



Linux und Bash: Warum und wie

Erstsemester-Einführung Informatik, 09.10.2019

Vorspann

Linux

Bash

Praxis

Abspann

<Eigenwerbung>

Wer sind wir?



- ▶ Hochschulgruppe an der TU (gegründet 2014, ca. 7 P.)
- ▶ Studierende (TU, HTW) und andere Leute
- ▶ Hochschulen als Zielgruppe (Multiplikationswirkung) und Arbeitsfeld (Räume, Strukturen)

Wer sind wir?



- ▶ Hochschulgruppe an der TU (gegründet 2014, ca. 7 P.)
- ▶ Studierende (TU, HTW) und andere Leute
- ▶ Hochschulen als Zielgruppe (Multiplikationswirkung) und Arbeitsfeld (Räume, Strukturen)

- ▶ Bisherige Projekte
 - ▶ Linux-Install-Party, Linux-Presentation-Day
 - ▶ Verschlüsselungsgewinnspiel
 - ▶ Sprechstunde zu \LaTeX u.a.
 - ▶ Formulierung eines Programmpapiers
 - ▶ „Uni-Stick“: 8 GB nützliche Freie Software fürs Studium
 - ▶ Ringvorlesung: Freie Software und Freies Wissen als Beruf

FSFW (1)



Wer sind wir?

- ▶ Hochschulgruppe (gegründet 2014, ca. 10 P.)
- ▶ bisherige Projekte:
 - ▶ Linux-Install-Party, Linux-Presentation-Day
 - ▶ Verschlüsselungsgewinnspiel
 - ▶ Monatliche [Sprechstunde](#) (Textsatzsystem \LaTeX , UniStick, ...)
 - ▶ Publikationen: [Programmpapier](#), [Blogbeiträge](#)
 - ▶ Workshops ([git](#), [python](#), [Mailverschlüsselung](#))
 - ▶ Ringvorlesung ('18): „[Freie Software / Freies Wissen als Beruf](#)“
 - ▶ [UniStick](#): 8 GB nützliche Freie Software fürs Studium

FSFW (2)



Warum machen wir das? → **Aus Überzeugung**

- ▶ *Überzeugung 1*: freie und quelloffene Software ist (oft) besser (technische + nicht technische Argumente)

FSFW (2)



Warum machen wir das? → **Aus Überzeugung**

- ▶ *Überzeugung 1*: freie und quelloffene Software ist (oft) besser (technische + nicht technische Argumente)
- ▶ *Überzeugung 2*: *öffentlich finanzierte* wissenschaftliche Inhalte (AutorInnen, GutachterInnen) sollten nicht von *öffentlich finanzierten* Bibliotheken für *horrende Summen* von Zeitschriften-Verlagen gekauft werden müssen



TECHNISCHE
UNIVERSITÄT
DRESDEN

<http://tu-dresden.de>

stura
DRESDEN

**Nicht:
Es muss etwas
geschehen,
sondern:
Ich muss etwas
tun.**

Hans Scholl, „Weiße Rose“

TU Dresden gegen Rechtsradikalismus und Intoleranz

WISSEN SCHAFFT
EXZELLENZ



TECHNISCHE
UNIVERSITÄT
DRESDEN

Technische Universität Dresden

Software: Frei versus Proprietär



Vier Freiheiten freier Software

Verwenden



Verstehen



Verändern



Verbreiten



Software: Frei versus Proprietär



Vier Freiheiten freier Software

Verwenden



Verstehen



Verändern



Verbreiten



Vorteile:

- ▶ Kontrolle behalten
- ▶ Erkenntnisgewinn
- ▶ Lizenzkosten: 0 €
- ▶ Anpassbarkeit an eigene Bedürfnisse
- ▶ Hersteller-Unabhängigkeit
(kein Vendor Lock-in)

Nachteile Proprietärer Software

- ▶ Intransparenz (Bsp: Wahlsoftware)
- ▶ Hintertüren? (Win10-Verbot für Dienstgebrauch)
- ▶ Abhängigkeit

Software: Frei versus Proprietär



Vier Freiheiten freier Software

Verwenden



Verstehen



Verändern



Verbreiten



Vorteile:

- ▶ Kontrolle behalten
- ▶ Erkenntnisgewinn
- ▶ Lizenzkosten: 0 €
- ▶ Anpassbarkeit an eigene Bedürfnisse
- ▶ Hersteller-Unabhängigkeit
(kein Vendor Lock-in)

Nachteile Proprietärer Software

- ▶ Intransparenz (Bsp: Wahlsoftware)
- ▶ Hintertüren? (Win10-Verbot für Dienstgebrauch)
- ▶ Abhängigkeit

Warum gibt es MS-Office eigentlich kostenlos für Studierende?

Software: Frei versus Proprietär



Vier Freiheiten freier Software

Verwenden



Verstehen



Verändern



Verbreiten



Vorteile:

- ▶ Kontrolle behalten
- ▶ Erkenntnisgewinn
- ▶ Lizenzkosten: 0 €
- ▶ Anpassbarkeit an eigene Bedürfnisse
- ▶ Hersteller-Unabhängigkeit
(kein Vendor Lock-in)

Nachteile Proprietärer Software

- ▶ Intransparenz (Bsp: Wahlsoftware)
- ▶ Hintertüren? (Win10-Verbot für Dienstgebrauch)
- ▶ Abhängigkeit

Warum gibt es MS-Office eigentlich „kostenlos“ für Studierende?

Software: Frei versus Proprietär



Vier Freiheiten freier Software

Verwenden



Verstehen



Verändern



Verbreiten



Vorteile:

- ▶ Kontrolle behalten
- ▶ Erkenntnisgewinn
- ▶ Lizenzkosten: 0 €
- ▶ Anpassbarkeit an eigene Bedürfnisse
- ▶ Hersteller-Unabhängigkeit
(kein Vendor Lock-in)

Nachteile Proprietärer Software

- ▶ Intransparenz (Bsp: Wahlsoftware)
- ▶ Hintertüren? (Win10-Verbot für Dienstgebrauch)
- ▶ Abhängigkeit

Warum gibt es MS-Office eigentlich „kostenlos“ für Studierende?

Offener Brief:

„public money ⇒ public code“

<https://publiccode.eu>

Freie Software an der TU



- ▶ ZIH: Es gibt eine **Open-Source-Strategie**
- ▶ StuRa: Bietet viele **digitale Dienste** auf Basis freier Software
 - ▶ Mumble (Telefonkonferenzen)
 - ▶ Kanban (Gruppenorganisation)
 - ▶ Etherpad (gemeinsam Texte schreiben)
 - ▶ ...

Zukunftsideen



- ▶ Fortführung „Uni-Stick“
- ▶ Studierende zum Nutzen/Verbessern freier Software animieren
 - ▶ Mehr Blog-Beiträge
 - ▶ Kurse (\LaTeX / Python / **Git** / Inkscape / ...)
 - ▶ Infrastruktur-Stipendium
 - ▶ OpenSource-Wettbewerb/Preis
 - ▶ ...
- ▶ Aufmerksamkeit erzeugen / Lobby-Arbeit („Landesverträge“)
- ▶ Vernetzung mit anderen Städten

Weitere Informationen



<https://fsfw-dresden.de/>

uni-stick
blog
newsletter
mitmachen
fork
git-ws
python-ws
bash-ws



</Eigenwerbung>

Vorspann

Linux

Bash

Praxis

Abspann

Hinweis



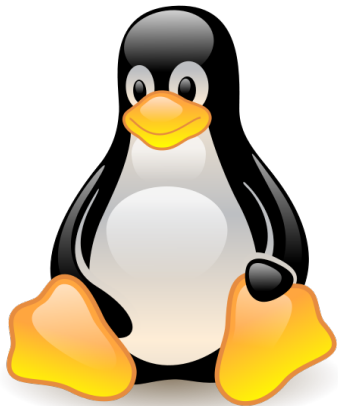
Die folgenden Folien basieren zum großen Teil auf [Material](#) von Dr. Ulf Markwardt (HPC@ZIH).

Vielen Dank!

General



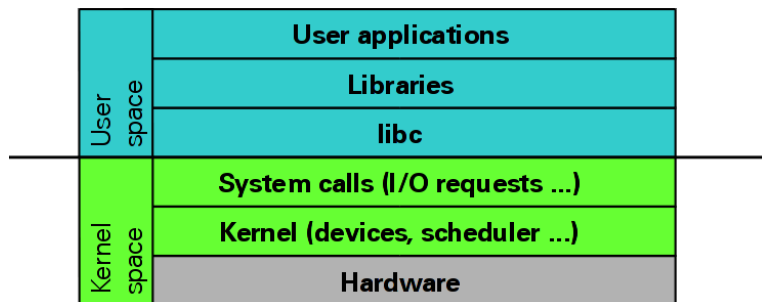
- ▶ first version 1991, Linus Torvalds
- ▶ hardware-independent operating system
- ▶ 'Linux' is the name of the kernel as well as of the whole operating system
- ▶ since 1993 under GNU public license (GNU/Linux)
- ▶ various distributions for all purposes (OpenSuSE, SLES, Ubuntu, Debian, Fedora, RedHat,...) <http://www.distrowatch.com>



Linux architecture



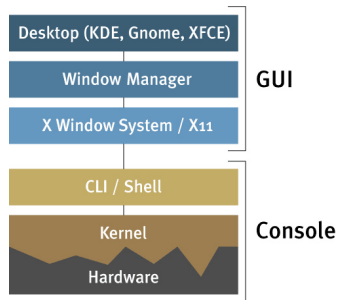
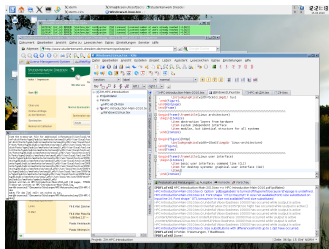
- ▶ abstraction layers from hardware
- ▶ system independent interface
- ▶ modular, but identical structure for all systems



Linux user interface



- ▶ basic user interface: command line (CLI)
- ▶ for desktop systems: graphical user interface (GUI)



Linux components



Major distributions come with rich variety of Open Source software:

- ▶ system kernel, libraries, device drivers
- ▶ file systems
- ▶ compilers, debuggers, scripting languages
- ▶ desktop tools like
 - ▶ GUI, window managers (KDE, Gnome, ICEWM..)
 - ▶ office tools (presentation, word processing, spread sheet)
OpenOffice
 - ▶ internet suite (browser, email, chat, video conferencing)
 - ▶ programming tools (IDE)
 - ▶ audio and video production
 - ▶ entertainment (audio, video, games)

Nearly all programs, libraries, devices have their manual pages easily accessible like `man ls` .

Vorspann

Linux

Bash

Praxis

Abspann

Execution of a program



“Bash is a command processor [...] where the user types commands that cause actions.” (Wikipedia)

- ▶ The shell tries to locate a program from an absolute or relative path like `/sbin/ifconfig` or `./myprog` or `bin/myprog`
- ▶ If this fails, it uses the search path to find the file:

```
/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin
```

- ▶ Program execution is controlled by command line options.

```
mark ~/devel> ls
a.out  test1  ulf.c
mark ~/devel> ls -la
total 40
drwxr-xr-x  3 mark zih  4096 Mar 29 09:22 .
drwxr-xr-x 58 mark zih 12288 Jun 18 11:21 ..
-rwxr-xr-x  1 mark zih 13739 Mar 29 09:22 a.out
drwxr-xr-x  8 mark zih  4096 Mar 28 23:17 test1
-rw-r--r--  1 mark zih  2426 Mar 29 09:18 ulf.c
```

Command shell - bash



[Main page](#)

[Contents](#)

[Featured content](#)

[Current events](#)

[Random article](#)

[Donate to Wikipedia](#)

[Wikipedia store](#)

Interaction

[Help](#)

[About Wikipedia](#)

[Community portal](#)

[Recent changes](#)

[Contact page](#)

Tools

Not logged in [Talk](#) [Contributions](#) [Create account](#) [Log in](#)

Article [Talk](#)

Read

[Edit](#)

[View history](#)



Command-line interface

From Wikipedia, the free encyclopedia

[...]

Today, many end users rarely, if ever, use command-line interfaces and instead rely upon [graphical user interfaces](#) and menu-driven interactions. **However**, many **software developers**;, system administrators and **advanced users** still rely heavily on command-line interfaces to perform tasks more efficiently, configure their machine, or access programs and program features that are not available through a graphical interface.

[...]

```

[...]
```

Screenshot of a sample **Bash** session in **GNOME Terminal 3, Fedora 15**



Command shell - bash



Umgebungsvariablen

- ▶ Beispiele:

`PATH` search path for binaries

`HOME` path to user's home directory

- ▶ Setzen:

```
export foobar="Hallo Welt"
```

- ▶ Auswerten:

```
echo $foobar
```

```
echo $HOME
```

```
echo $PATH
```

```
#!/bin/bash
FILES="*.tgz"
for f in "$FILES"
do
    tar -xzf $f # expand all files
done
```

Basic commands (1)



<code>pwd</code>	print work directory
<code>ls</code>	list directory (<code>ls -l /bin</code>)
<code>cd</code>	change directory (<code>cd</code> $\hat{=}$ <code>cd \$HOME</code>)
<code>mkdir</code>	create directory (<code>mkdir -p child/grandchild</code>)
<code>rm</code>	remove file/directory Caution: No trash bin!
<code>rmdir</code>	remove directory
<code>cp</code>	copy file/directory (<code>cp -r results /projectXY/</code>)
<code>mv</code>	move/rename file/directory (<code>mv results /projectXY/</code>)
<code>touch</code>	create file
<code>chmod</code>	change access properties (<code>chmod a+r readme.txt</code>)
<code>find</code>	find a file (<code>find . -name "*.c"</code>)

Basic commands (2)



<code>echo</code>	display text to stdout <code>echo \$PATH > tmp.txt</code>
<code>cat</code>	display contents of a file <code>cat tmp.txt</code>
<code>less</code> , <code>more</code>	pagewise display (<code>less tmp.txt</code>)
<code>grep</code>	search for words/text (<code>grep result out.res</code>)
<code>ps</code>	display running processes (<code>ps -A</code>)
<code>kill</code>	kill a process (<code>kill -9 12813</code>)
<code>htop</code>	display table of processes (interactive per default)
<code>ssh</code>	secure shell to a remote machine (<code>ssh s1230815@login.zih.tu-dresden.de</code>)

Basic commands (2)



<code>echo</code>	display text to stdout <code>echo \$PATH > tmp.txt</code>
<code>cat</code>	display contents of a file <code>cat tmp.txt</code>
<code>less</code> , <code>more</code>	pagewise display (<code>less tmp.txt</code>)
<code>grep</code>	search for words/text (<code>grep result out.res</code>)
<code>ps</code>	display running processes (<code>ps -A</code>)
<code>kill</code>	kill a process (<code>kill -9 12813</code>)
<code>htop</code>	display table of processes (interactive per default)
<code>ssh</code>	secure shell to a remote machine (<code>ssh s1230815@login.zih.tu-dresden.de</code>)

Editors:

- ▶ `vi` - cryptic, non-intuitive, powerful, universal editor
- ▶ `emacs` - cryptic, non-intuitive, powerful, universal editor
- ▶ `mcedit` - intuitiver Editor (Kommandozeile)
- ▶ `gedit` , `kate` - grafische Editoren

Help at the command line



Every Linux command comes with detailed manual pages. The command `man <program>` is the first aid kit for Linux questions.

CHMOD(1) User Commands CHMOD(1)

NAME

`chmod` - change file mode bits

SYNOPSIS

```
chmod [OPTION]... MODE[,MODE]... FILE...  
chmod [OPTION]... OCTAL-MODE FILE...  
chmod [OPTION]... --reference=RFILE FILE...
```

DESCRIPTION

This manual page documents the GNU version of `chmod`. `chmod` changes the file mode bits of each given file according to mode, which can be either a symbolic representation of changes to make, or an octal number representing the bit pattern for the new mode bits.

The format of a symbolic mode is [ugoa,...][[+]=][perms,...], where perms is either zero or more letters from the set rwxst, or a single letter from the set ugo. Multiple symbolic modes can be given, separated by commas.

A combination of the letters ugoa controls which users' access to the file will be changed; the user who owns it (u), other users in the file's group (g), other users not in the file's group (o), or all users (a). If none of these are given, the effect is as if a were given, but bits that are set in

Manual page `chmod(1)` line 1

Linux file systems



- ▶ after mounting, remote file systems can be accessed like local resources.
- ▶ names are **case sensitiv**
- ▶ there is no “drive letter” like C:

Linux file systems



Typical directory structure:

- ▶ system programs in `/bin` , `/usr/bin`
- ▶ ...
- ▶ third party applications, libraries and tools, special software trees e.g.
 - ▶ normally in `/opt`
 - ▶ ZIH's HPC systems in `/sw`
- ▶ every user has her own home directory
 - ▶ `/home/<login>`
 - ▶ e.g. `/home/mark`

Special directories:

- ▶ `~` $\hat{=}$ home directory (`cd` or `cd $HOME`)
- ▶ `.` $\hat{=}$ current directory
- ▶ `..` $\hat{=}$ parent directory

File properties



Every file or directory has its access properties:

- ▶ 3 levels of access: **u**ser, **g**roup, **o**ther
- ▶ 3 properties per level: **r**ead, **w**rite, **x**ecute (for directories: execute = enter)
- ▶ list directory `ls -l .`

-wxwx-x	1 mark zih	9828	Apr 22 13:19	omp
-w-----	1 mark staff	521	Apr 22 13:19	omp.c
-w-----	1 mark zih	310288384	May 7 19:01	p1s055.30880.core
-w-----	1 mark root	116007687	Apr 12 12:56	pluk.tgz
dwxwx-x	4 mark staff	4096	Mar 18 16:44	projekte

dir/link
user
group
other

Default: User has all access rights in their `$HOME`-directory.

Change file properties



Usage of the `chmod` command:

- ▶ indirectly: Which access rights shall be added/removed
 - ▶ set a file readable for all: `chmod a+r readme.txt`
 - ▶ remove all rights for the group: `chmod g-rwx readme.txt`
- ▶ directly: `chmod 700 a.out` Consult the man page:
`man chmod`

Redirection of I/O



Linux is a text-oriented operating system. Input and output is 'streamable'.

- ▶ standard streams are: stdin, stdout, stderr
- ▶ streams can be redirected from/to files
e.g. `myprog <in.txt >out.txt`
- ▶ error messages (warnings) are separated from normal program output
e.g. `myprog 2>error.txt >out.txt`
- ▶ merge error messages and output:
`myprog 2>&1 out_err.txt`

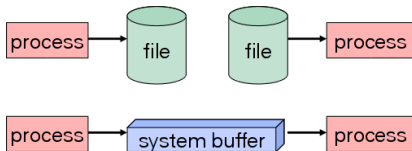
Attention:

The '`>`' operator will always empty an existing output file. For appending a stream output to a file use the '`>>`' operator. e.g. `myprog >>all_outs.txt`.

Command pipelines



Inputs and outputs can also be other programs.



```
ls -la | sort | more
```

```
echo 'Have fun!' | sed -s 's/fun/a break/g'
```

Versatility of Linux (and Linux like operating systems) comes from

- ▶ command line controlled program execution
- ▶ combining multiple programs in a pipelined execution
- ▶ mightful scripting, parsing, and little helper tools (shell, awk, sed, perl, grep, sort)

Vorspann

Linux

Bash

Praxis

Abspann

Ideen zum Üben



1. Verzeichnis `~/ese/bash-ws` anlegen und betreten
2. Aktuelles Arbeitsverzeichnis ausgeben
3. leere Datei `hallo.txt` anlegen
4. Datei `hallo.txt` mit Editor öffnen und "Hallo Welt" reinschreiben
5. Datei `hallo.txt` nach `hola.txt` kopieren
6. Datei `hallo.txt` nach `bonjour.txt` umbenennen
7. Hilfe-Seite von `mv` oder anderem Kommando anzeigen lassen
8. Alle laufenden Prozesse in Datei `processes.txt` ausgeben lassen
9. Zeilen von `processes.txt` zufällig sortieren (Tipp: `man sort`)
10. Script schreiben welches einen Namen erfragt und "Hallo <Name>" ausgibt (Tipp: Web-Suche nach "bash read stdin")
11. Script ausführbar machen und ausführen

Vorspann

Linux

Bash

Praxis

Abspann

Schlussbemerkungen (2): Blick über Tellerrand



Bits und Bäume



- ▶ Digitalisierung und Nachhaltigkeit verbinden
- ▶ Ressourcen, Freie Software, Privatsphäre, Demokratie, ...
- ▶ Größeres Treffen: 23.10., 19:00 Uhr, [Rosenwerk](#)
- ▶ <https://dresden.bits-und-baeume.org/>

Schlussbemerkungen (3): Blick über den Tellerrand



- ▶ TU-Umweltinitiative
 - ▶ Umweltringvorlesungen ab
 - ▶ Offenes Grillen/Essen: 24.10.
 - ▶ Wandern: 02.11.
 - ▶ <https://tuuwi.de/>

Umweltringvorlesungen
der TU Umweltinitiative
WiSe 19/20

TALKING ABOUT KLIMAWANDEL
Konfliktlinien auf dem Weg in eine postfossile Welt
Montag 7.10.
18.30 - 20.00 Uhr
Raum 22.31.18
HSC 471

Basiseminar GEWALTFREIE KOMMUNIKATION
nach M.B. Rosenberg
8.10. November 2019
(Wochentags-Veranstaltung)

WANDEL-GESCHICHTEN
Die Gesellschaft von morgen
Montag 8.10.
18.45 - 20.00 Uhr
Raum 22.31.18
HSC 471

THERES VO
PLANET B

Von Studierenden
Creditpunkte für AGos und Wahlpflichtbereich
Einschreibung und mehr Infos auf tuuwi.de

Schlussbemerkungen (3): Blick über den Tellerrand



- ▶ TU-Umweltinitiative
 - ▶ Umweltringvorlesungen ab
 - ▶ Offenes Grillen/Essen: 24.10.
 - ▶ Wandern: 02.11.
 - ▶ <https://tuuwi.de/>
- ▶ Students4Future:

Klimakrise, was tun?

Sei dabei und kämpfe für (d)eine nachhaltige Zukunft mit uns gemeinsam!

SEMESTERAUFTAKT

FREITAG, 25.10.

16:30 UHR IM BIO-BAU

(ZELLESCHER WEG 20B)



→ <https://ffdd.de>

Schlussbemerkungen (4)



- ▶ Fragen?
- ▶ **Unterstützung:** (im Rahmen unserer Möglichkeiten)
 - ▶ <https://fsfw-dresden.de/sprechstunde>
 - ▶ <https://fsfw-dresden.de/bash-ws>
 - ▶ kontakt@fsfw-dresden.de

Quellen und Links (Auswahl)



- ▶ `https://git-scm.com/documentation`
- ▶ `https://git-scm.com/documentation/external-links`
- ▶ `https://stackoverflow.com/questions/tagged/git`
- ▶ ...