

## H Rechnerkommunikation (TCP/IP)

### H.1 Überblick

- Ende der 70er Jahre in den USA Entwicklung neuer Netzwerkprotokolle, gefördert durch **DARPA** (*Defense Advanced Research Project Agency*)
- daraus entstandene Protokollfamilie: **TCP/IP** (nach den Namen der zentralen Protokolle - *Transmission Control Protocol* und *Internet Protocol*)
- ab 1983 Standardprotokoll im ARPANet
- in Berkeley Implementierung der Protokolle für UNIX (4.1c bsd)
- ab 1985/86 Einsatz von TCP/IP in den Netzen der **NSF** (*National Science Foundation*), **NASA** und anderer Institutionen
- in den 90er Jahren Übergang zu kommerziellen Netz Providern (ISPs) (z. B. USA: MCI, Sprint, AT&T, D: EUnet, XLink, MAZ)

AKBP I

Ausgewählte Kapitel der praktischen Betriebsprogrammierung

© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999

H-Net.doc 1999-01-27 09.05

H.1

Reproduktion jeder Art oder Verwendung dieser Unterrichts-, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

### H.1 Überblick (2)

- Implementierungen der Protokolle sind heute für alle in für Netzbetrieb relevanten Betriebssysteme verfügbar
- die TCP/IP-Protokolle sind unabhängig von der verwendeten Netzwerktechnologie - verwendbar sind z. B.
  - ◆ Ethernet, Token Ring, FDDI, ATM
  - ◆ serielle Leitungen, Modemverbindungen (PPP), ISDN
- die TCP/IP-Protokolle sind keine Standards im Sinn internationaler Standardisierungsgremien (ISO, CCITT), aber **de facto Standards**
- Arbeitsberichte, Protokollvorschläge und -standards werden in einer Serie von technischen Berichten veröffentlicht, den **Requests for Comments - RFCs**

### H.1 Überblick (2)

- IP ermöglicht den Verbund von Netzen, das entstandene Gesamtnetz wird als **Internet** bezeichnet
- Größe: 1987: ca. 20.000 Rechner, mehrere hundert Netze  
1994: über 3 Mio. Rechner in 61 Ländern  
Ende 98: >1.6 Mio. in D, > 8 Mio. in EU, ? weltweit (sichtbare Rechner), exponentieller Anstieg
- seit Ende der 80er Jahre starke Ausbreitung des **Internet** in Europa (NORDUnet, XLINK, SURFnet, ...)
- in Deutschland zuerst regionale Cluster (Bayern und Baden Württemberg), seit Ende 1989 bundesweites Internet mit dem WIN (Wissenschaftsnetz) als Transportnetz

AKBP I

Ausgewählte Kapitel der praktischen Betriebsprogrammierung

© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999

H-Net.doc 1999-01-27 09.05

H.2

Reproduktion jeder Art oder Verwendung dieser Unterrichts-, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

### H.2 Einordnung in das ISO/OSI Referenzmodell

7	Anwendungsschicht	ftp, telnet, rlogin, rsh, ...
6	Darstellungsschicht	-
5	Sitzungsschicht	
4	Transportschicht	TCP, UDP
3	Netzwerkschicht	IP
2	Verbindungsschicht	Ethernet
1	physikalische Schicht	

AKBP I

Ausgewählte Kapitel der praktischen Betriebsprogrammierung

© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999

H-Net.doc 1999-01-27 09.05

H.4

Reproduktion jeder Art oder Verwendung dieser Unterrichts-, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

### H.3 Internet Protocol - IP

- Netzwerkprotokoll zur Bildung eines virtuellen Netzwerkes auf der Basis mehrerer physischer Netze
- definiert Format der Datengrundeinheit - IP-Datagramm
- enthält Regeln, wie Pakete verarbeitet und Fehler behandelt werden (Fragmentierung, maximale Lebenszeit, ...)
- unzuverlässige Datenübertragung
- Routing-Konzepte

AKBP I

Ausgewählte Kapitel der praktischen Betriebsprogrammierung

© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999

H-Inet.doc 1999-01-27 09.05

Reproduktion jeder Art oder Verwendung dieser Unterrichtsfolie außer zu Lehrzwecken an der Universität Erlangen-Nürnberg bedarf der Zustimmung des Autors.

H.5

1

### Adressierung (2)

H.3 Internet Protocol - IP

#### Probleme des Adressierungsschemas

- Mehrere physische Netze in einem Class A oder B Netz
  - ◆ **Subnetting:** Der Rechneranteil der Adresse kann durch sog. **Subnetmask** in einen Unternetz- und den eigentlichen Rechneranteil unterteilt werden (Bit-Maske)
  - ◆ Nach "außen" wird nur das Class A/B-Netz bekanntgegeben, intern hat man mehrere Netze
- Nur 127 Class A- und ca. 17.000 Class B-Netze, aber über 2 Mio Class-C-Netze → **ROADS-Problem**
  - ◆ Mehrere Class-C-Netze werden zusammengefaßt (**Supernetting**)
    - statt einem Class-B-Netz z. B. 256 Class-C-Netze
  - ◆ Beim Routing wird der Netzblock gemeinsam behandelt

AKBP I

Ausgewählte Kapitel der praktischen Betriebsprogrammierung

© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999

H-Inet.doc 1999-01-27 09.05

Reproduktion jeder Art oder Verwendung dieser Unterrichtsfolie außer zu Lehrzwecken an der Universität Erlangen-Nürnberg bedarf der Zustimmung des Autors.

H.7

### 1 Adressierung

H.3 Internet Protocol - IP

#### Grundprinzipien

- einheitliche Adressierung, unabhängig von Netzwerk-Technologie (Ethernet, Token-Ring)
  - ↳ virtuelles Netz: **Internet**
- Netzwerkeinheitlich (über Router) - transparent für Benutzer
- Aufbau einer IP-Adresse:
  - ◆ 4 Byte: **a.b.c.d**, unterteilt in Netzwerk- und Rechneradresse
    - a < 128:** **a** = Netzwerk, **b.c.d** = Rechner (*Class A Net*)  
große Organisationen
    - 127 < a < 192:** **a.b** = Netzwerk, **c.d** = Rechner (*Class B Net*)  
Organisationen mittlerer Größe
    - a > 191:** **a.b.c** = Netzwerk, **d** = Rechner (*Class C Net*)  
kleinere Organisationen

AKBP I

Ausgewählte Kapitel der praktischen Betriebsprogrammierung

© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999

H-Inet.doc 1999-01-27 09.05

Reproduktion jeder Art oder Verwendung dieser Unterrichtsfolie außer zu Lehrzwecken an der Universität Erlangen-Nürnberg bedarf der Zustimmung des Autors.

H.6

### 1 Adressierung (2)

H.3 Internet Protocol - IP

#### IP-Adressen an der Uni Erlangen:

- ◆ **REVUE-Netz**
  - Class B Netz 131.188
  - ◆ Subnetmask: 0xFFFFFFF00  
(teilweise auch 0xFFFFFC0)
  - ◆ Beispiele:
    - Informatik Hauptnetz: 131.188.2
    - Inf.-CIP Subnetz: 131.188.30
    - Inf. 4 Subnetz: 131.188.34
  - faui01: 131.188.2.1,  
131.188.30.1
  - faui04a: 131.188.30.240
- ◆ zusätzlich Medizin-internes Netz  
**UNI\_ERL\_MED**
  - Class B Netz 141.69
- ↳ Rechner mit mehreren IP-Adressen unerwünscht — besser spezielle Geräte (Router) für die Verbindung zwischen

AKBP I

Ausgewählte Kapitel der praktischen Betriebsprogrammierung

© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999

H-Inet.doc 1999-01-27 09.05

Reproduktion jeder Art oder Verwendung dieser Unterrichtsfolie außer zu Lehrzwecken an der Universität Erlangen-Nürnberg bedarf der Zustimmung des Autors.

H.8

## 2 Routing

[H.3 Internet Protocol - IP](#)

### Motivation

- **Routing** bezeichnet die Aufgabe, einen Weg zu finden, über den IP-Pakete geschickt werden
- Prinzipiell gibt es zwei Arten des Routings:
  - ◆ **Direktes Routing:** Ein Paket wird direkt von einem Rechner zu einem anderen geschickt (**auf einem physischen Netz**, Grundlage für alle Internet-Kommunikation)
  - ◆ **Indirektes Routing:** Der Zielrechner ist **nicht auf demselben physischen Netz**, so daß das Paket über (einen) Router geschickt werden muß

AKBP I

Ausgewählte Kapitel der praktischen Betriebsprogrammierung

© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999

H-Net.doc 1999-01-27 09.05

H.9

Reproduktion jeder Art oder Verwendung dieser Unterrichts- oder Lehrzwecke an der Universität Erlangen-Nürnberg bedarf der Zustimmung des Autors.

H.10

Reproduktion jeder Art oder Verwendung dieser Unterrichts- oder Lehrzwecke an der Universität Erlangen-Nürnberg bedarf der Zustimmung des Autors.

## 2 Routing (3)

[H.3 Internet Protocol - IP](#)

### Indirektes Routing

- Beim indirekten Routing muß der Absender das verpackte Datagramm **an einen Router schicken**, der selbst wiederum das Datagramm zum Ziel-Netzwerk weiterleiten muß
- **Router** im Internet bilden eine **kooperative, verbundene Struktur**
- **Datagramme werden von Router zu Router weitergereicht**, bis sie das Ziel-Netzwerk erreicht haben.
- Da sowohl der Absender-Rechner als auch ein Router wissen müssen, welchen Router sie für ein bestimmtes Ziel-Netzwerk benutzen müssen, brauchen sie **Routing-Informationen**.
- Routing-Informationen werden in Routing-Tabellen gehalten

AKBP I

Ausgewählte Kapitel der praktischen Betriebsprogrammierung

© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999

H-Net.doc 1999-01-27 09.05

H.11

Reproduktion jeder Art oder Verwendung dieser Unterrichts- oder Lehrzwecke an der Universität Erlangen-Nürnberg bedarf der Zustimmung des Autors.

## 2 Routing (2)

[H.3 Internet Protocol - IP](#)

### Direktes Routing

- Der Absender verpackt das IP-Datagramm in ein Paket des physischen Netzes (Encapsulation)
- Die Ziel-IP-Adresse wird in eine Adresse des physischen Netzes abgebildet
- Das Paket wird mittels der Netzwerk-Hardware übertragen
- häufigstes Verfahren: **ARP (address resolution protocol)** (auf Broadcast-Basis)

AKBP I

Ausgewählte Kapitel der praktischen Betriebsprogrammierung

© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999

H-Net.doc 1999-01-27 09.05

H.10

Reproduktion jeder Art oder Verwendung dieser Unterrichts- oder Lehrzwecke an der Universität Erlangen-Nürnberg bedarf der Zustimmung des Autors.

## 2 Routing (4)

[H.3 Internet Protocol - IP](#)

### Routing-Informationen

- Um die **Routing-Tabellen so kompakt** wie möglich zu halten, beziehen die Routing-Informationen sich (im allgemeinen) **nicht auf Ziel-Adressen, sondern auf Zielnetzwerke**
- IP-Software schickt das jeweilige Datagramm **entsprechend dem Netzwerk-Teil der IP-Adresse** an einen bestimmten Router
- Routing-Tabelleneinträge sind normalerweise ein **Tupel (Netw, Router)**
- Um den Unfang der Routing-Tabellen zu begrenzen, kann ein **Default Router** angegeben werden, der für alle nicht explizit aufgeführten Netze benutzt wird.

AKBP I

Ausgewählte Kapitel der praktischen Betriebsprogrammierung

© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999

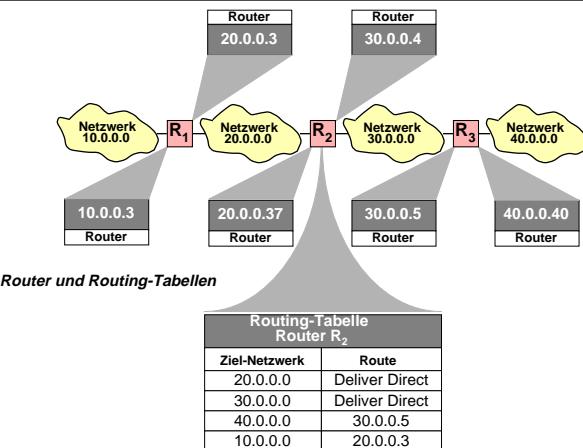
H-Net.doc 1999-01-27 09.05

H.12

Reproduktion jeder Art oder Verwendung dieser Unterrichts- oder Lehrzwecke an der Universität Erlangen-Nürnberg bedarf der Zustimmung des Autors.

## 2 Routing (5)

### H.3 Internet Protocol - IP



AKBP I

Ausgewählte Kapitel der praktischen Betriebsprogrammierung

© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999

H-Net.doc 1999-01-27 09.05

H.13

Reproduktion jeder Art oder Verwendung dieser Unterrichtsfolie außer zu Lehrzwecken an der Universität Erlangen-Nürnberg bedarf der Zustimmung des Autors.

## 2 Routing (7)

### H.3 Internet Protocol - IP

#### Routing-Protokolle

- **RIP** — Vektordistanz-Algorithmus, Distanz-Metrik = *hop-count*
  - ◆ jeder aktive RIP-Rechner überträgt seine Routing-Tabelle regelmäßig per Broadcast
  - ◆ jeder RIP-Rechner übernimmt Routing-Information, falls er nicht schon mindestens eine gleichwertige Route kennt
  - ◆ Routing-Information von RIP wird nach einer *timeout*-Phase ungültig, wenn sie nicht erneuert wurde
- **OSPF** — Shortest Path First-Algorithmus
  - ◆ es werden Informationen über die Netzaanschlüsse jedes Knotens ausgetauscht, jeder OSPF-Rechner erstellt für sich daraus den Verbindungsgraphen des Netzes und routet danach
  - ◆ viele administrative Einflußmöglichkeiten (*load balancing*, etc.)

AKBP I

Ausgewählte Kapitel der praktischen Betriebsprogrammierung

© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999

H-Net.doc 1999-01-27 09.05

H.15

Reproduktion jeder Art oder Verwendung dieser Unterrichtsfolie außer zu Lehrzwecken an der Universität Erlangen-Nürnberg bedarf der Zustimmung des Autors.

## 2 Routing (6)

### H.3 Internet Protocol - IP

#### Routing-Protokolle

- **Autonomes System (AS)**: Menge von Netzen und Routern einer Administrationseinheit (z. B. alle Netze im deutschen Wissenschaftsnetz WiN)
- Autonome Systeme tauschen Informationen über Netze und zugehörige AS-Pfade untereinander aus (*Exterior Gateway Protocol* — EGP, speziell *Border Gateway Protocol* — BGP)
- Innerhalb eines AS werden Routing-Informationen mit *Interior Gateway Protocols* ausgetauscht (vor allem **RIP** und **OSPF**)

AKBP I

Ausgewählte Kapitel der praktischen Betriebsprogrammierung

© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999

H-Net.doc 1999-01-27 09.05

H.14

Reproduktion jeder Art oder Verwendung dieser Unterrichtsfolie außer zu Lehrzwecken an der Universität Erlangen-Nürnberg bedarf der Zustimmung des Autors.

## H.4 User Datagram Protocol - UDP

### 1 Motivation

- bei IP definiert eine Adresse einen Rechner
- keine Möglichkeit, einen bestimmten Benutzer oder Prozeß (Dienst) anzusprechen
- die intuitive Lösung, als Ziel einen Prozeß zu nehmen hat Nachteile
  - Prozesse werden dynamisch erzeugt und vernichtet
  - Prozesse können ersetzt werden - die PID ändert sich dadurch
  - Ziele sollten aufgrund ihrer Funktion (Dienst) ansprechbar sein
  - Prozesse könnten mehrere Dienste anbieten (vgl. *inetd*)

AKBP I

Ausgewählte Kapitel der praktischen Betriebsprogrammierung

© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999

H-Net.doc 1999-01-27 09.05

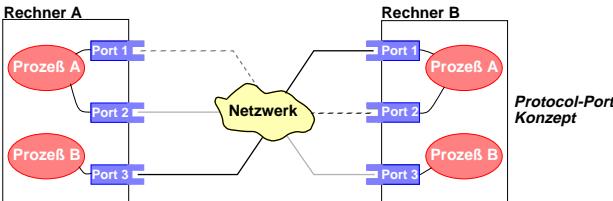
H.16

Reproduktion jeder Art oder Verwendung dieser Unterrichtsfolie außer zu Lehrzwecken an der Universität Erlangen-Nürnberg bedarf der Zustimmung des Autors.

## 2 Protocol-Port Konzept

### H.4 User Datagram Protocol - UDP

- Protocol-Ports sind abstrakte Kommunikations-Endpunkte
- Pakete, die einem Port zugestellt werden, werden in einer Warteschlange gehalten, bis sie abgeholt werden
- Empfangsanforderungen an einen Port werden blockiert, bis ein Paket eintrifft
- zur Kommunikation mit einem anderen Port müssen **Port-Nummer** und IP-Adresse bekannt sein



AKBP I

Ausgewählte Kapitel der praktischen Betriebsprogrammierung

© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999  
H-Net.doc 1999-01-27 09.05

H.17

Reproduktion jeder Art oder Verwendung dieser Unterrichts- oder Lehrressourcen an der Universität Erlangen-Nürnberg bedarf der Zustimmung des Autors.

## 5 Transmission Control Protocol - TCP

- IP stellt nur unzuverlässige Datenübertragung zur Verfügung
- TCP sichert die Übertragung gegen Fehler (Datenverlust, Paketverdopplung, falsche Reihenfolge)
- TCP versucht die zur Verfügung stehende Netzkapazität möglichst "freundlich" und optimal auszunutzen
- das Port-Konzept von UDP ist auch in TCP realisiert
- TCP-Kommunikation hat folgende Eigenschaften
  - ◆ **Stream-Orientierung** - die Daten werden als Byte-Strom entgegengenommen / abgeliefert
    - unstrukturiert (keine Paketgrenzen!)
    - gepuffert (IP-Pakete werden - wenn möglich - gefüllt)
  - ◆ **virtuelle Verbindung** - vor Beginn der Datenübertragung muß eine Verbindung zum Kommunikationspartner aufgebaut werden
  - ◆ voll-duplex Verbindung

AKBP I

Ausgewählte Kapitel der praktischen Betriebsprogrammierung

© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999  
H-Net.doc 1999-01-27 09.05

H.19

Reproduktion jeder Art oder Verwendung dieser Unterrichts- oder Lehrressourcen an der Universität Erlangen-Nürnberg bedarf der Zustimmung des Autors.

## 2 ... Protocol-Port Konzept (2)

### H.4 User Datagram Protocol - UDP

- für eine Reihe von Standard-Diensten (route-Protokoll, rwho, ...) sind Portnummern festgelegt (UNIX → Datei `/etc/services`)
- UDP agiert als Multiplexer/Demultiplexer auf IP
- UDP bietet sonst keine zusätzliche Funktionalität gegenüber IP - UDP-Pakete können
  - ◆ verloren gehen
  - ◆ in falscher Reihenfolge beim Empfänger ankommen

AKBP I

Ausgewählte Kapitel der praktischen Betriebsprogrammierung

© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999  
H-Net.doc 1999-01-27 09.05

H.18

Reproduktion jeder Art oder Verwendung dieser Unterrichts- oder Lehrressourcen an der Universität Erlangen-Nürnberg bedarf der Zustimmung des Autors.

## 6 Systemschnittstelle

### H.6 Systemschnittstelle

#### 1 Internet-Domain Sockets

##### 1 Adressierung

- Rechner-Adresse (32-bit, **IP-Address**) und Portnummer

- Include-Datei: <netinet/in.h>, Struktur `sockaddr_in`:

```
struct sockaddr_in {  
    short sin_family; /* =AF_INET */  
    short sin_port; /* Port */  
    struct in_addr sin_addr; /* IP-Addr */  
    char sin_zero[8];  
};
```

**sin\_port:** Portnummer (1 bis 1023 reserviert!)  
ab 1024 frei wählbar  
0 = Port wird vom System gewählt

**sin\_addr:** IP-Adresse, mit `gethostbyname(3)` zu finden

AKBP I

Ausgewählte Kapitel der praktischen Betriebsprogrammierung

© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999  
H-Net.doc 1999-01-27 09.05

H.20

Reproduktion jeder Art oder Verwendung dieser Unterrichts- oder Lehrressourcen an der Universität Erlangen-Nürnberg bedarf der Zustimmung des Autors.

## 2 Rechnernamen / IP-Adressen

[H.6 Systemschnittstelle](#)

- IP-Adressen sind für den Benutzer unhandlich  
→ Vergabe von Rechnernamen
- Form der Rechnernamen bei größerer Rechnerzahl wichtig
  - ◆ Abbildung Rechnername ↔ Adresse muß möglich sein
  - ◆ es dürfen keine Namenskonflikte auftreten
- flacher Namensraum
  - ◆ Namen sind Sequenz von Zeichen, ohne Strukturierung
  - ◆ zentrale Vergabe notwendig
  - ◆ bei großem Namensraum unmöglich zu verwalten
- hierarchischer Namensraum
  - ◆ vergleichbar mit UNIX-Dateinamen
  - ◆ Namensvergabe dezentral möglich
  - ◆ Abbildung Name ↔ Adresse ebenfalls dezentral möglich

AKBP I

Ausgewählte Kapitel der praktischen Betriebsprogrammierung

©

Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999

H-Inet.doc 1999-01-27 09.05

Reproduktion jeder Art oder Verwendung dieser Unterrichtsfolie außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

H.21

©

Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999

H-Inet.doc 1999-01-27 09.05

Reproduktion jeder Art oder Verwendung dieser Unterrichtsfolie außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

## 2 Rechnernamen / IP-Adressen (2)

[H.6 Systemschnittstelle](#)

- dezentrale Abbildung über **Nameserver**
  - ◆ Beispiel: faui04.informatik.uni-erlangen.de
- Abbildung Rechnernamen ↔ IP-Adressen (Schnittstelle)
- Funktionen:
  - gethostent(3)**
  - gethostbyname(3)**
  - gethostbyaddr(3)**
- Include-Datei: <netdb.h>  
Struktur **hostent**:

```
struct hostent {  
    char *h_name; /* official name of host */  
    /*  
    char **h_aliases; /* alias list */  
    int h_addrtype; /* host address type */  
    (AF_INET) */  
    int h_length; /* length of address */  
    char **h_addr_list /* list of addresses from  
    name server */
```

AKBP I

Ausgewählte Kapitel der praktischen Betriebsprogrammierung

©

Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999

H-Inet.doc 1999-01-27 09.05

Reproduktion jeder Art oder Verwendung dieser Unterrichtsfolie außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

H.22

©

Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999

H-Inet.doc 1999-01-27 09.05

Reproduktion jeder Art oder Verwendung dieser Unterrichtsfolie außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

## 3 weitere Verwaltungsdaten

[H.6 Systemschnittstelle](#)

- ★ analog zu den Rechnerdaten gibt es Zuordnungsmechanismen für:
  - Netzwerkadressen
    - Datei: /etc/networks
    - Funktionen: **getnetent(3), getnetbyname(3), getnetbyaddr(3)**
  - Protokollnamen (TCP, UDP, ...)
    - Datei: /etc/protocols
    - Funktionen: **getprotoent(3), getprotobynam(3), getprotobynumber(3)**
  - Service ↔ Portnummer
    - Datei: /etc/services
    - Funktionen: **getservent(3), getservbyname(3), getservbyport(3)**

AKBP I

Ausgewählte Kapitel der praktischen Betriebsprogrammierung

© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999

H-Inet.doc 1999-01-27 09.05

Reproduktion jeder Art oder Verwendung dieser Unterrichtsfolie außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

H.23

## 5 spezielle Funktionalität

- Broadcast-Pakete
  - ◆ nur bei Datagramm-Sockets möglich
  - ◆ kann von allen Rechnern auf dem Netz empfangen werden
  - ◆ netzwerkunabhängige Programmierung kompliziert, da zuerst die angeschlossenen Netzwerke erfragt werden müssen
- Out of band data
  - ◆ dringende Daten, die bevorzugt empfangen werden sollen
  - ◆ nur bei Stream-Sockets
  - ◆ senden mit **send(2)**, empfangen mit **recv(2)**
  - ◆ bewirkt beim Empfänger das Signal **SIGURG**

AKBP I

Ausgewählte Kapitel der praktischen Betriebsprogrammierung

© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999

H-Inet.doc 1999-01-27 09.05

Reproduktion jeder Art oder Verwendung dieser Unterrichtsfolie außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

H.24