

Eingebettete Systeme

Prof. Dr. rer. nat. Peter Gerwinski

24. November 2020

Eingebettete Systeme

<https://gitlab.cvh-server.de/pgerwinski/es>

1 Einführung

2 Einführung in Unix

2.1 Grundkonzepte

2.2 Die Kommandozeile: Grundlagen

2.3 Dateisysteme

2.4 Ein- und Ausgabeströme

2.5 Pipes

2.6 Verzweigungen und Schleifen

2.7 Skript-Programmierung

2.8 GNU screen

3 TCP/IP in der Praxis

4 Bus-Systeme

...

2.4 Ein- und Ausgabeströme

- Standard-Ausgabe in Datei umleiten

```
$ echo "Dies ist ein Test." > test.txt
```

```
$ cat test.txt
```

```
Dies ist ein Test.
```

- Standard-Ausgabe an Datei anhängen

```
$ echo "Dies ist noch ein Test." >> test.txt
```

```
$ cat test.txt
```

```
Dies ist ein Test.
```

```
Dies ist noch ein Test.
```

2.4 Ein- und Ausgabeströme

- Fehler-Ausgabe in Datei umleiten

```
$ cat gibtsnicht.txt > fehler.txt
cat: gibtsnicht.txt: No such file or directory
$ cat fehler.txt
$ cat gibtsnicht.txt 2> fehler.txt
$ cat fehler.txt
cat: gibtsnicht.txt: No such file or directory
```

2.4 Ein- und Ausgabeströme

- Standard-Eingabe aus Datei lesen

```
$ bc
bc 1.06.95
Copyright [...] 2006 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type 'warranty'.
2 + 2
4
$ echo "2 + 2" > test.bc
$ bc < test.bc
4
```

2.5 Pipes

Standard-Ausgabe von Programm A
wird zu Standard-Eingabe von Programm B

```
$ echo "2 + 2" | bc  
4
```

→ sehr mächtiger „Baukasten“

2.5 Pipes

- `sed`: *stream editor*

Suchen und Ersetzen (und noch viel mehr)

```
$ echo "Schlimmer geht nimmer." | sed -e 's/nim/im/g'  
Schlimmer geht immer.
```

2.5 Pipes

- **grep**: Standard-Eingabe durchsuchen

```
$ ls | grep slides
```

```
pgslides.sty
```

```
$ ls *.pdf | grep -v logo
```

```
es-20191002.pdf
```

```
Zeichen_123.pdf
```

```
$ ls -l $(ls *.pdf | grep -v logo)
```

```
-rw-r--r-- 1 ... 4619138 Okt 8 21:28 es-20191002.pdf
```

```
lrwxrwxrwx 1 ...          25 Okt 3  2016 Zeichen_123.pdf -> ...
```


2.5 Pipes

- Standard-Ausgabe im Befehl verwenden

```
$ grep -l "|" *.txt
regex-1.txt
regex-2.txt
regex-3.txt
regex-4.txt
shell-skripts-11.txt
X-1.txt
$ vi $(grep -l "|" *.txt)
```

2.6 Verzweigungen und Schleifen

```
$ if grep Blubb test.txt; then echo "gefunden"; \  
  else echo "nicht gefunden"; fi  
nicht gefunden  
$ for x in foo bar baz; do echo $x; done  
foo  
bar  
baz
```

2.7 Skript-Programmierung

Aufgabe: Programmieren Sie ein Unix-Shell-Skript, das etwas Nützliches macht. :-)

Beispiel: CSV-Import-Interface

- CSV-Datei: „comma-separated values“
Tabelle, durch Kommata getrennt
- Viele Tabellenkalkulationsprogramme können CSV-Dateien exportieren.
- Beispiel-Datei: [test.csv](#) (siehe auch: [test.ods](#))
- Beispiel-Anwendung: Ermitteln der durchschnittlichen Note
- Beispiel-Anwendung für Datenbank-Experten:
Speichere die Spalten 1 und 2 in einer Datenbank.
- Hinweis 1: `cat X-1.txt`
- Hinweis 2: `man cut`

2.8 GNU screen

- Text-Bildschirm und Eingabegeräte über's Netz
- `Ctrl+A c`: neues Fenster (create)
- `Ctrl+A Leertaste`: nächstes Fenster
- `Ctrl+A 3`: Fenster Nr. 3
- `Ctrl+A ESC`: hochscrollen, suchen, markieren (copy)
- `Ctrl+A]`: einfügen (paste)
- `Ctrl+A d`: von `screen` ablösen (detach)
- `screen -x`: an laufenden `screen` andocken
- ähnliche Funktionalität für Grafik: `VNC`, `x2go`

Eingebettete Systeme

<https://gitlab.cvh-server.de/pgerwinski/es>

1 Einführung

2 Einführung in Unix

2.1 Grundkonzepte

2.2 Die Kommandozeile: Grundlagen

2.3 Dateisysteme

2.4 Ein- und Ausgabeströme

2.5 Pipes

2.6 Verzweigungen und Schleifen

2.7 Skript-Programmierung

2.8 GNU screen

3 TCP/IP in der Praxis

4 Bus-Systeme

...

Eingebettete Systeme

<https://gitlab.cvh-server.de/pgerwinski/es>

- 1 Einführung**
- 2 Einführung in Unix**
- 3 TCP/IP in der Praxis**
 - 3.0** Vorbereitungen
 - 3.1** IP-Adressen
 - 3.2** MAC-Adressen
 - 3.3** TCP- und UDP-Ports
 - 3.4** TCP-Protokolle
 - 3.5** Routing
 - 3.6** Netzwerkanalyse
 - 3.7** SSH
 - 3.8** X11
 - 3.9** Programmierung
- 4 Bus-Systeme**

...

3 TCP/IP in der Praxis

Literatur:

<http://www.peter.gerwinski.de/download/net-2013ss.tar.gz>

3.0 Vorbereitungen **(Nicht jetzt ausführen!)**

- Verkabelung: Twisted-Pair-Kabel, Switches
- Automatismen abschalten

```
# service network-manager stop
```

3.1 IP-Adressen

- `ip addr` (Linux)
- `ifconfig` (Unix allgemein)
- `ipconfig` (MS Windows)
- `ip addr add <Netz>`
- `ip link`
- `ping <IP-Adresse>`

```
# ifconfig
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
    [...]

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.42.101  netmask 255.255.255.0
                                broadcast 192.168.42.255
    ether be:3f:ca:aa:7e:51 txqueuelen 1000  (Ethernet)
    [...]
```


3.1 IP-Adressen

IPv4-Adressen:

- 32 Bit
- dezimal, 4 Gruppen zu je 8 Bit (0–255), durch Punkte getrennt

IPv6-Adressen:

- 128 Bit
- hexadezimal, 8 Gruppen zu je 4 Hex-Ziffern, durch Doppelpunkte getrennt
- Führende Nullen dürfen weggelassen werden.
- Zwei Doppelpunkte bedeuten: Mit Nullen auffüllen.
- Literatur und Beispiel: <https://de.wikipedia.org/wiki/IPv6>

3.2 MAC-Adressen

MAC = Media Access Control

MAC-Adresse = Hardware-Adresse = Ethernet-Adresse

- `ip neigh`
`arp`

3.3 TCP- und UDP-Ports

- `nc <IP> <Port>`
Verbindung zu Programm $\langle \text{Port} \rangle$ auf Rechner $\langle \text{IP} \rangle$ aufnehmen
- `nc -l <Port>` oder `nc -p <Port> -l`
auf eingehende Verbindungen warten („lauschen“)
- TCP-Ports: Verbindungskonzept, Netzwerk prüft
- UDP-Ports: einzelne Pakete, Anwendung muß selbst prüfen
- ICMP: keine Ports, nur Rechner:
Erreichbarkeit, Eigenschaften der Übertragung

Anwendung: HTTP, SMTP, ...
Transport: TCP-/UDP-Ports, ICMP
Internet: IP-Adresse
Netzwerkzugang: Hardware-/MAC-Adresse

3.4 TCP-Protokolle

- **HTTP**

GET / HTTP/1.1

Host: www.hs-bochum.de

(Leerzeile)

3.4 TCP-Protokolle

- **HTTP**

```
GET / HTTP/1.1  
Host: www.hs-bochum.de  
(Leerzeile)
```

- **SMTP**

```
HELO cassini  
MAIL FROM: <example@example.com>  
RCPT TO: <beispiel@example.de>  
(E-Mail-Header – Teil der Nutzdaten)  
From: Eddie Example <example@example.com>  
To: Bert Beispiel <beispiel@example.de>  
Subject: Hello, world!  
(Leerzeile)  
Hi, there!  
.
```

- Protokolle „mal eben“ selbst schreiben: `nc -c` oder `inetsd`