ❑ Ein Prozeß (Rechenprozeß, Auftrag) ist die Ausführung eines Algorithmus (Ablauf), der durch ein ausführbares Programm beschrieben ist.
  - Ein Prozeß ist ein aktives Programm.
- ❑ Prozesse können
  - eine bessere Ressourcenausnutzung durch das Multiplexing erreichen und
  - eine einfachere Programmierung durch Aufteilung der Aufgaben erreichen.
- ❑ Threads (leichtgewichtige Prozesse):
  - Abläufe im gleichen Adreßraum (Programm- und Datensegment)
  - Separater Registersatz, Programmzähler, Kellerspeicher

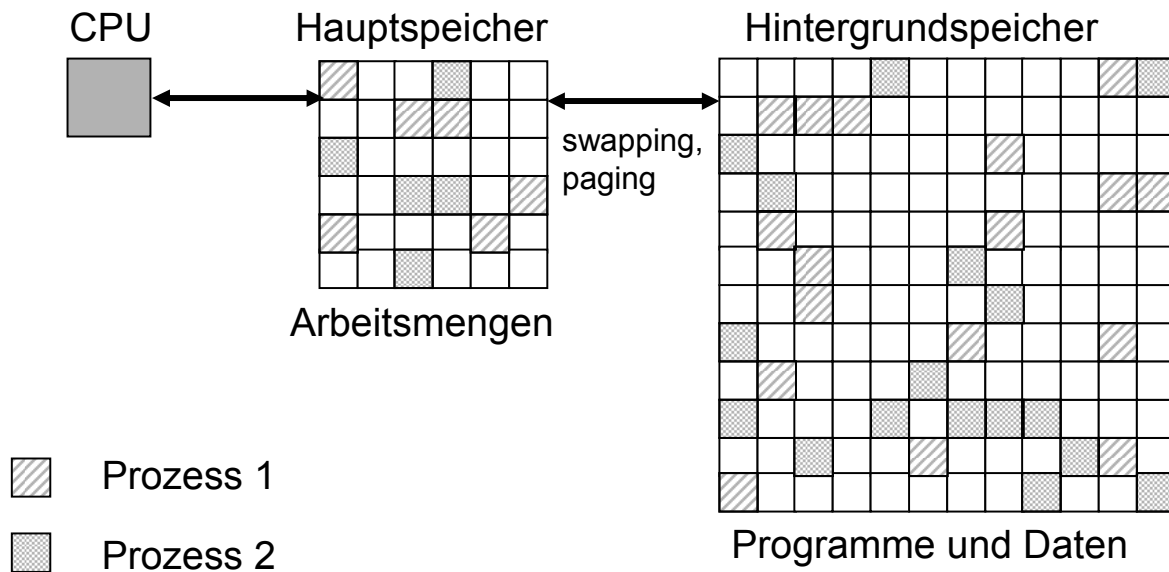
## Prozesse

- ❑ Ein Prozeß kann im Laufe seiner Bearbeitung im wesentlichen folgende unterscheidbaren Zustände (vereinfacht) annehmen (Prozeßzustandsmodell):



- ❑ Operationen zu ihrer Kontrolle:
  - Create, exit, join, mutex\_init, cond\_wait, ...
- ❑ Prozeßkoordination:
  - Mutex, semaphore, wait conditions, ...
- ❑ Inter Process Communication (IPC) und Inter Object Communications:
  - Pipes, sockets, RPC, RMI, COM, DCOM, Corba, .NET, .....

# Grundstruktur virtueller Speicherverwaltung

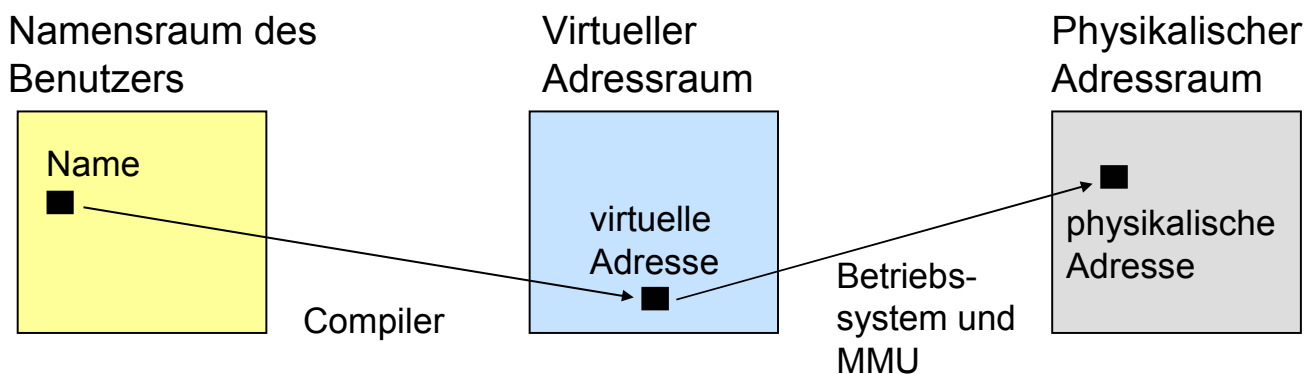


## Virtuelle Speicherverwaltung (1)

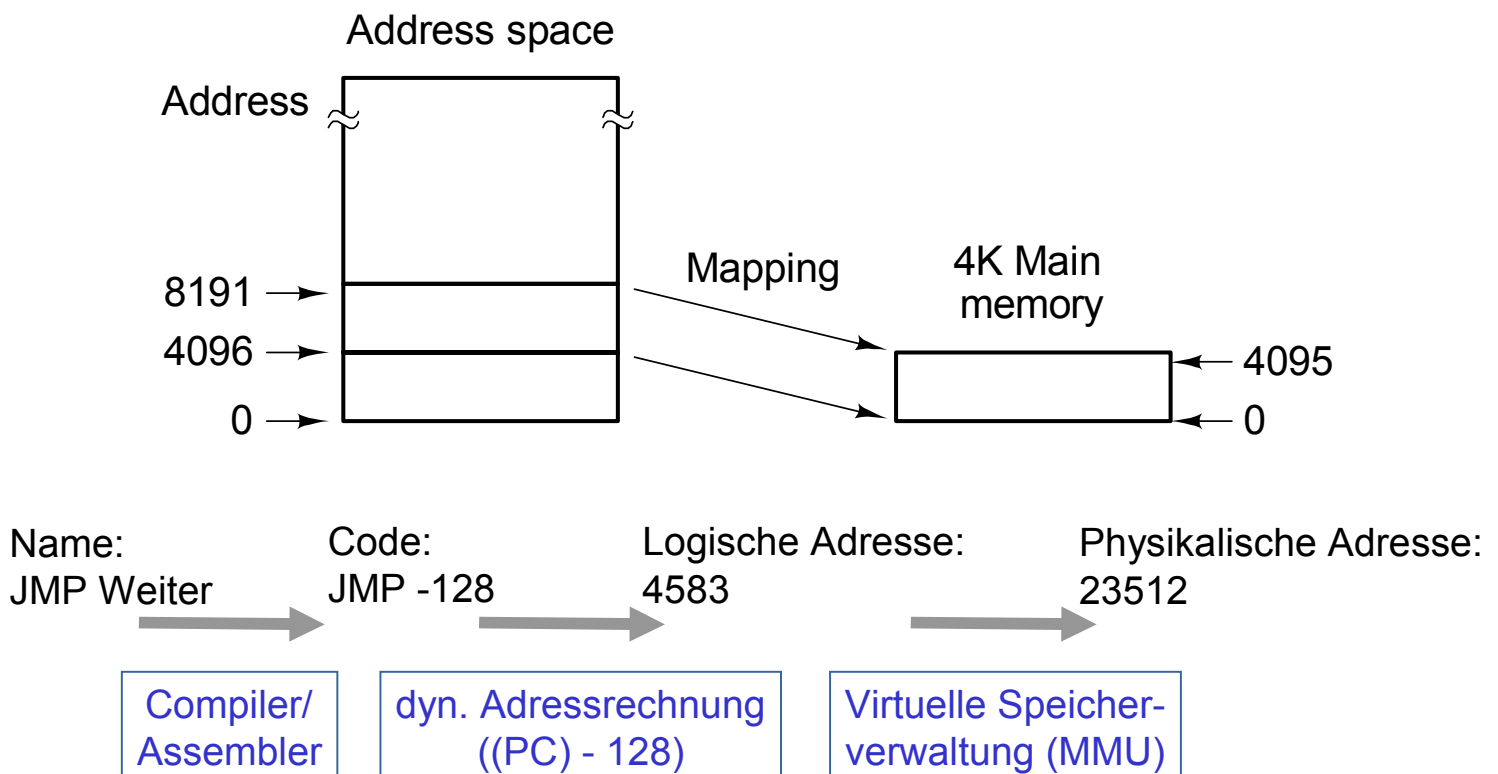
- ❑ Vorgang bleibt dem Anwender völlig verborgen, d.h. Arbeitsspeicher erscheint dem Anwender wesentlich größer, als er ist.
- ❑ Ein nach diesem Konzept verwalteter Speicher heißt virtueller Speicher.
- ❑ Von modernen Prozessoren wird die Verwaltung dieses virtuellen Speichers hardwaremäßig durch eine MMU (memory management unit) unterstützt.
- ❑ Hauptaufgabe dieser virtuellen Speicherverwaltung: Umsetzung virtueller (logischer) Adressen in physikalische Adressen

# Virtuelle Speicherverwaltung (2)

- ❑ Benutzer:
  - Kennzeichnet Objekte (Programme, Unterprogramme, Variablen) durch Namen
- ❑ Compiler:
  - Übersetzt diese Namen in virtuelle (logische) Adressen
- ❑ Virtuelle Speicherverwaltung:
  - Wandelt diese Adressen zur Laufzeit je nach gerade gegebener Speicherbelegung in physikalische Adresse (wirklicher Ort des Objekts im Hauptspeicher) um



## Abbildung virtueller Adressen



# Segmentierungs- und Seitenwechselverfahren

- Es existieren zur virtuellen Speicherverwaltung zwei grundlegende Verfahren: Seitenwechsel und Segmentierung
- Aufteilung in **Seiten (paging)**
  - Hierbei wird der logische und der physikalische Adreßraum in "Segmente fester Länge", sogenannte Seiten (pages) unterteilt.
  - Die Seiten sind relativ klein (256 Byte - 4 kByte)
  - Ein Prozeß wird auf viele dieser Seiten verteilt (keine logischen Zusammenhänge wie bei der Segmentierung)
- Aufteilung in **Segmente (swapping)**
  - Hierbei wird der virtuelle Adreßraum in Segmente verschiedener Länge zerlegt.
  - Jedem Prozeß sind ein oder mehrere Segmente z.B. für den Programmcode und die Daten, zugeordnet.
  - Die einzelnen Segmente enthalten logisch zusammenhängende Informationen (Programm- und Datenmodule) und können relativ groß sein.

## Seiten im virtuellen und realen Speicher

Page	Virtual addresses	Page frame	Physical addresses
15	61440 - 65535	7	28672 - 32767
14	57344 - 61439	6	24576 - 28671
13	53248 - 57343	5	20480 - 24575
12	49152 - 53247	4	16384 - 20479
11	45056 - 49151	3	12288 - 16383
10	40960 - 45055	2	8192 - 12287
9	36864 - 40959	1	4096 - 8191
8	32768 - 36863	0	0 - 4095
7	28672 - 32767		
6	24576 - 28671		
5	20480 - 24575		
4	16384 - 20479		
3	12288 - 16383		
2	8192 - 12287		
1	4096 - 8191		
0	0 - 4095		

(a)

Page frame	Physical addresses
7	28672 - 32767
6	24576 - 28671
5	20480 - 24575
4	16384 - 20479
3	12288 - 16383
2	8192 - 12287
1	4096 - 8191
0	0 - 4095

(b)

# Diskussion

## ❑ Seitenaufteilung

### – Vorteile:

- Durch kleine Seiten wird wirklich benötigter Teil des Programms eingelagert
- Geringerer Verwaltungsaufwand als Segmentierung

### – Nachteil:

- Häufiger Datentransfer

## ❑ Segmentierung

### – Vorteile:

- Segmentierung spiegelt logische Programmstruktur wieder
- Durch große Segmente relativ seltener Datentransfer
- Segmentspezifische Schutzmaßnahmen möglich

### – Nachteile:

- Wenn Datentransfer, dann jedoch umfangreich.
- Besteht Programm nur aus einem Code- und Daten-Segment (wird vom Compiler oder Benutzer festgelegt), so muß vollständig eingelagert werden

## Wesentliche Unterschiede

	Seiten	Segmentierung
❑ Programmierertransparenz?	Nein	Ja
❑ Anzahl linearer Adreßräume?	1	Viele
❑ Kann die virtuelle Adreßgröße die Speichergröße übertreffen?	Ja	Ja
❑ Können Tabellen variabler Größe behandelt werden?	Nein	Ja
❑ Grund der Erfindung?	Emulation großer Adreßräume	Angebot mehrfacher Adreßräume

❑ Viele der heutigen Prozessoren unterstützen beide Techniken!



- ❑ Betriebssysteme
- ❑ Auftrags- und Speicherverwaltung
- ❑ Einlagerung, Zuweisung, Ersetzung
- ❑ Kommunikationssysteme
- ❑ Internet

## Probleme der virtuellen Speicherverwaltung

---

- ❑ Beim Austausch von Daten zwischen Arbeits- und Hintergrundspeicher ergeben sich drei Problemkreise:
  1. Der Einlagerungszeitpunkt
    - Wann werden Segmente oder Seiten in den Arbeitsspeicher eingelagert ?
  2. Das Zuweisungsproblem
    - An welche Stelle des Arbeitsspeichers werden die Seiten oder Segmente eingelagert ?
  3. Das Ersetzungsproblem
    - Welche Segmente oder Seiten müssen ausgelagert werden, um Platz für neu benötigte Daten zu schaffen?

# 1. Einlagerungszeitpunkt

---

- Gängiges Verfahren:
  - Einlagerung auf Anforderung (Demand Paging bei Seitenverfahren)
- Hierbei werden Daten eingelagert, sobald auf sie zugegriffen wird, sie sich aber nicht im Arbeitsspeicher befinden.
- Der Zugriff auf ein nicht im Arbeitsspeicher vorhandenes Segment oder Seite heißt Segment- oder Seiten-Fehler (*segment fault, page fault*).

## 2. Zuweisungsproblem (1)

---

- Bei **Seitenwechselfahren**
  - Dieses Problem ist nicht existent, da alle Seiten gleich groß sind und somit immer “passende Lücken” nutzbar ist
  - keine externe Fragmentierung.
  - Jedoch: Problem der internen Fragmentierung
    - Diese entsteht bei der Aufteilung eines Programms auf die Seiten.
  - Einheitliche Seitengröße
  - auf der letzten Seite jedes Programm-Moduls entsteht mit hoher Wahrscheinlichkeit ein ungenutzter Leerraum.
- Bei **Segmentierungsverfahren**:
  - Hier muß eine ausreichend große Lücke im Arbeitsspeicher gefunden werden.
  - Drei Strategien:
    - *first-fit*: erste passende Lücke wird genommen
    - *best-fit* : kleinste passende Lücke wird genommen
    - *worst-fit*: größte passende Lücke wird genommen

## 2. Zuweisungsproblem (2)

□ Problem bei allen drei Verfahren:

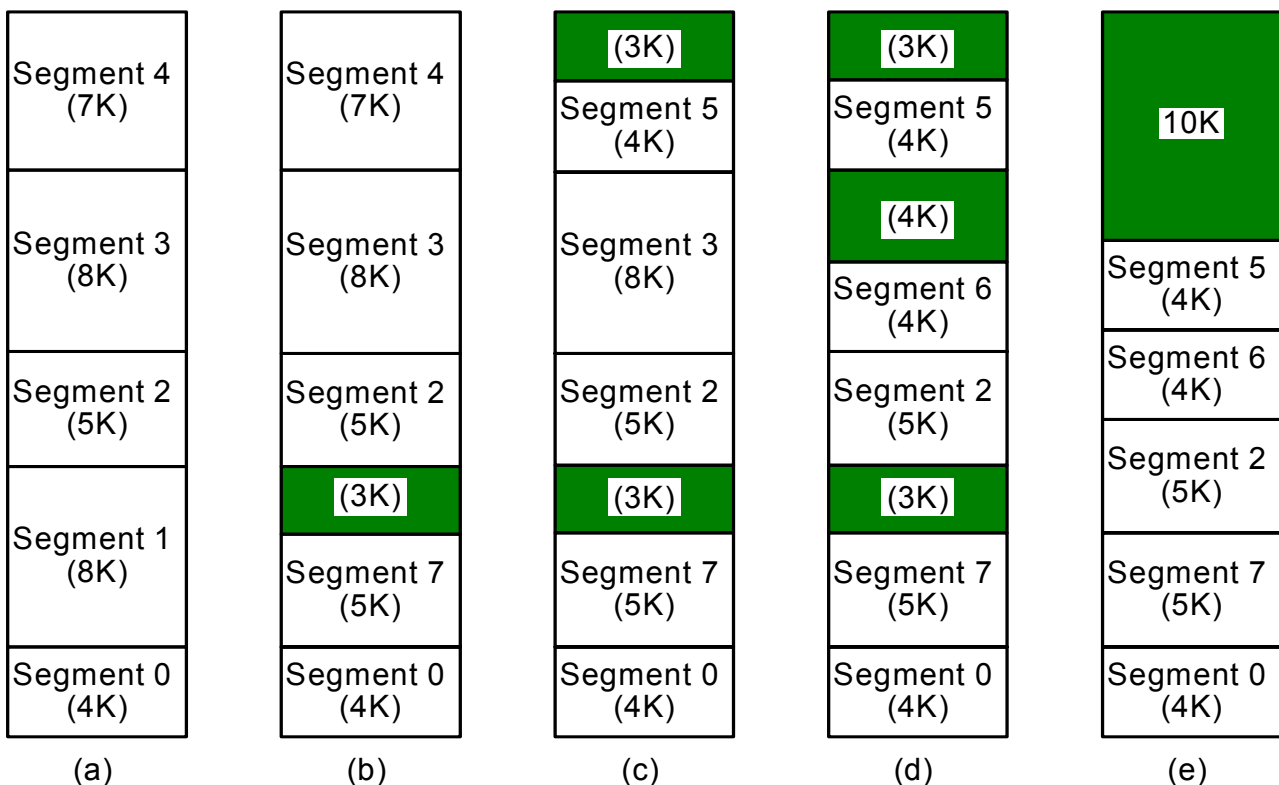
- Der Speicher zerfällt nach einiger Zeit in belegte und unbelegte Speicherbereiche  
 → externe Fragmentierung.

Arbeitsspeicher



- Die unbelegten Speicherbereiche sind hierbei oft zu klein, um Segmente aufnehmen zu können

## Externe Fragmentierung und Kompaktifizierung



# 3. Ersetzungsproblem

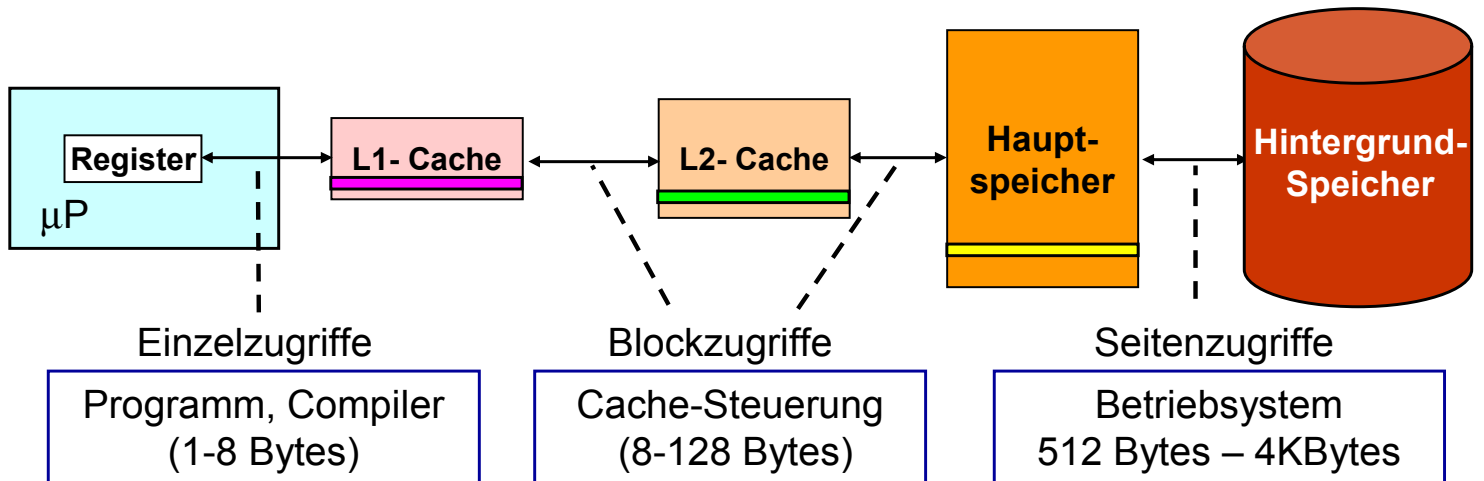
- Bei **Segmentierungsverfahren**:
  - Meist wird Anzahl gleichzeitig von einem Prozeß benutzbaren Segmente limitiert:
    - Bei Einlagerung eines neuen Segments wird ein zuvor für diesen Prozeß benutztes Segment ausgelagert
  - Es ist jedoch auch eine der im folgenden für Seitenwechsel-Verfahren beschriebenen Methoden möglich
- Bei **Seitenwechselverfahren**:
  - **FIFO (first-in-first-out)**: Sich am längsten im Arbeitsspeicher befindende Seite wird ersetzt
  - **LIFO (last-in-first-out)**: Zuletzt eingelagerte Seite wird ersetzt
  - **LRU (least recently used)**: Seite, auf die am längsten nicht zugegriffen wurde, wird ersetzt
  - **LFU (least frequently used)**: seit ihrer Einlagerung am seltensten benutzte Seite wird ersetzt
  - **LRD (least reference density)** : Mischung aus LRU und LFU. Seite mit der geringsten Zugriffsdichte (Anzahl Zugriffe / Einlagerungszeitraum) wird ersetzt

## Segment- und Seitenfehler

- Sowohl bei segmentorientierter wie bei seitenorientierter Speicherverwaltung gilt:
  - Befindet sich eine Seite oder ein Segment nicht im Hauptspeicher, so löst der Prozessor eine Unterbrechung aus, um die Seite oder das Segment durch das Betriebssystem zu laden (Seiten- oder Segmentfehler)
- Erkennung eines **Segmentfehlers**:
  - Bit im Segment-Deskriptor zeigt an, ob Segment im Hauptspeicher ist oder nicht
- Erkennung eines **Seitenfehlers**:
  - Seitennummer befindet sich nicht in der Seitentabelle (Seitenfehler)
  - Spezielles Kennungsbit im Seitentabellen-Verzeichnis (Seitentabellenfehler)

# Speicherhierarchie

- Daten werden nur zwischen aufeinanderfolgenden Ebenen der Speicherhierarchie kopiert.



## Speicherverwaltung (1)

- Das **Betriebssystem** führt über die freien Speicherbereiche Buch und lagert bei nicht ausreichendem Freiplatz im Hauptspeicher diejenigen Speicherinhalte auf den Hintergrundspeicher aus, die gegenüber ihrem Originalen auf dem Hintergrundspeicher verändert worden sind.
- Die eindeutige Abbildung des großen virtuellen Speichers auf die effektive Hauptspeicherkapazität wird von der Hardware durch **Speicherverwaltungseinheiten** unterstützt (*memory management units, MMU*)
- Die erforderliche Abbildungsinformation stellt das Betriebssystem in Form einer oder mehrerer **Übersetzungstabellen** zur Verfügung.

# Speicherverwaltung (2)

---

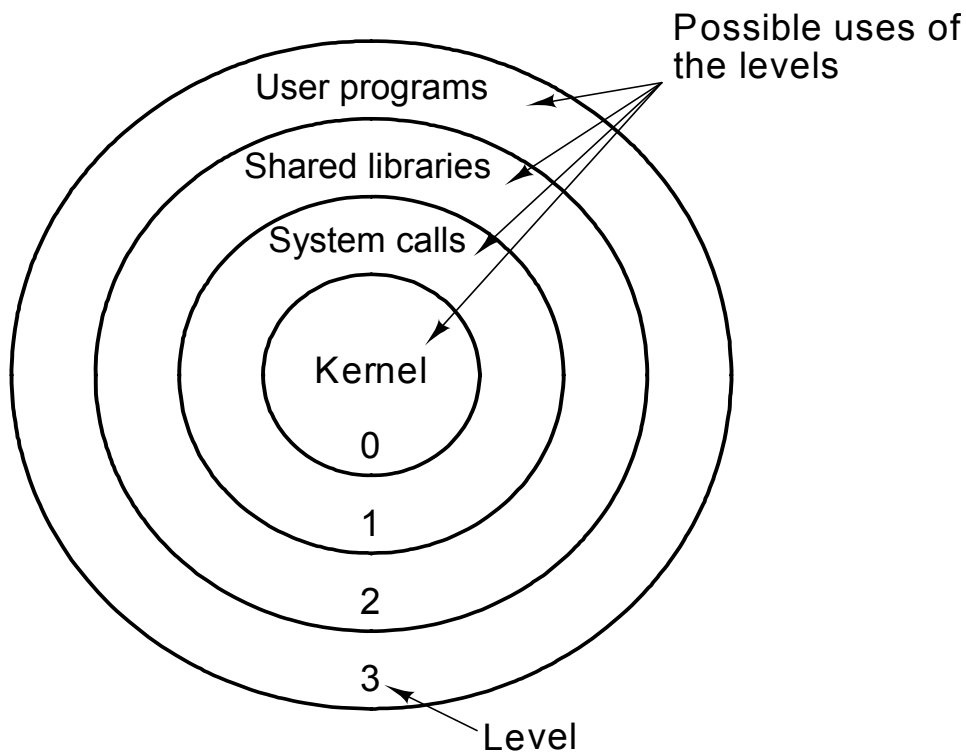
- ❑ Lokalitätseigenschaften von Programmen und Daten:
  - Programme greifen in einem kleinen Zeitintervall auf einen relative kleinen Teil des Adreßraumes zu.
  
- ❑ Die **Lokalitätseigenschaften** gewährleisten eine hohe Wahrscheinlichkeit, daß die Daten, welche die CPU anfordert, im physikalischen Hauptspeicher zu finden sind.
  
- ❑ Zwei Arten der Lokalität:
  - Zeitliche Lokalität:
    - Falls ein Datum oder ein Befehl referenziert wird, so werden sie bald wieder referenziert.
  - Örtliche Lokalität:
    - Falls ein Datum oder ein Befehl referenziert wird, werden bald Daten oder Befehle mit benachbarten Adressen referenziert.

# Schutzmechanismen (1)

---

- ❑ Moderne Mikroprozessoren bieten Schutzmechanismen an, um während der Laufzeit von Programmen unerlaubte Speicherzugriffe zu verhindern.
  
- ❑ Dies geschieht im wesentlichen durch:
  - Trennung der Systemsoftware, z.B. des Betriebssystems Insbesondere des Ein-/Ausgabe-Subsystem (BIOS, basic I/O system), von den Anwendungsprozessen.
  
  - Trennung der Anwendungsprozessen voneinander Ist dies gewährleistet, könnte ein fehlerhaftes Anwenderprogramm andere, fehlerfreie Programme beeinflussen (Schutzebenen und Zugriffsrechte)

# Schutzmechanismen (2)

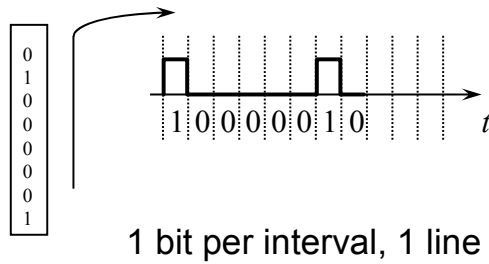


## Modul 7: Betriebs- & Kommunikationssysteme

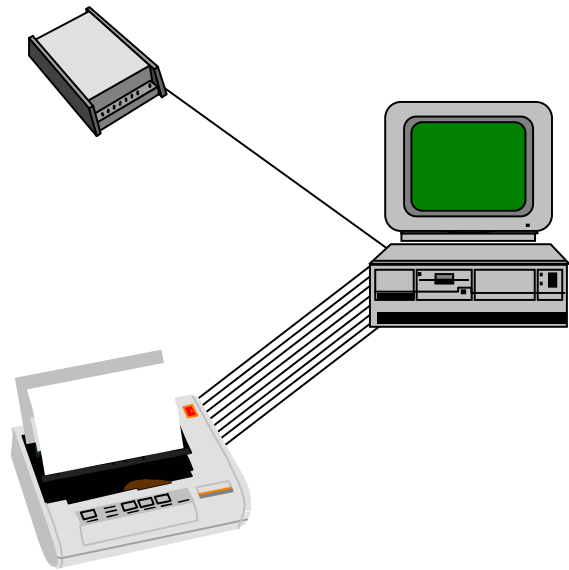
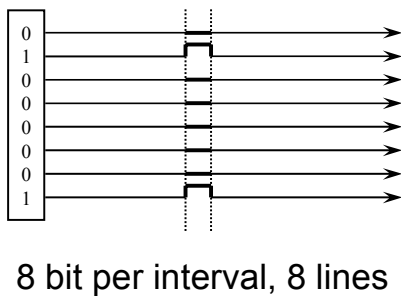
- ❑ BS Definition und Aufgaben
- ❑ Auftrags- und Speicherverwaltung
- ❑ Einlagerung, Zuweisung, Ersetzung
- ❑ **Kommunikationssysteme**
- ❑ Internet

# Transmission Schemes and Interfaces

## – Serial transmission

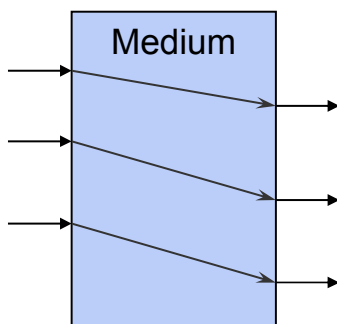


## – Parallel transmission



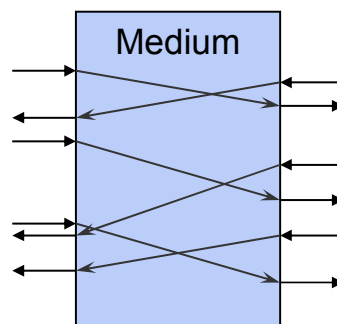
# Directions of Use

## Simplex



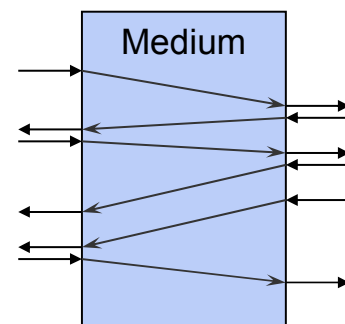
- Fire alarm
- Sensors
- Pager

## Duplex



- Phone

## Half duplex



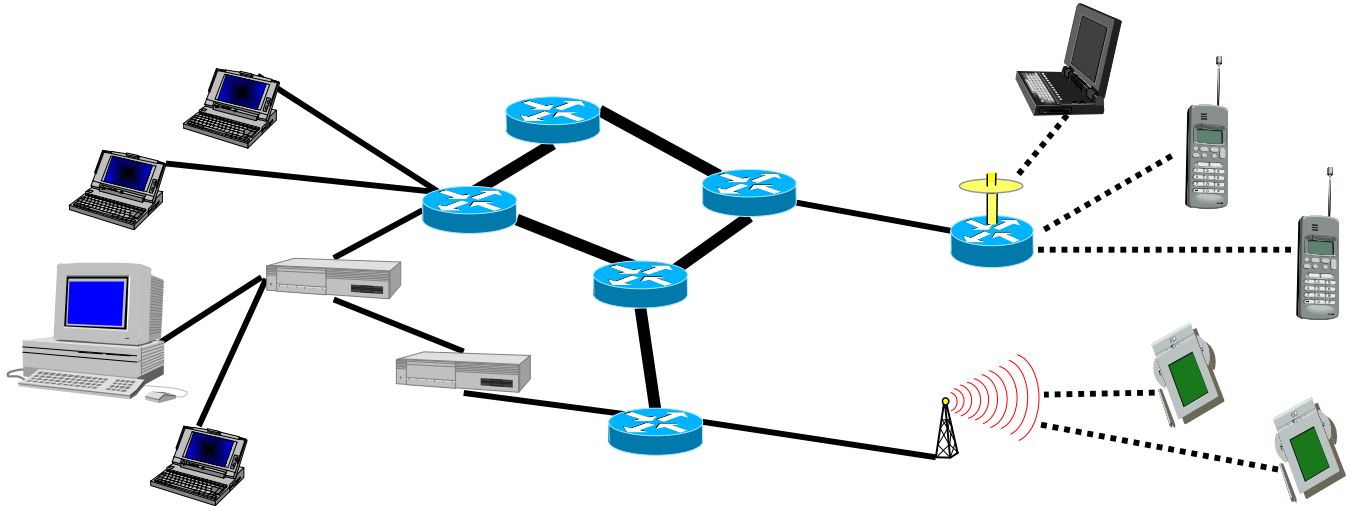
- Inter phone
- Some GSM connections

GSM: Global System for Mobile Communications



# Major Network Components

- ❑ End-systems, terminals, computer, nodes, hosts ...
  - Fixed or mobile
- ❑ Intermediate systems, bridges, routers, gateways ...
- ❑ Links, lines, cables, circuits, “radio links”, channels ...
  - Access and backbone

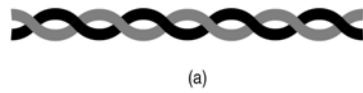


# Tasks of Communication Networks

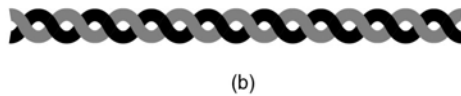
- ❑ Networks provide an infrastructure for:
  - Interconnecting machines or services (connectivity)
    - Functional combinations
    - Performance enhancements
  - Making available scarce resources (resource sharing)
    - Extending functional combinations
    - Independence of user and resource location
  - Equalizing traffic volumes (load balancing)
    - Optimization of network traffic flows
  - Providing alternative fallbacks (reliability)
    - Enhancing availability
    - Enhancing robustness

# Cabling Examples (1)

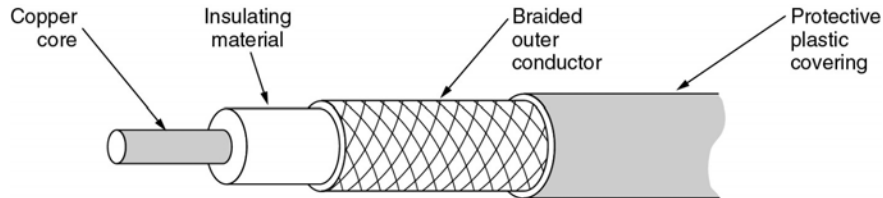
- UTP Cat3



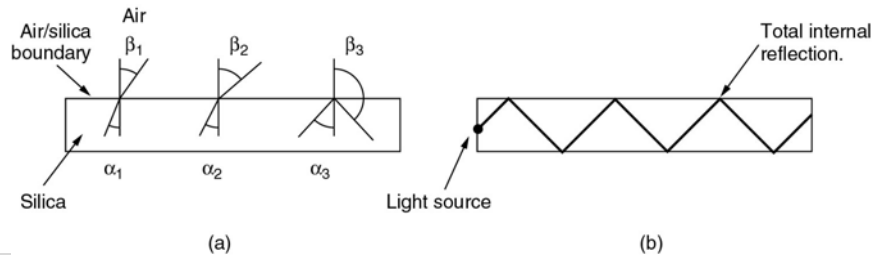
- UTP Cat5



- Copper

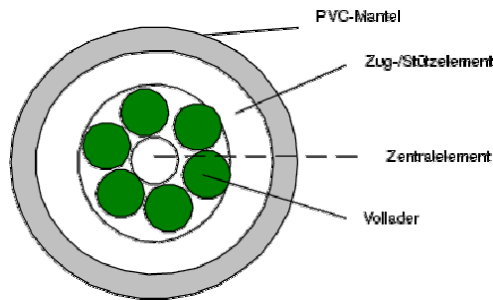


- Optical



# Cabling Examples (2)

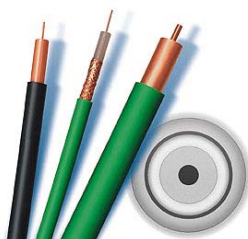
- Fiber optic:



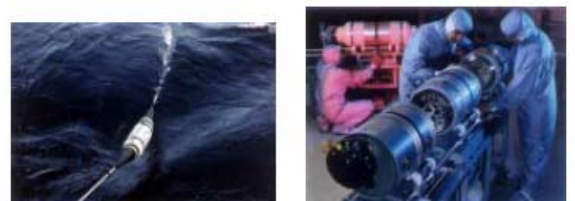
- Twisted Pair:



- Coaxial:



Kabelverlegegeschwindigkeit bis zu 15 km/h in Tiefen bis 7000 m. Das Füllen eines Schiffes mit Kabel dauert i.a. mehrere Wochen.



Unterswasserverstärker: 1-2 m Länge, 30-50 cm Breite, 300-500 kg, 40W, 0.5-1 Mio \$, in bis zu 7000 m Tiefe.

# Network Classification

---

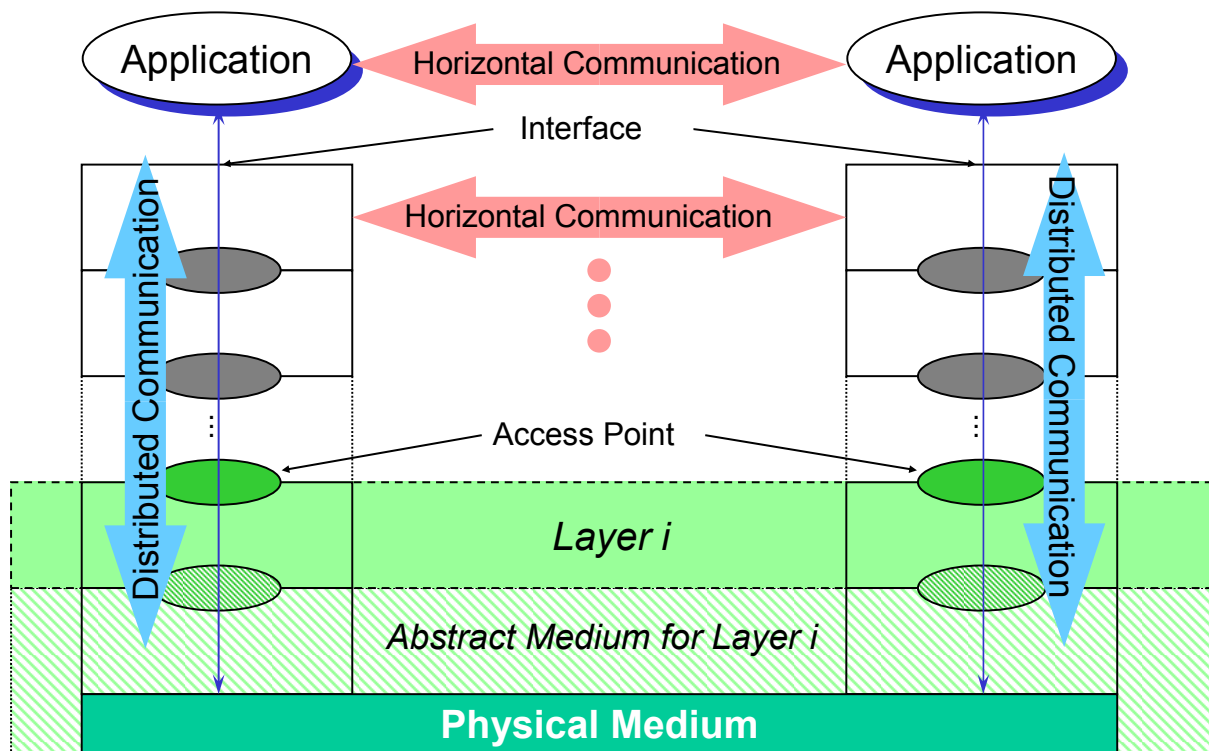
- ❑ **Personal Area Network: PAN**
  - Less than 1 m, personalized, local working area
- ❑ **Storage Area Network: SAN**
  - A couple m, interconnection of computing service, memory
- ❑ **Local Area Network: LAN**
  - 10 m to a couple 1000 m, work groups, departments
- ❑ **Metropolitan Area Network: MAN**
  - City, campus, interconnect for powerful computers
- ❑ **Wide/Global Area Network: WAN/GAN**
  - World-wide reachability, many participants, potential universal accessibility

# Communications Architectures

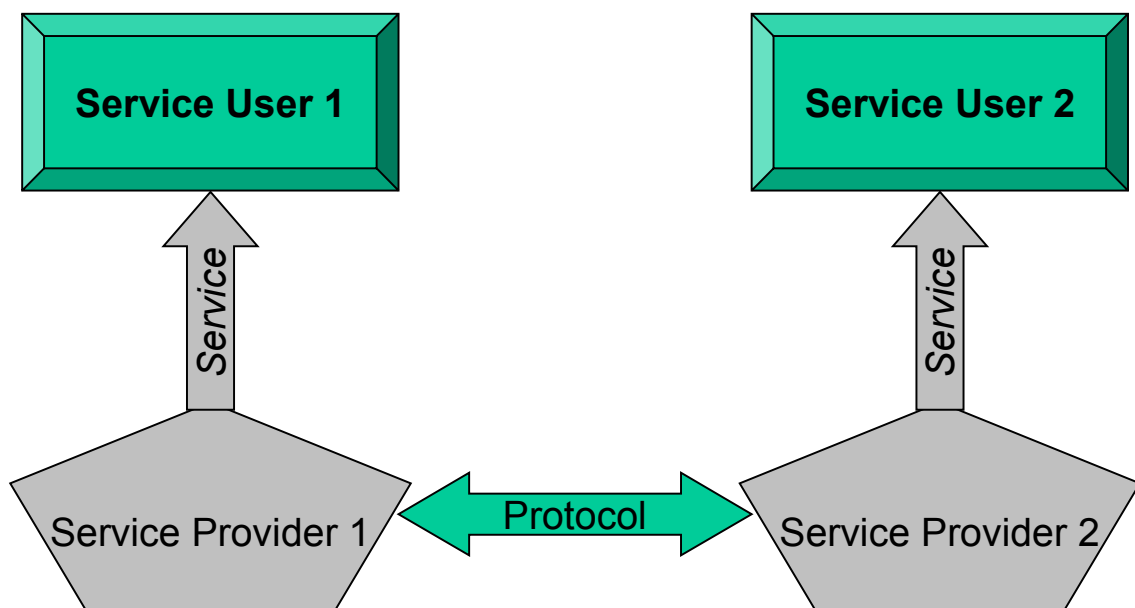
---

- ❑ To specify and implement communication processes a standardized communication architecture is required:
  - Physical connectivity: Copper, optical, radio link
  - Functions:
    - Control of the communication process (start, break, stop)
    - Addressing of communicating partners
    - Guaranteeing quality required
    - Adaptation to different formats
    - ...
  - Interfacing applications
- ⇒ Layered communication architectures, where upper layers make use of services of lower layers

# Layered Communication Systems



# Service and Protocol — Overview



# Connectionless and -oriented Services

---

- ❑ **Connectionless service:**
  - Each data transfer is considered separately
  - No connection status known or even required
  
- ❑ **Connection-oriented service:**
  - Set up of a connection between service users of the (N)-Layer by (N-1)-Entities
  - Request performed by (N-1)-Layer service primitives
  - Protocol-dependent negotiation of transmission parameters: partners, Quality-of-Service (QoS), route
  - Data transfer in this connection under the constraints of the context of the connection (connection status)
  - Tear down of the connection required (im- or explicit)

## ISO/OSI Basic Reference Model (BRM)

---

- ❑ **International standardization of services and protocols to implement “Open System Interconnection (OSI)”:**
  - Basis for enabling the communication for systems of different vendors
  - BRM serves as a guideline to understand and model all tasks for communications
  - Major implementations existed in Europe in the late 80’s
  - The Internet model took over in the 90’s
  
  - ISO/IEC IS 7498: Information Processing Systems - Open Systems Interconnection - Basic Reference Model, International Standard, October 15, 1984
  - Included in ITU-T Recommendation X.200

# OSI Layers — Overview

---

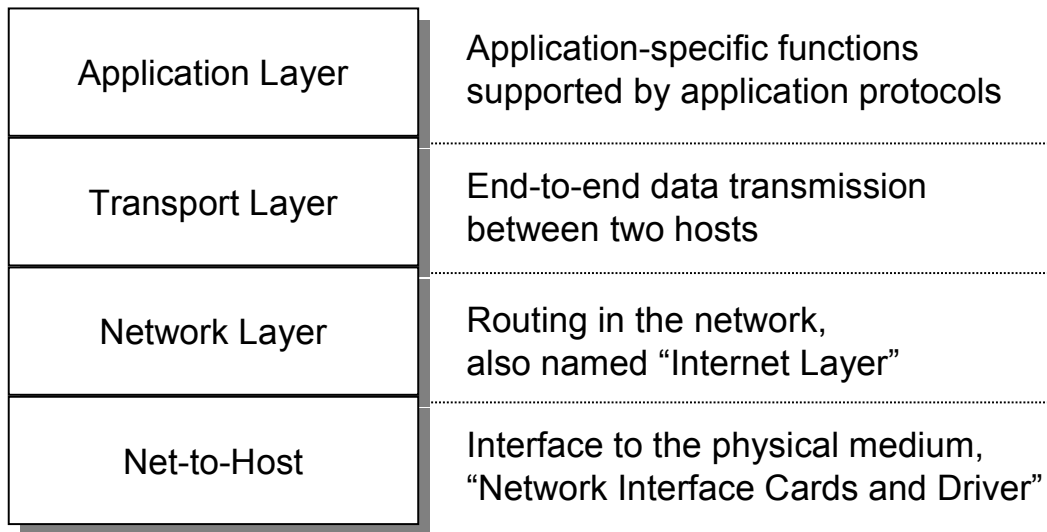
<i>Anwendungsschicht</i>	Layer 7 (A-Layer)	<b>Application Layer</b>
<i>Darstellungsschicht</i>	Layer 6 (P-Layer)	<b>Presentation Layer</b>
<i>Kommunikations- steuerungsschicht</i>	Layer 5 (S-Layer)	<b>Session Layer</b>
<i>Transportschicht</i>	Layer 4 (T-Layer)	<b>Transport Layer</b>
<i>Vermittlungsschicht</i>	Layer 3 (N-Layer)	<b>Network Layer</b>
<i>Sicherungsschicht</i>	Layer 2 (DL-Layer)	<b>Data Link Layer</b>
<i>Bitübertragungsschicht</i>	Layer 1 (Ph-Layer)	<b>Physical Layer</b>

## Modul 7: Betriebs- & Kommunikationssysteme

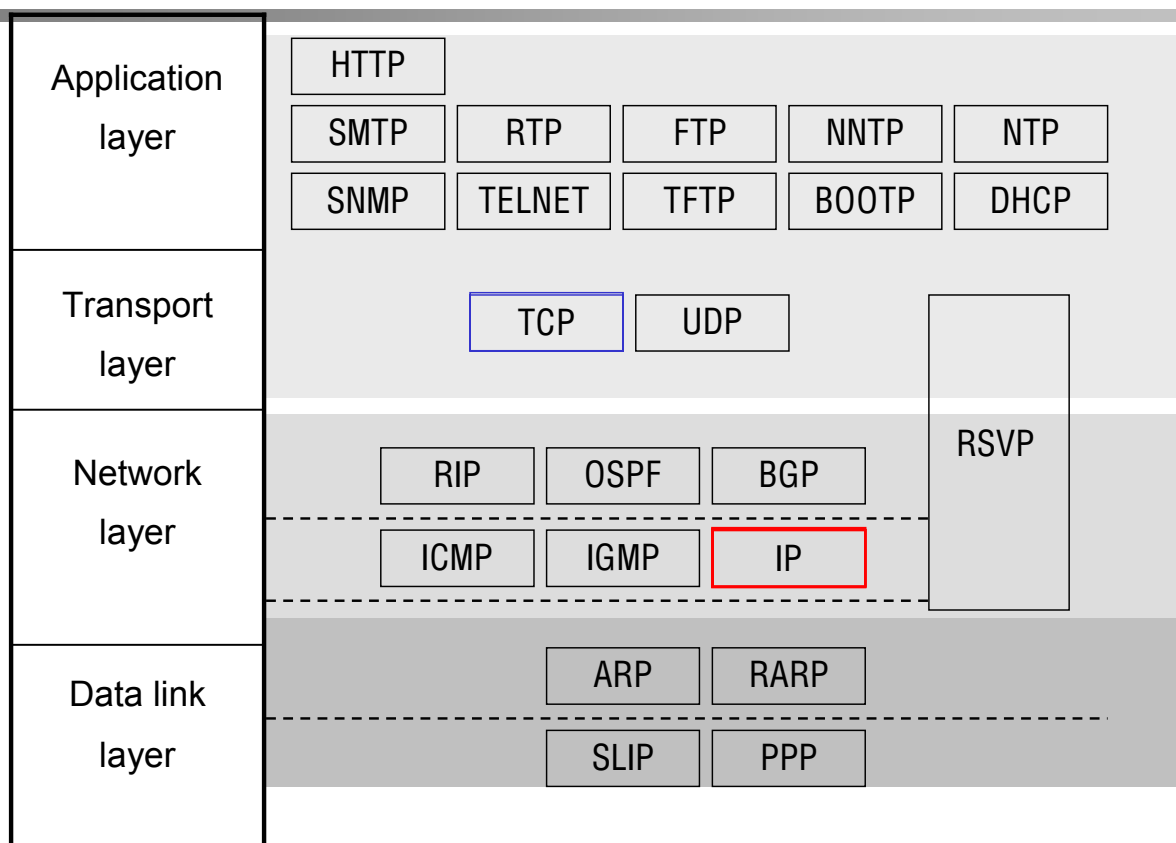
---

- ❑ BS Definition und Aufgaben
- ❑ Auftrags- und Speicherverwaltung
- ❑ Einlagerung, Zuweisung, Ersetzung
- ❑ Kommunikationssysteme
- ❑ **Internet**

# Internet Protocols



## Internet Protocols — Examples

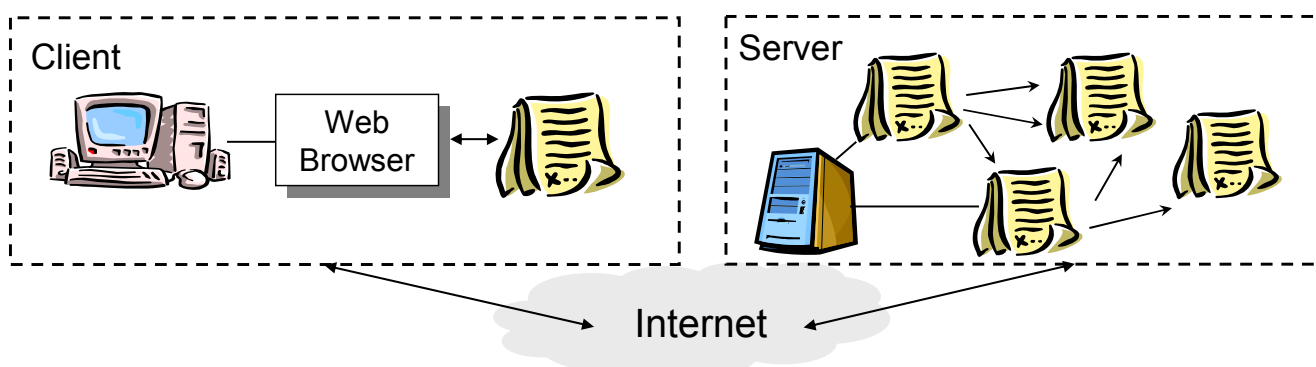


# World Wide Web (WWW)

- ❑ **Goal:**
  - Simple and world-wide exchange of documents
  - Proposal by Tim Berners-Lee (UK) at CERN, CH
- ❑ **Implementation:**
  - First prototype 1990
  - Graphical version (NEXTStep) and command-line based
- ❑ **User interface:**
  - WWW-client Mosaic by Marc Andreessen und Eric Bina (University of Illinois)
  - X-Windows systems
  - Source code publicly available ⇨ fast distribution
  - Marc Andreessen founded Netscape in 1995

## Client/Server-based WWW Architecture

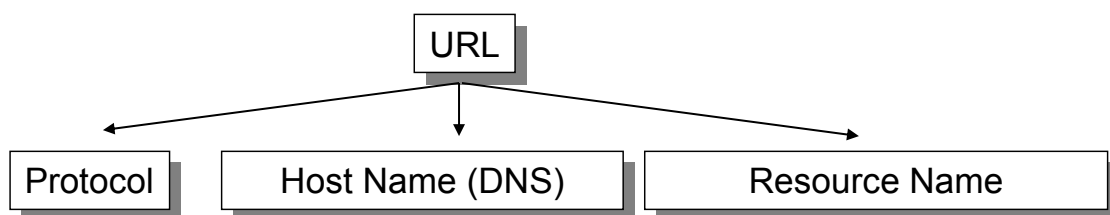
- ❑ **Client/Server-based architecture:**
  - Web browser displaying hypertext documents/hypermedia objects
  - Hyperlinks enable navigation
- ❑ **Solutions required for:**
  - Addressing a web page uniquely
  - Transfer of a web page
  - Description of web page content, especially of hyperlinks





# Addressing a Web Document

- A resource description identifies an object to be accessed within a server:
  - WWW: selected web page
  - FTP: to be transmitted file
  - Mail: Receiver of this message
- Uniform Resource Locator (URL):
  - Determines the resource's location as a name, the protocol to access this resource, and a name of the resource
    - *E.g.*, <http://www.csg.uzh.ch>



## Uniform Resource Locator (URL)

- Compact representation of location and access to Internet resources based on URLs:

**<scheme>: <scheme-specific-part>**

`ftp://<user>:<password>@<host>:<cwd1>..`

`<cwdN>/<name>;type=<typecode>`

`http://<host>:<port>/<path>?<searchpart>`

`https://<host>:<port>/<path>?<searchpart>`

`mailto:<rfc822-addr-spec>`

`nntp://<host>:<port>/`

`<newsgroup-name>/<article-number>`

`telnet://<user>:<password>@<host>:<port>`

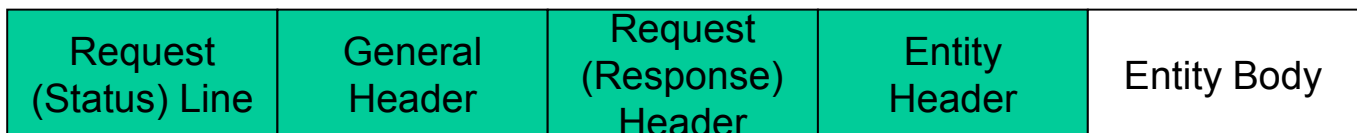
`file://<host>/path`

- Web browser support a large number of protocols

# WWW Application Protocol HTTP

- ❑ HTTP (Hypertext Transport Protocol):
  - Version 0.9/1.0 RFC1945, since January 1997 version 1.1 RFC2068
  - Mainly transfer of Web pages
  
- ❑ Characteristics:
  - ASCII-based application protocol (layer 7)
  - Utilized a reliable TCP connection at default port 80
  - Short-lived connection: HTTP server closes connection after a reply on a request of a client has been issued
  
- ❑ Sample commands:
  - **GET**: Request of a document
  - **HEAD**: Request of header information of this document
  - **POST**: Attachment of data to an existing document
  - **PUT**: Generation of a document

## HTTP Messages



- ❑ Request Line:
  - Message type, requested resource, ...
- ❑ Status Line:
  - State information (error messages, HTTP version, ...)
- ❑ General Header:
  - Cache control, proxy information, ...
- ❑ Request Header:
  - Request- and client information, e.g., acceptable media/coding, ...
- ❑ Response Header:
  - Response information (location, server software, retry information, ...)
- ❑ Entity Header:
  - Information on resources and body  
(Compression, date of modification, language, length)
- ❑ Entity Body:
  - Message body

# Surfin' ... Surfin' ... Surfin' the Net :-)

User of a web browser



Web-Server

