Script **generated by TTT**

Title: Einf_HF (15.07.2013)

Date: Mon Jul 15 14:15:13 CEST 2013

Duration: 95:46 min

Pages: 48

Grundlagen von Rechnernetzen

Fragestellungen des Abschnitts:

- Welche Übertragungsmedien gibt es?
- Was ist das Internet? Wie ist es aufgebaut?
- Wie werden Rechner im Internet addressiert?
- Wie sieht das Kommunikationsreferenzmodell für das Internet aus?

Generated by Targeteam

Referenzmodell

Ein Referenzmodell beschreibt den Aufbau und das Zusammenwirken der Netzwerkprotokolle. Beispiele sind

ISO/OSI Protokollfamilie

TCP/IP Protokollsuite (Referenzmodell des Internet; TCP = Transmission Control Protocol, IP = Internet Protokoll)

Generated by Targeteam

TCP/IP Referenzmodell

Das TCP/IP-Referenzmodell ist auf die Internet-Protokolle zugeschnitten. Es ist der de-facto Standard;

Zur Kommunikation zwischen Rechnern über ein Rechnernetz sind Protokolle notwendig. Ein ***Protokoll*** besteht aus einer Menge von Datenstrukturen (Nachrichtenaufbau) und Konventionen, wie der Ablauf der Kommunikation stattfindet und wie die Informationen jeweils zu interpretieren sind, z.B. Syntax der Nachrichten, Folge und Bedeutung von Nachrichten.

Zur Reduzierung der Komplexität beim Entwurf eines offenen Rechnernetzes wird das Netz in aufeinander aufbauende Protokoll-Schichten unterteilt.

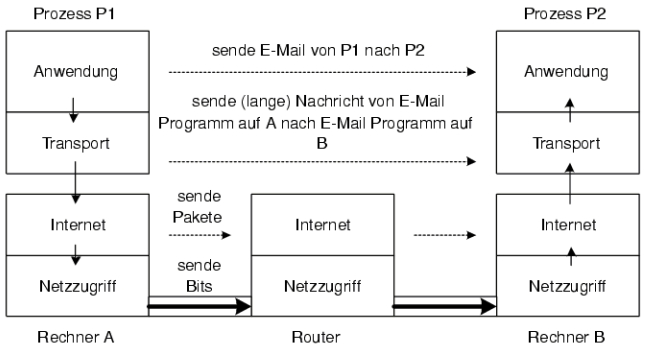
Generated by Targeteam

Prinzipien für Schichtung

Zur Gliederung der Kommunikationsaufgaben werden in Netzwerken funktionale Ebenen, so genannte Schichten (layer), unterschieden.

- jede Schicht repräsentiert eine Abstraktionsebene
- jede Schicht führt eine wohldefinierte Funktion aus
- schmale Schnittstellen zwischen Schichten, um Informationsfluss zu minimieren
- Funktion einer Schicht ist aufgrund von international spezifizierten Standardprotokollen definiert

Aufbau des Internet-Schichtenmodells



Gepunktete Pfeile repräsentieren logische Übertragungen.

Bedeutung der Schichten

Netzzugriff : bietet eine Übertragungsmöglichkeit einzelner Dateneinheiten (Bits) unter bestimmten Zeitbedingungen an; Nachrichtenübertragung zwischen zwei benachbarten Rechnern.

Aufgaben: transparente Übertragung von Bitsequenzen, Berücksichtigung der Eigenschaften der Übertragungsmodi (elektrisch, Lichtwelle), Zusammenfassung von Bitsequenzen zu Rahmen (Frames), Fehlererkennung und Fehlerkorrektur auf Rahmenebene

Internet : fehlerfreie Übermittlung eines Pakets von einem Endrechner, über ein Netz von Routern (Vermittlungsrechnern) hinweg, bis hin zum zweiten Endrechner; beinhaltet Routing und Adressierung; fügt Netzwerk-Header hinzu.

Aufgaben: Zusammenschaltung von Teilstrecken zu einer End-zu-End Verbindung, Wegewahl und Vermittlung, Transporteinheit abhängig von der Vermittlungstechnik (bei Paketvermittlung Verwendung von Paketen)

Transport : fehlerfreier Transport von Nachrichten zwischen zwei kommunizierenden Prozessen auf zwei Endrechnern; bildet die anwendungs-orientierten Schichten auf die netz-abhängigen Schichten ab; fügt Transport-Header hinzu.

Aufgaben: netzunabhängiger Transport von Nachrichten zwischen zwei Endsystemen; passt die vom Anwendungssystem geforderte Übertragungsqualität an die vom darunterliegenden Transportnetz angebotene Übertragungsqualität an

Beispiele: Transmission Control Protocol (TCP: Internet), User Datagram Protocol (UDP: Internet).

Anwendung : es sind verschiedene Applikationsdienst-Elemente festgelegt; deren Auswahl hängt von den ablaufenden Anwendungen ab, z.B. Dateizugriff (FTP), Fernverarbeitung (Telnet), Elektronische Post (SMTP), Name Service, WWW (HTTP).

IP-Adresskonzept

Teil des TCP/IP-Referenzmodells ist die Festlegung der Adressierung von Rechnern im Netz. Dazu erhält jedes Computersystem eine eindeutige "Rufnummer", seine IP-Adresse. Eine IP-Adresse besteht aus vier Byte. Diese werden üblicherweise dezimal durch Punkte getrennt geschrieben, z.B. 131.159.24.30

Die weltweit eindeutige Vergabe von IP-Adressen übernehmen die NICs (Network Information Center). In Deutschland ist das beispielsweise das **DENIC** .

Adressklassen

Neben den IP-Adressen können Computersysteme noch einen oder mehrere einfach zu merkende Namen bekommen, wie z.B. www.in.tum.de. Die möglichen letzten Silben (top level domain) sind global festgelegt, die zweitletzte Silbe wird von jeweils zuständigen Stellen vergeben (für die .de-Domäne beispielsweise DE-NIC)

Durch den Boom des Internet werden die Adressen bereits knapp; deshalb Erweiterung der IP-Adressen auf 128 Bit (IPv6 Adresskonzept).

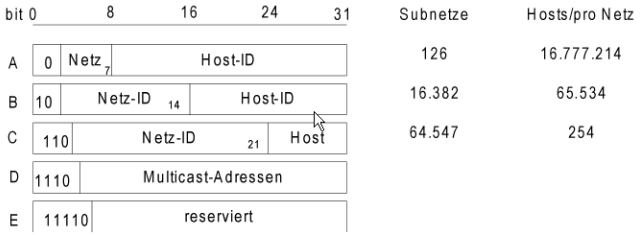
Eine Netzverbindung zwischen 2 entfernten Prozessen wird charakterisiert durch:

verwendetes Protokoll, z.B. TCP.

Adresse des lokalen Prozesses, d.h. IP-Adresse und Portnummer des lokalen Rechners.

Adresse des entfernten Prozesses, d.h. IP-Adresse und Portnummer des entfernten Rechners.

IP-Adressen werden in Blöcke zusammengefasst um die Netzverwaltung zu erleichtern. Die Adresse besteht dazu aus Netz-ID und Host-ID. Nur die Netz-IDs werden zentral vom Network Information Center vergeben. Die Klasse gibt an, wie groß der Byte-Anteil der Netz-ID ist.



Klasse A: Adressbereich 1.0.0.0 - 127.255.255.255

Klasse B: Adressbereich 128.0.0.0 - 191.255.255.255

Klasse C: Adressbereich 192.0.0.0 - 223.255.255.255

Klasse D: Adressbereich 224.0.0.0 - 239.255.255.255 (verwendet durch Videokonferenzsysteme)

Klasse E: Adressbereich 240.0.0.0 - 247.255.255.255

Teil des TCP/IP-Referenzmodells ist die Festlegung der Adressierung von Rechnern im Netz. Dazu erhält jedes Computersystem eine eindeutige "Rufnummer", seine IP-Adresse. Eine IP-Adresse besteht aus vier Byte. Diese werden üblicherweise dezimal durch Punkte getrennt geschrieben, z.B. 131.159.24.30

Die weltweit eindeutige Vergabe von IP-Adressen übernehmen die NICs (Network Information Center). In Deutschland ist das beispielsweise das DENIC .

Adressklassen

Neben den IP-Adressen können Computersysteme noch einen oder mehrere einfach zu merkende Namen bekommen, wie z.B. www.in.tum.de. Die möglichen letzten Silben (top level domain) sind global festgelegt, die zweitletzte Silbe wird von jeweils zuständigen Stellen vergeben (für die .de-Domäne beispielsweise DE-NIC)

Durch den Boom des Internet werden die Adressen bereits knapp; deshalb Erweiterung der IP-Adressen auf 128 Bit (IPv6 Adresskonzept).

Eine Netzverbindung zwischen 2 entfernten Prozessen wird charakterisiert durch:

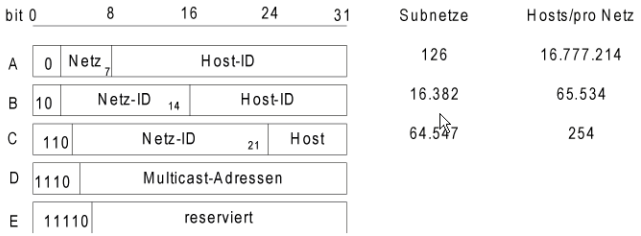
verwendetes Protokoll, z.B. TCP.

Adresse des lokalen Prozesses, d.h. IP-Adresse und Portnummer des lokalen Rechners.

Adresse des entfernten Prozesses, d.h. IP-Adresse und Portnummer des entfernten Rechners.

Adressklassen

IP-Adressen werden in Blöcke zusammengefasst um die Netzverwaltung zu erleichtern. Die Adresse besteht dazu aus Netz-ID und Host-ID. Nur die Netz-IDs werden zentral vom Network Information Center vergeben. Die Klasse gibt an, wie groß der Byte-Anteil der Netz-ID ist.



Klasse A: Adressbereich 1.0.0.0 - 127.255.255.255

Klasse B: Adressbereich 128.0.0.0 - 191.255.255.255

Klasse C: Adressbereich 192.0.0.0 - 223.255.255.255

Klasse D: Adressbereich 224.0.0.0 - 239.255.255.255 (verwendet durch Videokonferenzsysteme)

Klasse E: Adressbereich 240.0.0.0 - 247.255.255.255

Generated by Targeteam

IP-Adresskonzept

Teil des TCP/IP-Referenzmodells ist die Festlegung der Adressierung von Rechnern im Netz. Dazu erhält jedes Computersystem eine eindeutige "Rufnummer", seine IP-Adresse. Eine IP-Adresse besteht aus vier Byte. Diese werden üblicherweise dezimal durch Punkte getrennt geschrieben, z.B. 131.159.24.30

Die weltweit eindeutige Vergabe von IP-Adressen übernehmen die NICs (Network Information Center). In Deutschland ist das beispielsweise das DENIC .

Adressklassen

Neben den IP-Adressen können Computersysteme noch einen oder mehrere einfach zu merkende Namen bekommen, wie z.B. www.in.tum.de. Die möglichen letzten Silben (top level domain) sind global festgelegt, die zweitletzte Silbe wird von jeweils zuständigen Stellen vergeben (für die .de-Domäne beispielsweise DE-NIC)

Durch den Boom des Internet werden die Adressen bereits knapp; deshalb Erweiterung der IP-Adressen auf 128 Bit (IPv6 Adresskonzept).

Eine Netzverbindung zwischen 2 entfernten Prozessen wird charakterisiert durch:

verwendetes Protokoll, z.B. TCP.

Adresse des lokalen Prozesses, d.h. IP-Adresse und Portnummer des lokalen Rechners.

Adresse des entfernten Prozesses, d.h. IP-Adresse und Portnummer des entfernten Rechners.

Generated by Targeteam

Referenzmodell

Ein Referenzmodell beschreibt den Aufbau und das Zusammenwirken der Netzwerkprotokolle. Beispiele sind

ISO/OSI Protokollfamilie

TCP/IP Protokollsuite (Referenzmodell des Internet; TCP = Transmission Control Protocol, IP = Internet Protokoll)

TCP/IP Referenzmodell

IP-Adresskonzept

Sicherung gegen Fehler

Anwendungen von Rechnernetzen

- Verteilte Informationsverarbeitung nutzt die Verbindung mehrerer Rechner durch Rechnernetze zum Aufbau komplexer Systeme, die mehr als einen Rechner einbeziehen. Hier gehen wir näher auf Verteilte Anwendungen im Allgemeinen und auf das momentan in diesem Zusammenhang wichtigste Teilgebiet E-Commerce ein. Weiterhin werden Sicherheitsfragen bei Verteilten Anwendungen behandelt.
- Fragestellungen des Abschnitts:
  - Was versteht man unter dem Client/Server-Modell?
  - Was ist E-Commerce? Welche verteilten Anwendungen spielen hier eine Rolle?
  - Was versteht man unter Verschlüsselung? Welche grundlegenden Verfahren gibt es?
  - Was versteht man unter digitalen Signaturen?

Verteilte Anwendungen

Sicherheit in verteilten Systemen

Verteilte Systeme

Arten der Verteilung

Hardwarekomponenten

ggf. Ziel: räumliche Verteilung

Daten

Partitionierung und/oder Replikation.

Kontrolle/Steuerung

Gegenseitige Abstimmung, keine zentrale Entscheidung. Z.B. verteiltes Betriebssystem.

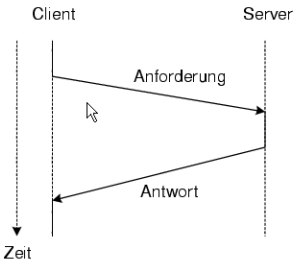
Verarbeitung

Komponenten lösen gemeinsame Aufgabe.

Client-Server-Modell

Möglichkeit zur Strukturierung von (verteilten) Anwendungen: ***Server*** stellen Dienste zur Verfügung, die von (den Servern) vorher unbekannten ***Clients*** in Anspruch genommen werden können.

Client und Server



Client ruft Operation eines Servers auf, erhält ggf. Ergebnis zurück. Während Operation wird Ablauf des Client meist unterbrochen.

Definition: Client

Anwendungsteil, der auf Anforderung einen Dienst von einem Server erhält.

Clients sind meist a-priori beim Server nicht bekannt.

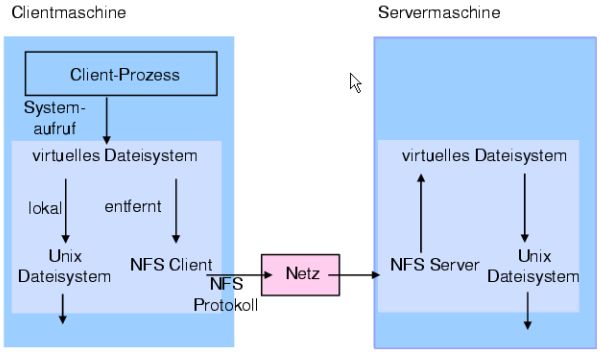
Definition: Server

Subsystem, das bestimmten Dienst für a-priori unbekannte Clients zur Verfügung stellt.

Client und Server kommunizieren über Nachrichten.

Antwort bestätigt Empfang der Anforderung durch den Server.

Beispiel NFS (Network File System)



Beispiel-Services

Datei-Service

Entfernte, zentralisierte Datenspeicherung für Arbeitsplatzrechner.

Beispiel NFS (Network File System)

Namens-Service

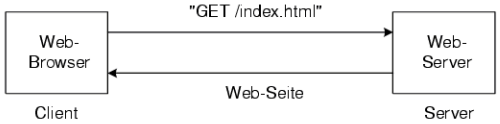
Entfernte, zentralisierte Namensverwaltung für Objekte (Dateien, andere Server, Services, Drucker, Benutzer etc.).

Zeit-Service

Synchronisierte Systemzeit für Rechner.

World Wide Web

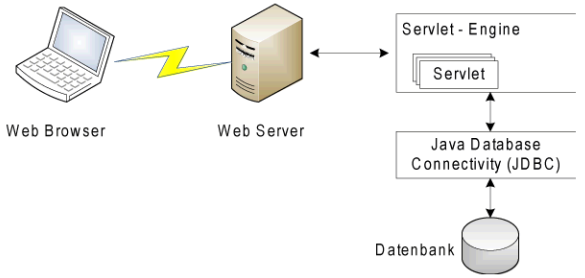
Organisiert nach Client/Server Architektur.



- Zieladresse wird mit Hilfe einer URL angegeben
  - Beispiel: http://www11.in.tum.de:80/lehre/vorlesungen/
  - http:// gibt das Kommunikationsprotokoll für den Zugriff auf Web-Seiten an.
  - www11.in.tum.de gibt den Web-Server an.
  - lehre/vorlesungen/ gibt ein Verzeichnis/Dokument innerhalb des Web-Servers an.
  - Standardport des Web-Servers: 80.
- Weitere Kommunikationsprotokolle
  - https:// Kommunikationsprotokoll für den gesicherten Zugriff auf Web-Seiten.
  - file:// Zugriff auf Dateien am lokalen Rechner.
  - ftp:// Zugriff auf Dateien an einem entfernten Rechner; Nutzung des Filetransfer Dienstes.
  - mailto: verschicken von Emails an die angegebene Adresse.
- Methoden des http Protokolls
  - GET: anfordern einer Ressource (z.B. eine Web-Seite), die mittels einer URL spezifiziert ist.
  - PUT: dient dazu eine Ressource (z. B. eine Web-Seite) unter Angabe der Ziel-URL auf einen Webserver hochzuladen.

Anbindung einer Datenbank

Mit Hilfe von Informationen aus Datenbanken können Inhalte von Web-Seiten dynamisch gestaltet werden; dazu Abruf der DB-Information über Servlets und spezielle Schnittstellen, z.B. Java DB Connectivity (JDBC).



Sicherheit in verteilten Systemen

Ein wichtiger Aspekt eines Rechnersystems besteht darin, dass Daten nicht von Unbefugten abgefragt werden können. Bei nicht-vernetzten Systemen läßt sich dies durch eine Sicherung des physikalischen Zugangs zu den Rechnern und einfache Passwortsysteme garantieren. Bei verteilten Systemen werden zusätzliche Dienste benötigt.

Sicherheitsanforderungen

Garantierung der Identität eines Absenders (Identitätsprüfung, Authentifizierung).

Verhinderung von Mithören (Verschlüsselung).

Sicherstellung der Integrität (Unverfälschtheit) von Nachrichten (Hash-Werte).

Verhinderung des Wiedereinspielens von Nachrichten (Integritätssicherung und Zeitstempel).

Arten von Schadsoftware

Verschlüsselung

Identitätsprüfung

Verschlüsselung

Verschlüsselung bedeutet, dass man Daten für die Übertragung oder Speicherung so codiert, dass sie nur vom beabsichtigten Empfänger wieder in "Klartext" zurückcodiert werden können. Für alle eventuellen "Mithörer" bleiben die Daten eine uninterpretierbare Bitfolge.

Web-Adresse: https://..... ⇒ Daten werden verschlüsselt zwischen Web Browser und Web Server übertragen (z.B. für Online Bestellungen bei Amazon).

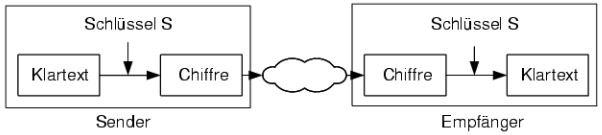
Es gibt zwei grundlegende Verfahrensklassen für die Verschlüsselung von Daten: symmetrische Verschlüsselung und asymmetrische Verschlüsselung.

Symmetrische Verschlüsselung

Asymmetrische Kryptosysteme

Symmetrische Verschlüsselung

Beide Partner (Sender und Empfänger) kennen einen geheimen Schlüssel, der zum Ver- und zum Entschlüsseln verwendet wird. Eine Kenntnis des Schlüssels erlaubt Zugriff auf die Daten.



Austausch des geheimen Schlüssels über einen anderen, sicheren Kommunikationskanal (z.B. persönlich, Brief)

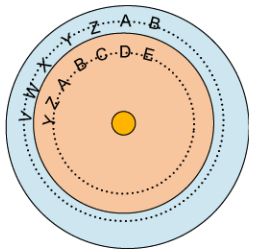
Für jede mögliche Verbindung wird ein eigener Schlüssel benötigt (Falls derselbe Schlüssel für Nachrichten an B und C benutzt wird, könnte C Nachrichten an B lesen).

Beispiele: DES, Tripple DES, IDEA

Erweiterte Caesar-Chiffre

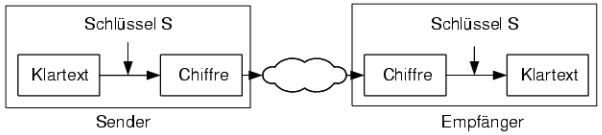
Erweiterte Caesar-Chiffre

- Verschlüsselungsanweisung: Ersetze jeden Buchstaben im Originaltext durch den Buchstaben, der n Stellen im Alphabet weiter hinten (rechts) steht.
- Entschlüsselungsanweisung: Ersetze jeden Buchstaben im verschlüsselten Text durch den Buchstaben, der n Stellen im Alphabet weiter vorne (links) steht.
- Schlüssel: n (natürliche Zahl zwischen 0 und 26)
- Beispiel
  - Originaltext: "VENI VIDI VICI", Schlüssel = 3
  - verschlüsselter Text: "YHQL YLGL YLFL"
- **Cäsar-Scheibe**



Symmetrische Verschlüsselung

Beide Partner (Sender und Empfänger) kennen einen geheimen Schlüssel, der zum Ver- und zum Entschlüsseln verwendet wird. Eine Kenntnis des Schlüssels erlaubt Zugriff auf die Daten.



Austausch des geheimen Schlüssels über einen anderen, sicheren Kommunikationskanal (z.B. persönlich, Brief)

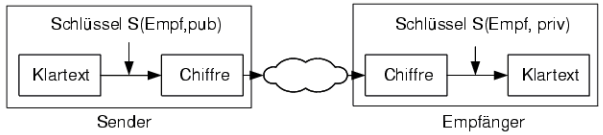
Für jede mögliche Verbindung wird ein eigener Schlüssel benötigt (Falls derselbe Schlüssel für Nachrichten an B und C benutzt wird, könnte C Nachrichten an B lesen).

Beispiele: DES, Tripple DES, IDEA

Erweiterte Caesar-Chiffre

Asymmetrische Kryptosysteme

Zum Ver- und Entschlüsseln wird ein Schlüsselpaar (priv, pub) verwendet. Es existiert ein personenbezogener ***privater Schlüssel*** priv und ein ***öffentlicher Schlüssel*** pub, der allgemein zugänglich und jedem bekannt sein darf. Alle Nachrichten, die mit einem Schlüssel codiert (chiffriert) worden sind, können mit dem jeweils anderen Schlüssel wieder decodiert (dechiffriert) werden.



Für sicheren Datenaustausch wird die Nachricht mit dem öffentlichen Schlüssel des Empfängers S(Empf, pub) verschlüsselt. Dann hat nur der Empfänger selbst mit seinem privaten Schlüssel S(Empf, priv) Zugang zum Inhalt.

Geheimer Schlüssel darf nicht aus dem öffentlichen Schlüssel ableitbar sein.

Je länger der Schlüssel desto sicherer ist das System (Angriffsmöglichkeit durch Ausprobieren aller möglichen Schlüssel).

Beispiel: RSA (Schlüssellänge für Home: 256 Bit, Standard: 512 Bit, Militär: 1024 Bit)

Verschlüsselung mit asymmetrischen Verfahren ist üblicherweise langsamer als mit symmetrischen Verfahren (RSA etwa um den Faktor 1000 langsamer als DES). Deshalb werden die Verfahren in der Praxis oft kombiniert.

Nutzung des asymmetrischen Kryptoverfahrens zum Austausch des geheimen Schlüssels.

Sicherheit in verteilten Systemen

Ein wichtiger Aspekt eines Rechnersystems besteht darin, dass Daten nicht von Unbefugten abgefragt werden können. Bei nicht-vernetzten Systemen läßt sich dies durch eine Sicherung des physikalischen Zugangs zu den Rechnern und einfache Passwortsysteme garantieren. Bei verteilten Systemen werden zusätzliche Dienste benötigt.

Sicherheitsanforderungen

Garantierung der Identität eines Absenders (Identitätsprüfung, Authentifizierung).

Verhinderung von Mithören (Verschlüsselung).

Sicherstellung der Integrität (Unverfälschtheit) von Nachrichten (Hash-Werte).

Verhinderung des Wiedereinspielens von Nachrichten (Integritätssicherung und Zeitstempel).

Arten von Schadsoftware

Verschlüsselung

Identitätsprüfung

Einführung

Allgemeine grundlegende Themen in Informatik

- **Fragestellungen dieses Kapitels**
  - Übersicht über die verschiedenen Aspekte der Informatik
    - Mit welchen Bereichen beschäftigt sich Informatik?
    - Was gehört alles zu einem Computersystem?
  - Darstellung von Information
    - Was ist ein Byte?
    - Information und Nachricht

Was ist Informatik?

Computer

Darstellung von Information

Information und Nachrichten

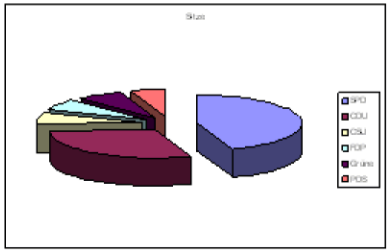
Information an Repräsentation (Darstellung) gebunden. Übertragung zwischen Repräsentationen: Codierung.

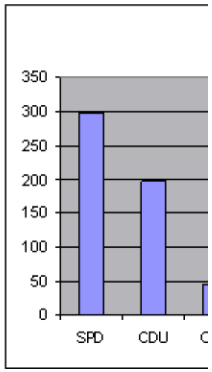
Information-/Datenverarbeitung

Darstellung von Information

Dieselbe Information lässt sich auf verschiedene Arten darstellen, z.B. Sitzverteilung im Bundestag. Im Rechner identische Speicherung

Partei	SPD	CDU	CSU	FDP	Grüne	PDS
Sitze	298	198	47	43	47	36



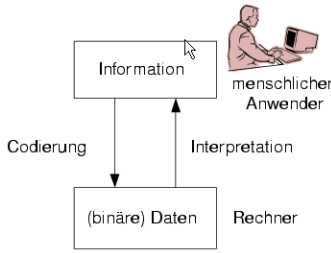


Nachrichten

Einführung in die Informatik für an
Übersicht
Einführung
Was ist Informatik?
Was arbeitet ein Inform
Technische Informatik
Angewandte Informatik
Computer
Darstellung von Informati
Information und Nachri
Information-/Datenv
Nachrichten
Bits und Bytes
Datenbanken und Informatic
Rechnerarchitektur
Systemsoftware
Grundlagen der Programmie
Datenstrukturen und Algorith
Software-Entwicklung
Grundlagen von Rechnerne
Anwendungen von Rechner
Zusammenfassung

Information-/Datenverarbeitung

Geeignete Darstellung im Rechner nötig.



Menschlicher Anwender: Informationen. Im Rechner: (binäre) Daten. Benutzerebene: "Informationsverarbeitung", Rechnerebene: "Datenverarbeitung".

Daten: als Folge von Bits dargestellt.

Datenbankmanagementsystem

Das Datenbankmanagementsystem (DBMS) ist die Gesamtheit aller Programme für den Umgang mit den Daten. Es ist verantwortlich für

die sichere und einheitliche Verwaltung persistenter (langlebiger) Daten,

den Datenaustausch zwischen Datenbank und Anwendungsprogrammen,

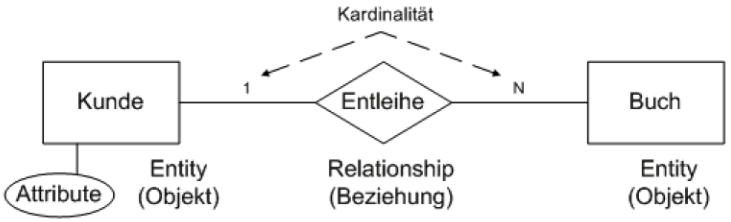
die Verhinderung von unkontrollierten Zugriffen auf den Datenbestand und

die effiziente Zugriffsmöglichkeit auf die in der Regel sehr großen Datenbestände.

Einführung in die Informatik für an
Übersicht
Einführung
Datenbanken und Informatic
Dateisysteme
Datenbanksysteme
Logischer Aufbau
Datenbankmanageme
Beispiele aus der Prax
Anforderungen an ein
Datenbankentwurf
Relationale Datenbanksy
WWW - Informationssyst
Rechnerarchitektur
Systemsoftware
Grundlagen der Programmie
Datenstrukturen und Algorith
Software-Entwicklung
Grundlagen von Rechnerne
Anwendungen von Rechner
Zusammenfassung

ER-Modell

Entity-Relationship-Modell eignet sich zur Darstellung des Datenbankschemas.



Graphisches Hilfsmittel zur semantischen Modellierung eines Anwendungsgebietes, d.h. zum Entwurf einer Datenbank, unabhängig vom konkreten DBS.

Grundidee: Reale Welt (Mini-Welt) läßt sich durch Objekte und Beziehungen zwischen Objekten beschreiben (Objekte: Entities, Beziehungen: Relationships).

Gleichartige Entities (Objektinstanzen) werden zu Entity-Typen (vergleichbar Klassen) bzw. Relationships zu Relationship-Typen zusammengefasst.

Beispiele: Entity-Typ: "Bibliotheksbenutzer", "Buch", und Relationship-Typ: "entleiht".

Attribute bei Entities und Relationships, z.B. "Name" bei Entity "Bibliotheksbenutzer", "Entleihdatum" bei Relationship "leiht aus".

Entity-Typ

Relationship-Typ

Relationship-Typ

Ein Relationship-Typ umfasst die Menge gleichartiger Relationships. Ein Relationship-Typ R stellt die Beziehung zwischen Entity-Typen E1 und E2 her, d.h. R ⊆ E1 * E2.

Kardinalität von Relationship-Typen zur Spezifikation der Art der Beziehung zwischen Entities.

Man kann verschiedene Kardinalitätsarten unterscheiden

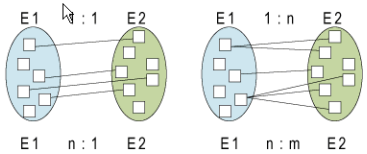
1:1 , ein Entity aus E1 kann mit höchstens einem Entity aus E2 über R in Beziehung stehen kann und umgekehrt.

n:1 , ein Entity aus E1 kann mit höchstens einem Entity aus E2 , aber ein Entity aus E2 mit beliebig vielen Entities aus E1 über R in Beziehung stehen.

1:n , ein Entity aus E1 kann mit beliebig vielen Entities aus E2 , aber ein Entity aus E2 mit höchstens einem Entity aus E1 über R in Beziehung stehen.

Beispiel: 1 Kunde kann n Bücher ausleihen; 1 Buch kann nur von einem Kunden ausgeliehen werden.

n:m , ein Entity aus E1 kann mit beliebig vielen Entities aus E2 über R in Beziehung stehen und umgekehrt.



Inhalt

Einführung in die Informatik für an
Übersicht
Einführung
Datenbanken und Informatic
Dateisysteme
Datenbanksysteme
Logischer Aufbau
Datenbankmanageme
Beispiele aus der Prax
Anforderungen an ein
Datenbankentwurf
Relationale Datenbanksy
WWW - Informationssyste
Rechnerarchitektur
Systemsoftware
Grundlagen der Programmie
Datenstrukturen und Algorith
Software-Entwicklung
Grundlagen von Rechnerne
Anwendungen von Rechner
Zusammenfassung

Relationale Datenbanksysteme

Entity-Relationship-Diagramm: logisches Modell einer Datenbank. Für Implementierung in DBS: physikalisches Modell nötig.

Beispiel: relationales Datenbankmodell.

Relationales Modell

Tabellendarstellung

Normalisierung

Umsetzung des ER-Modells

Sichten

Verwendung von Sichten zur Auswahl einer Teilmenge der Tabellendaten.

Beispiel für eine Sicht: Vorname und Nachname von Kunden, die in München wohnen.

Views sind damit nichts anderes als benannte Such-Abfragen (z.B. SQL-Anfragen in MS ACCESS mit Namen).

Abfragesprache SQL

Beispielsysteme

mySQL (Open Source), Microsoft Access, Microsoft SQL Server, Oracle, DB2 (IBM), Sybase, Informix

Microsoft Access

Inhalt

Einführung in die Informatik für an
Übersicht
Einführung
Datenbanken und Informatic
Dateisysteme
Datenbanksysteme
Logischer Aufbau
Datenbankmanageme
Beispiele aus der Prax
Anforderungen an ein
Datenbankentwurf
Relationale Datenbanksy
WWW - Informationssyste
Rechnerarchitektur
Systemsoftware
Grundlagen der Programmie
Datenstrukturen und Algorith
Software-Entwicklung
Grundlagen von Rechnerne
Anwendungen von Rechner
Zusammenfassung

Relationale Datenbanksysteme

Entity-Relationship-Diagramm: logisches Modell einer Datenbank. Für Implementierung in DBS: physikalisches Modell nötig.

Beispiel: relationales Datenbankmodell.

Relationales Modell

Tabellendarstellung

Normalisierung

Umsetzung des ER-Modells

Sichten

Verwendung von Sichten zur Auswahl einer Teilmenge der Tabellendaten.

Beispiel für eine Sicht: Vorname und Nachname von Kunden, die in München wohnen.

Views sind damit nichts anderes als benannte Such-Abfragen (z.B. SQL-Anfragen in MS ACCESS mit Namen).

Abfragesprache SQL

Beispielsysteme

mySQL (Open Source), Microsoft Access, Microsoft SQL Server, Oracle, DB2 (IBM), Sybase, Informix

Microsoft Access

Generated by Targeteam

Inhalt

Einführung in die Informatik für an
Übersicht
Einführung
Datenbanken und Informatic
Dateisysteme
Datenbanksysteme
Logischer Aufbau
Datenbankmanageme
Beispiele aus der Prax
Anforderungen an ein
Datenbankentwurf
Relationale Datenbanksy
WWW - Informationssyste
Rechnerarchitektur
Systemsoftware
Grundlagen der Programmie
Datenstrukturen und Algorith
Software-Entwicklung
Grundlagen von Rechnerne
Anwendungen von Rechner
Zusammenfassung

Umsetzung des ER-Modells

Jeder Entity-Typ wird zu einer eigenen Tabelle

die Attribute des Entity-Typs werden zu den Spalten.

ein Attribut (oder eine Kombination von Attribute) wird als ***Primärschlüssel*** definiert.

eine Tabellenzeile repräsentiert eine Instanz des Entity-Typs (Objektinstanz, Entität)

Umsetzung von Relationship-Typ: 1:1

Umsetzung von Relationship-Typ: 1:n

Umsetzung von Relationship-Typ: n:m

Entity-Relationship-Diagramm: logisches Modell einer Datenbank. Für Implementierung in DBS: physikalisches Modell nötig.

Beispiel: relationales Datenbankmodell.

Relationales Modell

Tabellendarstellung

Normalisierung

Umsetzung des ER-Modells

Sichten

Verwendung von Sichten zur Auswahl einer Teilmenge der Tabellendaten.

Beispiel für eine Sicht: Vorname und Nachname von Kunden, die in München wohnen.

Views sind damit nichts anderes als benannte Such-Abfragen (z.B. SQL-Anfragen in MS ACCESS mit Namen).

Abfragesprache SQL

Beispielsysteme

mySQL (Open Source), Microsoft Access, Microsoft SQL Server, Oracle, DB2 (IBM), Sybase, Informix

Microsoft Access

"Structured Query Language": Suche und Ändern von Tabelleneinträgen; seit 1989 international genormt; für fast alle relationalen Datenbanken verfügbar; Abfrage liefert als Ergebnis alle gefundenen Lösungen (d.h. mengenorientiert).

Elementare Operationen bei Abfragen

Relationen

INSERT (Einfügen)

INSERT INTO Entleihe VALUES (300, 100, '01/12/97')

UPDATE (Aktualisierung)

UPDATE Kunde SET PLZ = "80330" WHERE Strasse = 'Arcisstrasse'

SELECT (Abfrage)

Finde alle Kunden, die in der Arcisstraße in München wohnen:

SELECT Vorname, Nachname, Straße FROM Kunde WHERE Ort = 'München' AND Straße = 'Arcisstraße' ORDER BY Nachname

Aggregatfunktionen und Gruppierung

- Fragestellungen des Abschnitts:
  - Aus welchen (Hardware-)Elementen setzt sich ein Rechner zusammen?
  - Wie kommunizieren die einzelnen Komponenten eines Rechners?
  - Wie sieht die Schnittstelle zwischen Hardware und Software aus (d.h. Maschinenbefehle)?
  - Wie werden Zahlen, Text, Bilder, und Töne intern dargestellt?

Aufbau eines Rechners

Maschinenbefehle

Befehlszyklus

Interndarstellung von Information

Ohne Programme ist Hardware nicht arbeitsfähig. Zwei Klassen: Anwendungsprogramme, Systemprogramme (insbes.: Betriebssystem; elementare Dienste).

- Fragestellungen des Abschnitts:
  - Was sind die Aufgaben eines Betriebssystems?
  - Welche Dienste bietet ein Betriebssystem zur Arbeit mit Massenspeichern (Festplatte)?
  - Was sind Prozesse (im Gegensatz zu Programmen)?
  - Wie wird der Arbeitsspeicher verwaltet?

Einführung

Dateiverwaltung

Prozessorverwaltung / Prozessorzuteilung

Wesentliche Komponenten eines Betriebssystems

Dateiverwaltung (externer Speicher): Transparente Verfügbarmachung von externen Speichermedien; auf Festplatte werden Informationen in Dateien gespeichert.

Prozessverwaltung : Aufteilung des Prozessors auf mehrere gleichzeitig laufende Aufgaben.

Speicherverwaltung (interner Speicher): Aufteilung des Arbeitsspeichers auf mehrere gleichzeitig laufende Aufgaben (Programme).

Ein/Ausgabegeräte-Verwaltung : Abstraktion der Geräteeigenschaften und Behandlung konkurrierender Zugriffe.

Prozessorverwaltung / Prozessorzuteilung

Aufgabe der Prozessverwaltung ist die Koordination mehrerer gleichzeitig laufender Programme.

Prozesse

Prozessorzuteilung

Prozess-Unterbrechungen (interrupt)

Prozesse

Prozess ("process, task"): Ablauf eines Programms, vom Betriebssystem verwaltet. Bestimmt durch Befehle und Daten des Programms.

Eigenschaften

"Programm in Ausführung" (Folge von Maschinenbefehlen eines Programms das gerade durch den Prozessor (CPU) ausgeführt wird).

Prozess hat einen Zustand und wird durch einen Kontext beschrieben (Prozesskontrollblock).

Verwaltungseinheit des Betriebssystems. Hierarchische Beziehung durch Start von Prozessen durch andere Prozesse. Asynchron / Synchron.

Operiert in einem "eigenen" Teil des Arbeitsspeichers (Prozessadressraum).

Konkurriert mit anderen Prozessen um Betriebsmittel (z.B. Arbeitsspeicher, Prozessor, Zugriff auf Festplatte).

Kommuniziert mit anderen Prozessen über Nachrichten.

Unterscheidung zwischen

Benutzerprozessen : vom Benutzer gestartete Programme.

Systemprozessen : erbringen System-Dienste des Betriebsystems, beispielsweise Drucken.

Zustände eines Prozesses

Grundlagen der Programmierung

"Kunst des Programmierens". Grundlagen zu Datenstrukturen, Programmkonstrukte, Strukturierung von Programmen, objekt-orientierte Programmierung.

- Fragestellungen des Abschnitts:
  - Was ist ein Algorithmus?
  - Welche elementaren Datenstrukturen gibt es?
  - Was sind die grundlegenden Konstrukte einer Programmiersprache?
  - Was ist unter Objekt-orientierter Programmierung zu verstehen?
  - Was versteht man unter Modularisierung und Rekursion?

Einführung

Algorithmus

Datentypen und Ausdrücke

Programmkonstrukte

Objektorientierte Programmierung

Modularisierung von Programmen

Rekursion

"Kunst des Programmierens". Grundlagen zu Datenstrukturen, Programmkonstrukte, Strukturierung von Programmen, objekt-orientierte Programmierung.

- Fragestellungen des Abschnitts:
  - Was ist ein Algorithmus?
  - Welche elementaren Datenstrukturen gibt es?
  - Was sind die grundlegenden Konstrukte einer Programmiersprache?
  - Was ist unter Objekt-orientierter Programmierung zu verstehen?
  - Was versteht man unter Modularisierung und Rekursion?

Einführung

Algorithmus

Datentypen und Ausdrücke

Programmkonstrukte

Objektorientierte Programmierung

Modularisierung von Programmen

Rekursion

Softwaresystem realisiert durch Menge von Objekten. Gegensatz prozedurale Programmierung: Anweisungen im Vordergrund.

Objektorientiertes Programmieren: Daten im Vordergrund. In Objekten zusammengefasst ("Verkapselung"). Funktionen lokal bei Objekten definiert.

Die Funktionen werden Methoden genannt.

Objekt - Klasse

Erzeugen eines Objekts

Vererbung

Spezielle Form der Modularisierung. Zu definierendes Modul wird in seiner Definition selbst benutzt.

"natürlichere" Darstellung bei bestimmten Algorithmen und Datenstrukturen.

Beispiel Summe

Definiertes Ende einer rekursiven Schachtelung.

```
if (n > 0)
   summe = summe(n-1) + n;
else summe = 0; /* definiertes Ende, wenn n <= 0 */
```

Beispiel 'Fibonacci-Zahlen'

Beispiel 'Größter gemeinsamer Teiler'

Beispiel 'Türme von Hanoi'

Schadsoftware ("Malware") sind Programme, die Aktionen ausführen, die unerwünscht und meist schädlich sind.

Computerviren: sich selbst verbreitende Programme, die sich in anderen Programmen einschleusen.

Computerwurm: vervielfältigt sich selbst, wenn die Software, in die es eingebettet ist, ausgeführt wird.

Trojanisches Pferd: Software, die vortäuscht eine nützliche Anwendung zu sein, und somit dazu verführt, sie auszuführen.

SPAM: unerwünschte Nachrichten, die dem Empfänger unverlangt zugestellt werden.

Spyware: forscht den Rechner und das Verhalten des jeweiligen Nutzers ohne dessen Wissen aus und sendet die Daten an den Hersteller der Spyware.

Phishing: Versuche, um an geheime Daten eines Nutzers zu gelangen.

Adware: bei normaler Installation oder beim Herunterladen nützlicher Software wird Reklamesoftware installiert.

Dialer: bauen heimlich im Hintergrund über das Telefonnetz eine Wählverbindung zu teueren 0190 bzw. 0900-Nummern auf.

Ein wichtiger Aspekt eines Rechnersystems besteht darin, dass Daten nicht von Unbefugten abgefragt werden können. Bei nicht-vernetzten Systemen läßt sich dies durch eine Sicherung des physikalischen Zugangs zu den Rechnern und einfache Passwortsysteme garantieren. Bei verteilten Systemen werden zusätzliche Dienste benötigt.

Sicherheitsanforderungen

Garantierung der Identität eines Absenders (Identitätsprüfung, Authentifizierung).

Verhinderung von Mithören (Verschlüsselung).

Sicherstellung der Integrität (Unverfälschtheit) von Nachrichten (Hash-Werte).

Verhinderung des Wiedereinspielens von Nachrichten (Integritätssicherung und Zeitstempel).

Arten von Schadsoftware

Verschlüsselung

Identitätsprüfung