

Datenbanken und Datensicherheit

Prof. Dr. rer. nat. Peter Gerwinski

9. Oktober 2024

Vorab: Online-Werkzeuge

- Diese Veranstaltung findet **in Präsenz** statt.
Wir versuchen aber, auch eine Online-Teilnahme zu ermöglichen.
- **Mumble**: Seminarraum 2
Fragen: Mikrofon einschalten oder über den Chat
Umfragen: über den Chat – **auch während der Präsenz-Veranstaltung**
- **VNC**: Kanal 6, Passwort: `testcvh`
Eigenen Bildschirm freigeben: VNC-Software oder Web-Interface *yesVNC*
Eigenes Kamerabild übertragen: Web-Interface *CVH-Camera*
- Allgemeine Informationen: <https://www.cvh-server.de/online-werkzeuge/>
- Notfall-Schnellzugang: <https://www.cvh-server.de/virtuelle-raeume/>
Seminarraum 2, VNC-Passwort: `testcvh`
- **Lehrmaterialien**: <https://gitlab.cvh-server.de/pgerwinski/dbs>

Was sind Datenbanken?

Datenbank = Datenbestand + System, um darauf zuzugreifen

Anforderungen an Datenbank [\[Wikipedia\]](#):

- effizient → Sortierung, Zugriffsalgorithmen
- widerspruchsfrei → automatische Konsistenzprüfungen
- dauerhaft → Backup, Ausfallsicherheit

Was sind Datenbanken?

Datenbank ohne Computer

- menschliche Anwesenheit erforderlich
- sorgfältig geordnet für effizienten Zugriff
→ sorgfältiger Umgang erforderlich, um Ordnung zu erhalten
- Jeweils 1 Person kann an 1 Papier arbeiten.
→ Hardware-Unterstützung für Konsistenz
- Kopien sehr aufwendig
→ sorgfältige Archivierung erforderlich



Bildquelle: https://commons.wikimedia.org/wiki/File:A_Day_in_the_Life_of_a_Wartime_Housewife-_Everyday_Life_in_London,_England,_1941_D2379.jpg

Was sind Datenbanken?

Zentraler Computer

- menschliche Anwesenheit erforderlich
- sorgfältig geordnet für effizienten Zugriff
 - Computer kann helfen, Ordnung zu erhalten
- Jeweils 1 Person kann am Computer arbeiten.
 - Hardware-Unterstützung für Konsistenz
- Backups möglich



Bildquelle: [https://commons.wikimedia.org/wiki/File:Ken_Thompson_\(sitting\)_and_Dennis_Ritchie_at_PDP-11_\(2876612463\).jpg](https://commons.wikimedia.org/wiki/File:Ken_Thompson_(sitting)_and_Dennis_Ritchie_at_PDP-11_(2876612463).jpg)

Was sind Datenbanken?

Zentraler Computer, Zugriff von Arbeitsplätzen aus

- Arbeiten auf Entfernung möglich
- sorgfältig geordnet für effizienten Zugriff
→ Computer kann helfen, Ordnung zu erhalten
- Mehrere Personen können unbemerkt gleichzeitig an denselben Daten arbeiten.
→ Computer muß helfen, Konsistenz zu erhalten
- Backups möglich



Bildquelle: https://commons.wikimedia.org/wiki/File:Computergebouw_van_KLM_voor_automatische_boekingsmethode_Corda_in_Amstelveen._,_BestanddeelNr_923-3365.jpg

Was sind Datenbanken?

Zentraler Computer, öffentlicher Zugriff

- Selbstbedienung auf Entfernung möglich
- sorgfältig geordnet für effizienten Zugriff
→ Computer muß selbständig Ordnung erhalten
- Mehrere Personen können unbemerkt gleichzeitig an denselben Daten arbeiten.
→ Computer muß selbständig Konsistenz erhalten
- Backups möglich
- Bessere Zugriffsmöglichkeiten → Datensicherheit wird Herausforderung



Bildquelle: https://commons.wikimedia.org/wiki/File:Shopping_online_with_bank_card.jpg

Was ist Datensicherheit?

(CIA)

- Vertraulichkeit (confidentiality) → Verschlüsselung
- Integrität (integrity) → Konsistenzprüfungen
- Verfügbarkeit (availability) → Backups, Ausfallsicherheit

- Identifizierbarkeit
(Authentizität, Nichtabstreitbarkeit, Zurechenbarkeit)
→ Passwörter, Signaturen

bzw.

- Anonymität
(plausible Abstreitbarkeit, Nichtzurechenbarkeit)
→ Pseudonymisierung, Anonymisierung

In dieser Lehrveranstaltung

- Kurzeinführung: Unix-Shell
- Kurzeinführung: TCP/IP
- relationale Datenbank-Management-Systeme (DBMS)
- SQL-Programmierung
- sonstige Datenbank-Management-Systeme

- Kurzeinführung: Kryptographie
- Kryptographie in der Praxis:
Passwörter, Verschlüsselung, Signaturen,
Schlüssel-Infrastrukturen
- Netzwerksicherheit
- Sicherheit von Web-Anwendungen
- Datenschutz



Änderungen
vorbehalten

In dieser Lehrveranstaltung

3 Praktikumsversuche

1. Selbstbau eines einfachen DBMS
2. Selbstbau einer prototypischen, sicheren Datenbankanwendung
3. E-Mail-Verschlüsselung

Prüfungsleistung: Klausur



Änderungen
vorbehalten

In dieser Lehrveranstaltung

3 Praktikumsversuche

- 0. Praxiserfahrung mit Unix und TCP/IP
- 1. Selbstbau eines einfachen DBMS
- 2. Selbstbau einer prototypischen, sicheren Datenbankanwendung
- 3. E-Mail-Verschlüsselung

Keine festen Abgabetermine, sondern:
Angebot zum betreuten Arbeiten im DV-Pool,
Vorzeigen (Testieren) der Ergebnisse

Prüfungsleistung: Klausur



Änderungen
vorbehalten

Datenbanken und Datensicherheit

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

1 Einführung

- 1.1 Was sind Datenbanken?
- 1.2 Was ist Datensicherheit?
- 1.3 In dieser Lehrveranstaltung

2 Kurzeinführung 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

3 Kurzeinführung TCP/IP

...



Änderungen
vorbehalten

2 Kurzeinführung Unix

2.1 Grundkonzepte

- 1965** Vorgänger: Multics (Multiplexed Information and Computing Service)
„überladen“
- 1970** Unix: Einfachheit als Grundkonzept
- 1972** Umstellung auf neu entwickelte Programmiersprache C
- 1975** AT&T: Unix inkl. Quelltext für Universitäten
- 1977** Berkeley Software Distribution (BSD)
- 1983** GNU-Projekt
- 1987** Minix
- 1991** Linux
- 1993** FreeBSD, NetBSD
- 1994** OpenBSD
- 2000** Darwin (Mac OS X, BSD-basiert)
- 2008** Android (Linux-basiert)

2 Kurzeinführung Unix

2.1 Grundkonzepte

Unix und C: Einfachheit als Grundkonzept

- Vermeiden von Ausnahmen
- Baukastensystem

C: Hauptprogramm
= „normale“ Funktion

```
int main (int argc, char **argv)
{
    printf ("Hello, _world!\n");
    return 0;
}
```

Unix: übergeordnetes Verzeichnis = „normales“ Verzeichnis

```
cassini/home/peter/foo> ls -la
insgesamt 24
drwxr-xr-x  2 peter peter  4096 Okt  6 13:30 .
drwxr-xr-x 172 peter peter 20480 Okt  6 13:30 ..
cassini/home/peter/foo> cd ..
cassini/home/peter>
```

2 Kurzeinführung Unix

2.1 Grundkonzepte

Unix und C: Einfachheit als Grundkonzept

- Vermeiden von Ausnahmen
- Baukastensystem

C: Bibliotheken

z. B.: `printf()` = „normale“ Funktion
aus eine Bibliothek (`libc`)

Unix: Programme arbeiten zusammen

```
cassini/home/peter/bo> find . -name "*klausur*.tex" \  
| xargs grep -l "PBM-Datei"  
./2014ws/ainf/20150130.0/ainf-klausur-20150130.tex  
./2016ws/hp/20170920.0/klausur.tex  
./2016ws/hp/20170206.0/klausur.tex  
./2011ws/rarch/20120322.0/rarch-klausur-20120322.tex  
./2012ws/klausuren-gerwinski/rarch-klausur-20120322.tex  
./2013ws/ainf/20140918.0/ainf-klausur-20140918.tex  
./2017ws/hp/20180213.k1/klausur.tex  
./2017ws/hp/20180205/klausur.tex  
./2015ws/ainf/20160913/ainf-klausur-20160913.tex
```

2.2 Die Kommandozeile: Grundlagen

- Programm aufrufen: Namen eingeben, z. B.: `ls`
- Optionen: `ls -l`
- Lange Optionen (GNU-Konvention): `ls --help`
- Text schreiben: `echo "Hello, world!"`
- (String-)Variable setzen: `FOO=bar`
- Variable einlesen: `read FOO`
- Variable abrufen: `echo $FOO`
- Aus Sicherheitsgründen: `echo "$FOO"`

```
cassini/home/peter/bo> FOO=ls
cassini/home/peter/bo> echo "$FOO"
ls
cassini/home/peter/bo> $FOO
2011ws  2012ws  2013ws  doc          misc  projekte
2012ss  2013ss  briefe  material    orga
cassini/home/peter/bo>
```


2.2 Die Kommandozeile: Grundlagen

- Befehl zurückholen: Pfeiltasten \uparrow , \downarrow
- Befehl bearbeiten: Pfeiltasten \leftarrow , \rightarrow usw.
- Befehl vervollständigen: TAB
- Befehl rückwärts suchen: Ctrl+R
- Bildschirm löschen: Ctrl+L
- Befehl abbrechen: Ctrl+C

- Hilfe-Option: `ls --help`
- Unix-Handbuch – *manual*: `man ls`
(Beenden mit `q`)

Datenbanken und Datensicherheit

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

1 Einführung

- 1.1 Was sind Datenbanken?
- 1.2 Was ist Datensicherheit?
- 1.3 In dieser Lehrveranstaltung

2 Kurzeinführung 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

3 Kurzeinführung TCP/IP

...



Änderungen
vorbehalten

2.2 Die Kommandozeile: Grundlagen

- Verzeichnisse für Programme: `echo $PATH`
- Programm in explizitem Verzeichnis aufrufen: `/bin/ls -l`
- Programm im aktuellen Verzeichnis aufrufen: `./hello`

MS-DOS: Ausführbare Programme werden gefunden,
wenn sie im `PATH` stehen
oder sich im aktuellen Verzeichnis befinden.

Unix: Ausführbare Programme werden gefunden,
wenn sie im `PATH` stehen.

—→ Vermeiden von Ausnahmen

Das aktuelle Verzeichnis (`.`) *kann* im `PATH` stehen,
muß dies aber nicht.

2.2 Die Kommandozeile: Grundlagen

- Verzeichnisse für Programme: `echo $PATH`
- Programm in explizitem Verzeichnis aufrufen: `/bin/ls -l`
- Programm im aktuellen Verzeichnis aufrufen: `./hello`

MS-DOS: Ausführbare Programme werden gefunden,
wenn sie im `PATH` stehen
oder sich im aktuellen Verzeichnis befinden.

Unix: Ausführbare Programme werden gefunden,
wenn sie im `PATH` stehen.

—→ Vermeiden von Ausnahmen

Das aktuelle Verzeichnis (`.`) *kann* im `PATH` stehen,
muß dies aber nicht –
und sollte es aus Sicherheitsgründen auch nicht.

2.3 Dateisysteme

- Dateien listen: `ls`
langes Listenformat: `ls -l`
rückwärts nach Zeit sortiert: `ls -lrt`
- Datei ausgeben: `cat hello.c`
- Datei anzeigen: `less hello.c`

2.3 Dateisysteme

- Arbeitsverzeichnis anzeigen: `pwd`
- Arbeitsverzeichnis wechseln: `cd script`
(*kein* Programm, sondern Shell-Befehl)
- übergeordnetes Verzeichnis: `cd ..`
- eigenes *Home*-Verzeichnis: `cd`
- Wurzelverzeichnis: `cd /`
- wieder zurück: `cd -`

```
cassini/home/peter/bo/2013ss/net/script> cd /usr/bin
cassini/usr/bin> cd ../lib
cassini/usr/lib> cd
cassini/home/peter>
```

2.3 Dateisysteme

- Dateien kopieren (*copy*): `cp`
- Dateien verschieben/umbenennen (*move*): `mv`
- Dateien löschen (*remove*): `rm`

```
cassini/home/peter> cp -p foo/test.txt
cp: missing destination file operand after `foo/test.txt'
Try `cp --help' for more information.
cassini/home/peter> cp -p foo/test.txt .
cassini/home/peter> mv test.txt bla.txt
cassini/home/peter> cat bla.txt
Dies ist ein Test.
cassini/home/peter> rm bla.txt
cassini/home/peter>
```

Aktuelles Verzeichnis: `.`

2.3 Dateisysteme

- *Zugriffsrechte*

```
cassini/home/peter/bo/2019ws/es/20191009> ls -l
...
-rw-r--r-- 1 peter peter 24523 Okt  8 21:47 es-20191009.tex
```


2.3 Dateisysteme

- Zugriffsrechte

```
cassini/home/peter/bo/2019ws/es/20191009> ls -l
```

```
...
```

```
-rw-r--r-- 1 peter peter 24523 Okt  8 21:47 es-20191009.tex
```



Benutzer (u – *user*) darf lesen und schreiben


2.3 Dateisysteme

- Zugriffsrechte

```
cassini/home/peter/bo/2019ws/es/20191009> ls -l
```

```
...
```

```
-rw-r--r-- 1 peter peter 24523 Okt  8 21:47 es-20191009.tex
```



Gruppe (g – *group*) darf lesen

2.3 Dateisysteme

- *Zugriffsrechte*

```
cassini/home/peter/bo/2019ws/es/20191009> ls -l
```

```
...
```

```
-rw-r--r-- 1 peter peter 24523 Okt  8 21:47 es-20191009.tex
```



alle anderen (o – *other*) dürfen lesen

2.3 Dateisysteme

- *Zugriffsrechte*

```
cassini/home/peter/bo/2019ws/es/20191009> ls -l
...
-rw-r--r-- 1 peter peter 24523 Okt  8 21:47 es-20191009.tex
```

- Zugriffsrechte ändern:

```
chmod o-r es-20191009.tex – Lesezugriff entziehen
chmod g+w es-20191009.tex – Schreibzugriff gewähren
chmod 640 es-20191009.tex – auf -rw-r----- setzen
```

2.3 Dateisysteme

- Zugriffsrechte

```
cassini/home/peter/bo/2019ws/es/20191009> ls -l
...
-rw-r--r-- 1 peter peter 24523 Okt  8 21:47 es-20191009.tex
```

- Zugriffsrechte ändern:

```
chmod o-r es-20191009.tex – Lesezugriff entziehen
chmod g+w es-20191009.tex – Schreibzugriff gewähren
chmod 640 es-20191009.tex – auf rw-r----- setzen
                        6   4   0
```

2.3 Dateisysteme

- *ausführbare* Dateien

```
cassini/home/peter/bo/2019ws/es/20191002> cat test2.txt
ls -l
cassini/home/peter/bo/2019ws/es/20191002> chmod +x test2.txt
cassini/home/peter/bo/2019ws/es/20191002> ls -l test2.txt
-rwxr-xr-x 1 peter peter 6 Okt  2 13:43 test2.txt
cassini/home/peter/bo/2019ws/es/20191002> ./test2.txt
insgesamt 4828
lrwxrwxrwx 1 peter peter      18 Apr 13  2016 csa2.jpg -> ..
-rw-r--r-- 1 peter peter 4619138 Okt  8 21:28 es-20191002.pdf
...
```

- ausführbare Textdateien: *Skripte*

hier: ausführbare Textdatei mit Shell-Befehlen
(ohne spezielle Kennung): Shell-Skript

2.3 Dateisysteme

- *ausführbare* Dateien

```
cassini/home/peter/bo/2019ws/es/20191002> cat test2.txt
ls -l
cassini/home/peter/bo/2019ws/es/20191002> chmod +x test2.txt
cassini/home/peter/bo/2019ws/es/20191002> ls -l test2.txt
-rwxr-xr-x 1 peter peter 6 Okt  2 13:43 test2.txt
cassini/home/peter/bo/2019ws/es/20191002> ./test2.txt
insgesamt 4828
lrwxrwxrwx 1 peter peter      18 Apr 13  2016 csa2.jpg -> ..
-rw-r--r-- 1 peter peter 4619138 Okt  8 21:28 es-20191002.pdf
...
```

- ausführbare Textdateien: *Skripte*

hier: ausführbare Textdatei mit Shell-Befehlen
(ohne spezielle Kennung): Shell-Skript

Kennung: 1. Zeile enthält `#!` und den Interpreter,
z. B. `#!/bin/bash`

2.3 Dateisysteme

- Datenträger in Verzeichnis *einhängen*: `mount`

```
cassini/home/peter> ls /media/usb1/  
cassini/home/peter> mount /media/usb1  
cassini/home/peter> ls /media/usb1/  
es-20191002.pdf  hello.c  hexapode  KIS-Bericht.pdf  
cassini/home/peter> umount /media/usb1  
cassini/home/peter> ls /media/usb1/  
cassini/home/peter>
```


2.3 Dateisysteme

- *Symbolische Verknüpfungen – symbolic links*

Verweis auf die eigentliche Datei

→ Wenn man die Datei löscht, zeigt der Link ins Leere.

Verknüpfung anlegen: `ln -s datei link`

(Richtung: wie bei `cp`)

Beispiel: `ln -s ../common/GNU-GPL-3 gpl.txt`

2.3 Dateisysteme

- *Symbolische Verknüpfungen – symbolic links*

Verweis auf die eigentliche Datei

→ Wenn man die Datei löscht, zeigt der Link ins Leere.

Verknüpfung anlegen: `ln -s datei link`

(Richtung: wie bei `cp`)

Beispiel: `ln -s ../common/GNU-GPL-3 gpl.txt`

- *Harte Verknüpfungen – hard links*

Dieselben Daten auf dem Datenträger

sind unter mehreren Namen verfügbar.

→ Wenn man einen löscht, sind die Daten noch da.

```
cassini/home/peter/bo/2019ws/es/20191002> ls -l
```

```
...
```

```
-rw-r--r-- 1 peter peter      1202 Okt  2 13:35 shell-06.tx
```

```
drwxr-xr-x 2 peter peter     4096 Okt  2 13:16 test
```



Anzahl der („harten“) Links auf diese Datei / dieses Verzeichnis

2.3 Dateisysteme

- `grep`: Dateien durchsuchen

```
cassini/home/peter/bo/2019ws/es/20191002> grep gcc *.txt  
shell-03.txt: cassini/...> gcc -Wall -O hello.c -o hello
```

2.3 Dateisysteme

- **find**: Dateien anhand ihrer Eigenschaften suchen

```
$ find . -name "*.txt"
./shell-06.txt
./shell-03.txt
./shell-05.txt
./test.txt
./test/test.txt
...
$ find . -name "*.txt" -perm /u+x
./test2.txt
$ find . -name "*.txt" -perm /u+x -exec ls -l {} \;
-rwxr-xr-x 1 peter peter 6 Okt  2 13:43 ./test2.txt
```

2.4 Ein- und Ausgabeströme

- Standard-Ausgabe in Datei umleiten

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

```
$ cat test.txt
```

```
Dies ist ein Test.
```

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 -> ...
```

2.6 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
```

Datenbanken und Datensicherheit

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

1 Einführung

- 1.1 Was sind Datenbanken?
- 1.2 Was ist Datensicherheit?
- 1.3 In dieser Lehrveranstaltung

2 Kurzeinführung 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

3 Kurzeinführung TCP/IP

...



Änderungen
vorbehalten