

Rechnerpraxis

09 Finden

Hochschule
für Technik
Stuttgart

SS19 | INF 1

Ben Lebherz
benjamin.lebherz@hft-stuttgart.de

Vorlesungsplan

Datum	Thema
19. März	Motivation & Organisation
2. April	Grundlagen Unix/Linux
11. April	Dateisystem und Rechteverwaltung
18. April	dotfiles & Linux Installation
25. April	Prozesse und Benutzer
2. Mai	Wiederholung
9. Mai	Shell Magic & Scripting
16. Mai	VIM & Gastvortrag
28. Mai	Finden & Wiederholung
4. Juni	PVL-Vorbereitung Raum 2/433
6. Juni	PVL Termin
<i>13. Juni</i>	<i>Pfingstferien</i>
20. Juni	Zusammenfassung/Feedback (Vortrag: "Smart Home für jedermann")

Prüfung / PVL

Am **6. Juni 2019**

Bewertung

Bestanden oder Nicht Bestanden - keine Note

**! Bei Nichtbestehen *keine* Zulassung zur Klausur
Einführung in die Informatik !**

Prüfung / PVL | 6. Juni 2019

Erlaubte Hilfsmittel

- jegliche anderen Hilfsmittel werden als Betrugsversuch gewertet -

- 1x DIN A4 Zettel, doppelseitig, handschriftlich (nicht gedruckt)
- Vorlesungsfolien, ausgedruckt oder am Rechner geöffnet
- Terminal/Shell, am LIDA PC, keine eigenen Rechner

Prüfung / PVL | 6. Juni 2019

- Gruppeneinteilung wird im Moodle-Kurs bekannt gegeben, bitte beachten
- Bei genehmigter, längerer Prüfungszeit bitte rechtzeitig per Mail melden
- Während der PVL darf außer des E-Tests und den Vorlesungsfolien am Rechner kein weiteres Browserfenster / kein weiterer Browsertab geöffnet werden, d.h. es darf nicht auf das Internet zugegriffen werden (dies wird als Betrugsversuch mit entsprechenden Konsequenzen gewertet)
- Digitale Kommunikation zwischen Studenten jeglicher Art ist nicht erlaubt und wird als Betrugsversuch mit entsprechenden Konsequenzen gewertet
- Jegliche Kommunikation zwischen den Gruppen beim Wechsel ist nicht erlaubt und wird als Betrugsversuch mit entsprechenden Konsequenzen gewertet
- Bei technischen Problemen wird der E-Test gegebenenfalls zu einem anderen Termin wiederholt/nachgeholt
- Bei Nichtbestehen der PVL: *Keine Zulassung zur Klausur Einführung in die Informatik*

Evaluation



<https://evasys-online.hft-stuttgart.de>

15min

grep

Suche **in** Dateien

grep in einer Datei

Inhalt von Datei **file1**

```
delia hft $ cat file1
hallo
hallo
nein
nein
warum
hallo
nein
delia hft $
```


grep in einer Datei

Zeige nur Zeilen in **file1** die **nein** enthalte

grep in einer Datei

Inhalt von Datei **/etc/services**

```
delia hft $ cat /etc/services
# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, officially ports have two entries
# even if the protocol doesn't support UDP operations.
#
# Updated from https://www.iana.org/assignments/service-names-port-numbers/serv
#
# New ports will be added on request if they have been officially assigned
# by IANA and used in the real-world or are needed by a debian package.
# If you need a huge list of used numbers please install the nmap package.
```

grep in einer Datei

Port für **mysql** in der /etc/services Datei

```
delia hft $ cat /etc/services | grep mysql
mysql      3306/tcp
mysql      3306/udp
mysql-proxy 6446/tcp
mysql-proxy 6446/udp
delia hft $
```

Wie war der Befehl noch gleich?
Er enthielt irgendwie das Wort **freetz-wg...**

Wie war der Befehl noch gleich? Er enthielt irgendwie
freetz-wg

```
$ history | grep freetz-wg
33792: cp /home/ben/freetz-wg/images/7490_07.01-freetz-master-5629f9509-
33794: cp /home/ben/freetz-wg/images/7490_07.01-freetz-master-5629f9509-
33800: lh /home/ben/freetz-wg/images/
33801: cp /home/ben/freetz-wg/images/7490_07.01-freetz-master-5629f9509-
33802: cp /home/ben/freetz-wg/images/7490_07.01-freetz-master-5629f9509-
```

Und der Befehl direkt davor? oder danach?

Und der Befehl direkt davor? oder danach?

- **-A** X

"after"

zeigt X Zeilen nach dem gefundenen Wort

- **-B** X

"before"

zeigt X Zeilen vor dem gefundenen Wort

- **-C** X

zeigt X Zeilen davor und danach

Und der Befehl direkt davor? oder danach?

- **-A** (after), zeigt X Zeilen nach dem gefundenen Wort
- **-B** (before), zeigt X Zeilen vor dem gefundenen Wort
- **-C**, zeigt X Zeilen davor und danach

```
$ history | grep freetz-wg -B 2
33790: apt install bc
33791: su - ben
33792: cp /home/ben/freetz-wg/images/7490_07.01-freetz-
--
35828: cat /etc/services | grep "80/tcp"
35829: history
35830: history | grep freetz-wg
```


grep in mehreren Datei

grep in mehreren Datei

neue Domain

**In welchen Konfigurationsdateien ist meine (alte)
Domain eingetragen?**

In welchen Konfigurationsdateien ist meine Domain eingetragen?

- Domain: **benleb.de**
- Konfigurationsdateien in

In welchen Konfigurationsdateien ist meine Domain eingetragen?

- Domain: **benleb.de**
- Konfigurationsdateien in **/etc**

In welchen Konfigurationsdateien ist meine Domain eingetragen?

- Domain: **benleb.de**
- Konfigurationsdateien in **/etc** (und Unterordner da **-r**)

```
$ grep -r "benleb.de" /etc/
```

In welchen Konfigurationsdateien ist meine Domain eingetragen?

```
$ grep -r "benleb.de" /etc/  
/etc/network/interfaces:      dns-search benleb.de  
/etc/postfix/main.cf:myhostname = delia.benleb.de  
/etc/mailname:delia.benleb.de  
Binary file /etc/aliases.db matches  
/etc/aliases:root:           b@benleb.de  
/etc/hosts:10.0.0.10      delia.benleb.de      delia  
/etc/dnsmasq.conf:domain=benleb.de
```

In welchen Konfigurationsdateien ist meine Domain eingetragen?

- Domain: **benleb.de**
- Konfigurationsdateien in **/etc**

find

Suche **nach** Dateien

find

Durchsucht die Festplatte nach Dateien/Ordnern

find <WO> <WAS>

- **WO** soll gesucht werden?
- **WAS** für Kriterien zum suchen (Name, Größe, Alter, ...)
Klammern von Ausdrücken möglich
Negieren mit ! möglich

find

Durchsucht die Festplatte nach Dateien/Ordnern

find <WO> <WAS>

- **-name** <name> Datei/Ordnername nach dem gesucht wird

```
$ find /etc -name services
/etc/services
/etc/avahi/services
```

*Findet Dateien namens **services** in /etc*

find

Kriterien standardmäßig **AND** verknüpft

- **-a** verknüpft Kriterien mit **AND**
- **-o** verknüpft Kriterien mit **OR**

```
$ find /etc -name services -o -name aliases  
/etc/services  
/etc/aliases  
/etc/avahi/services
```

*Findet Dateien namens **services** ODER **aliases** in /etc*

find

Kriterien standardmäßig AND verknüpft

- **-type f** für Files/Dateien
- **-type d** für Directories/Order

```
$ find /etc -name services -o -iname alias* -type f
/etc/apparmor.d/tunables/alias
/etc/services
/etc/aliases.db
/etc/aliases
/etc/avahi/services
```

*Findet Dateien namens **services** ODER **alias*** in /etc*

find

Kriterien standardmäßig **AND** verknüpft

- **-type f** für Files/Dateien
- **-type d** für Directories/Order

```
$ find $HOME -name .ssh -type d  
/root/.ssh
```

*Findet Ordner namens **.ssh** in \$HOME*

Was sie für die PVL wissen sollten

- Keine vollständige Liste aber ein guter Leitfaden -

Was sie wissen sollten ^{1/3}

- Basics
 - Umgebungsvariablen, Bash Prompt, ...
 - **man, echo, env, export, id**, ...
- Dateisystem
 - Ordnerstruktur
 - Absolute/Relative Pfade
 - **pwd, cd, ls, mkdir, rm, cp, mv**
- Rechte
 - Oktalzahlen <> Rechte (rwx)
 - **chmod**

Was sie wissen sollten ^{2/3}

- Prozesse
 - Ausgabe von **ps** verstehen
 - **top, ps, kill, renice**, ...
- Dateien anzeigen/editieren
 - **cat, less, touch, vim**, ...
- Dateien mit VIM editieren können
 - Unterschied Command- und Insert-Modus verstanden haben und wechseln können
 - Elementare Operationen wie "x Zeilen löschen", "x Zeilen kopieren", "speichern und schließen", "aktuelle Zeile löschen", ...

Was sie wissen sollten ^{3/3}

- Standardausgabe weiterleiten und umleiten
 - Pipe |
 - stdout > und >>
- Shellskripte
 - Befehle automatisieren können
 - Einfache Shellskripte (in VIM) erstellen können
 - Parameter in Shellskripten benutzen (\$1, \$2, ...)
- Suchen und Finden
 - **grep** , **find**

grep & find Hands-on! 🧐💻

– Always work in your \$HOME –

1. Lassen Sie sich alle Dateien im aktuellen Verzeichnis anzeigen, die mit **f** anfangen
2. Lassen Sie sich **ihre** aktuell laufenden Prozesse anzeigen. Nutzen sie dazu **ps** und **grep**
 1. leiten Sie diese Prozess Auflistung in eine Datei **meineprozesse.log** um
3. Lassen Sie sich mit **find** die Home-Verzeichnisse aller Studenten **Ihres Jahrgangs** auflisten
4. Zeigen Sie alle Dateien in Ihrem \$HOME an, die (**in den letzten 2 Wochen verändert wurden** und **größer als 50 Bytes** sind) oder (mit **a** anfangen)
5. Mit welchen einzeiligen Befehl schreiben Sie nur die 4-Zeile der detaillierten Auflistung aller Dateien in Ihrem Home-Verzeichnis in eine Datei **4.log**? (Tipp: *head* und *tail* Befehle)
6. **Hard**: Lassen Sie sich alle aktuell laufenden Prozesse anzeigen, die Ihnen "gehören", aber davon jeweils nur die Prozess-ID ausgeben; leiten Sie diese Prozess-ID Auflistung in eine Datei **meineprozess-ids.log** um

Scripting Hands-on! 🧐💻

- Always work in your \$HOME -

1. Schreiben Sie ein Shellscript,

1. das einen Parameter annimmt
2. das alle Dateien des aktuellen Verzeichnisses in ein gzip-Archiv packt, der Name des Archivs ist der dem Skript übergebene Parameter, die Endung ist .gzip

2. Schreiben Sie ein Shellscript,

1. das einen Parameter annimmt
2. das eine ausführliche Auflistung aller in dem aktuellen Verzeichnis vorhandenen Dateien/Verzeichnisse in einer Datei ablegt, die den Namen des ersten Parameters des Skriptes trägt plus die Endung .log

3. Schreiben Sie ein "Superscript",

1. das drei Parameter annimmt
2. das die oben in Aufgabe 1 und 2 erstellten Skripte mit den Parametern 1 (für das erste Skript) und 2 (für das zweite Skript) aufruft
3. die erstellten .gzip und .log Dateien zusammen in eine .gzip Datei packt, die den Namen des dritten Parameters trägt

4. Modifizieren Sie das Superscript aus Aufgabe 3 so,

1. dass geprüft wird, ob die Datei \$3.zip schon existiert; wenn ja, soll eine Fehlermeldung auf der Konsole ausgegeben werden
2. Ein vierter Parameter entscheidet, ob bei Existenz von \$3.zip diese Datei überschrieben werden soll, oder mit Fehlermeldung abgebrochen werden soll

done! 🦊

Thanks!