



Linux und Bash: Warum und wie

Erstsemester-Einführung Informatik, 09.10.2019

Vorspann

Linux

Bash

Praxis

Abspann

<Eigenwerbung>

# FSFW (1)



## Wer sind wir?

- ▶ Hochschulgruppe (gegründet 2014, ca. 7 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 ( $\text{\LaTeX}$ / 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



[kontakt@fsfw-dresden.de](mailto:kontakt@fsfw-dresden.de)

</Eigenwerbung>



Vorspann

**Linux**

Bash

Praxis

Abspann

# Hinweis



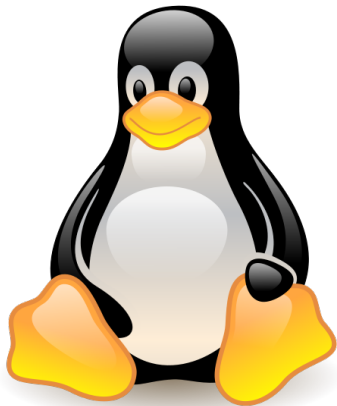
Die folgenden (engl.) Folien basieren zum großen Teil auf [Material](#) von Dr. Ulf Markwardt (High Performance Computing @ZIH).

Vielen Dank!

## General



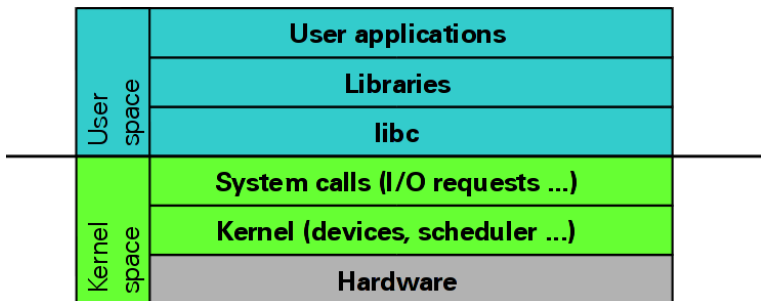
- ▶ first version 1991, Linus Torvalds
- ▶ hardware-independent operating system
- ▶ 'Linux' is the name of the kernel as well as of the whole operating system
- ▶ since 1993 under GNU public license (GNU/Linux)
- ▶ various distributions for all purposes (OpenSuSE, SLES, Ubuntu, Debian, Fedora, RedHat,...)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



# Linux architecture



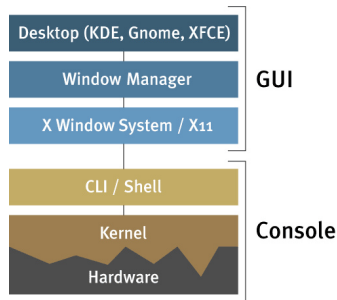
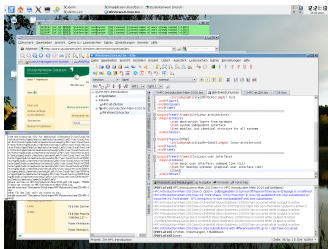
- ▶ abstraction layers from hardware
- ▶ system independent interface
- ▶ modular, but identical structure for all systems



# 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..)
  - ▶ LibreOffice (presentation, word processing, spread sheet)
  - ▶ 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`.

# Vorteile von Linux



1. Kontrolle über das System
2. Zugang zum Quellcode
3. Anpassbarkeit
4. Performant auf  
älterer/schwächerer  
Hardware (VMs)

# Vorteile von Linux



1. Kontrolle über das System
2. Zugang zum Quellcode
3. Anpassbarkeit
4. Performant auf älterer/schwächerer Hardware (VMs)
5. Stabilität (weniger Reboots)
6. Sehr gut für Entwicklung geeignet
7. Keine Lizenzkosten
8. Quasi keine Viren
9. Bessere Privatsphäre



# Vorteile von Linux



1. Kontrolle über das System
2. Zugang zum Quellcode
3. Anpassbarkeit
4. Performant auf älterer/schwächerer Hardware (VMs)
5. Stabilität (weniger Reboots)
6. Sehr gut für Entwicklung geeignet
7. Keine Lizenzkosten
8. Quasi keine Viren
9. Bessere Privatsphäre


Zitat Student 9. Semester:


*„Hätte ich früher gewusst, was Linux bietet, hätte ich viel Zeit und vor allem Nerven gespart.“*

Fazit: Linux-Kenntnisse empfehlenswert.

# zu Datenschutz und IT- Sicherheit




**TECHNISCHE UNIVERSITÄT DRESDEN**


**DRESDEN concept**

---

Der Prorektor für Universitätsentwicklung

Technische Universität Dresden, 01062 Dresden

Rektor, Prorektoren, Dekane/Dekaninnen, Sprecher/innen der Bereiche und Fachrichtungen, geschäftsf. Direktoren/ Direktorinnen bzw. Vorstand der Institute, Zentralen Einrichtungen und des Experimentellen Zentrums, Dezenternen/ Dezenterninnen, Sachgebietsleiter/innen, Personalrat, Studierendenrat, Gleichstellungsbeauftragte, Schwerbehindertenvertretung, Direktoren/Direktorinnen der Kliniken und Polikliniken des UKD, kaufm. Vorstand des UKD z. Ktn.,

Dresden, 04.09.2019

**Mitteilung des Prorektors für Universitätsentwicklung 3/2019**

**Einsatz von Windows 10**

**Sachwörter:** Datensicherheit (Windows 10)  
Datenschutz (Windows 10)  
Betriebssystem (Windows 10)

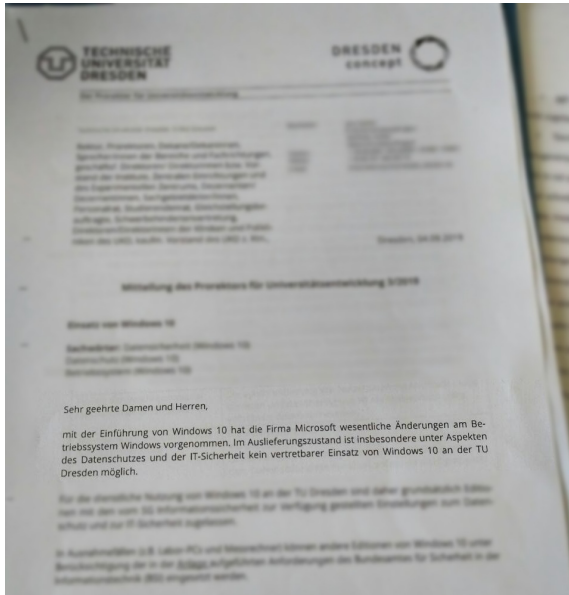
Sehr geehrte Damen und Herren,

mit der Einführung von Windows 10 hat die Firma Microsoft wesentliche Änderungen am Betriebssystem Windows vorgenommen. Im Auslieferungszustand ist insbesondere unter Aspekten des Datenschutzes und der IT-Sicherheit kein vertretbarer Einsatz von Windows 10 an der TU Dresden möglich.

Für die dienstliche Nutzung von Windows 10 an der TU Dresden sind daher grundsätzlich Editionen mit den vom SG Informationssicherheit zur Verfügung gestellten Einstellungen zum Datenschutz und zur IT-Sicherheit zugelassen.

In Ausnahmefällen (z.B. Labor-PCs und Messrechner) können andere Editionen von Windows 10 unter Berücksichtigung der in der Anlage aufgeführten Anforderungen des Bundesamtes für Sicherheit in der Informationstechnik (BSI) eingesetzt werden.

# zu Datenschutz und IT- Sicherheit



## Weitere Infos dazu



### Microsoft Dilemma (Problematisierung von Monopolstrukturen)

- ▶ Artikel <https://www.heise.de/.../Das-Microsoft-Dilemma-Windows-10-und-Office-in-Behoerden...>
- ▶ 45min Doku (ARD)  
[https://www.youtube.com/watch?v=\\_ZaDuinGf2o](https://www.youtube.com/watch?v=_ZaDuinGf2o)

## Weitere Infos dazu



### Microsoft Dilemma (Problematisierung von Monopolstrukturen)

- ▶ Artikel <https://www.heise.de/.../Das-Microsoft-Dilemma-Windows-10-und-Office-in-Behoerden...>
- ▶ 45min Doku (ARD)  
[https://www.youtube.com/watch?v=\\_ZaDuinGf2o](https://www.youtube.com/watch?v=_ZaDuinGf2o)

### Gegenmaßnahmen:

- ▶ Public Money Public Code: <https://vimeo.com/234308738>
- ▶ Mitmachen bei FSFW, FSFE, C3D2, ...
  
- ▶ Sich selbst unabhängig machen. → Freies OS verwenden

Vorspann

Linux

**Bash**

Praxis

Abspann

# Kommandozeile vs. Bash



(fast) Synonyme für *textbasierte Benutzerschnittstelle*

- ▶ Kommandozeile
- ▶ Terminal
- ▶ Konsole
- ▶ Shell

# Kommandozeile vs. Bash



(fast) Synonyme für *textbasierte Benutzerschnittstelle*

- ▶ Kommandozeile
- ▶ Terminal
- ▶ Konsole
- ▶ Shell

Sehr weit verbreitete:



(Skriptsprache und Shell-Software)



# Bash?



*“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



WIKIPEDIA  
The Free Encyclopedia

[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](#)

Search Wikipedia



## 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**

# Bash: Umgebungsvariablen



- ▶ Beispiele:

`PATH` search path for binaries

`HOME` path to user's home directory

- ▶ Setzen:

`foo="Hallo Welt"` oder `export foo="Hallo Welt"`

- ▶ Auswerten:

`echo $foo`

`echo $HOME`

`echo $PATH` (geordnete Suchpfad-Liste für ausführbare Programme)

## Skript:

```
#!/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 <b>Caution: No trash bin!</b>
<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 &gt; 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 &gt; 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, special software
  - ▶ in `/opt`
- ▶ every user has their own home directory
  - ▶ `/home/<login>`
  - ▶ e.g. `/home/s1230815`

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, **e**xecute  
(for directories: execute  $\hat{=}$  enter)
- ▶ list directory `ls -l .`

```

-rwxrwxr-x 1 mark zih      9828 Apr 22 13:19 omp
-rw-r--r-- 1 mark staff    521 Apr 22 13:19 omp.c
-rw-r--r-- 1 mark zih     310288384 May  7 19:01 pls055.30880.core
-rw-r--r-- 1 mark root     116007687 Apr 12 12:56 pluk.tgz
d-rwxr-xr-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 via 3 octal numbers: `chmod 700 a.out`
- See `man chmod` for details

# 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`

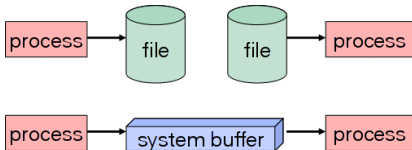
## Attention:

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 (| $\hat{=}$ “pipe”)



Inputs and outputs can also be other programs.



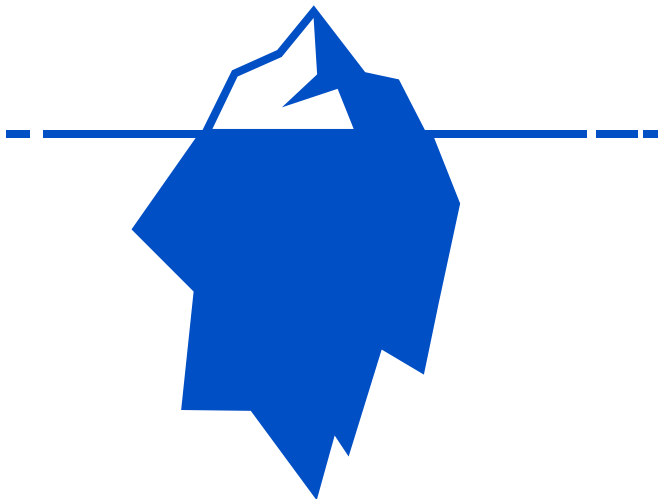
```
ls -la | sort | more
```

```
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)

Es gäbe noch viel mehr zu sagen...



Tipparbeit sparen mit Auto-Vervollständigung (TAB) und History (↑); Jobs beenden bzw. pausieren: STRG+C,

STRG+Z; Jobs im Hintergrund und Vordergrund `gedit xyz.txt &`, `bg 1`, `fg 1`, ...

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 (Credit Points)
  - ▶ 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  
Vendredi 7.09.  
18.00 - 20.00 Uhr  
Saal 22.03.18  
H22.001

**WANDEL- GESCHICHTEN**  
Die Gesellschaft von morgen  
Montag 8.09.  
18.00 - 19.30 Uhr  
Saal 22.03.18  
H22.001

**Basisseminar GEWALTFREIE KOMMUNIKATION**  
nach M.B. Rosenberg  
8.-13. November 2019  
(Wochenend-Seminar)

THERES NO PLANET B

Von Studio für Strauß  
Creditpoints für Adus und Wahlpflichtbereich  
Einschreibung und mehr Infos auf [tuuwi.de](https://tuuwi.de)

# Schlussbemerkungen (3): Blick über den Tellerrand



- ▶ TU-Umweltinitiative
  - ▶ Umweltringvorlesungen (Credit Points)
  - ▶ 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](mailto:kontakt@fsfw-dresden.de)

# Links



- ▶ <http://tldp.org/LDP/abs/html/part4.html> (Umfassende Kommando-Übersicht)
- ▶ <http://swcarpentry.github.io/shell-novice/>
- ▶ [https://wiki.ubuntuusers.de/Shell/Tipps\\_und\\_Tricks/](https://wiki.ubuntuusers.de/Shell/Tipps_und_Tricks/)
- ▶ ...

# Lösung zu Aufgabe 11



myscript.sh

```
#!/bin/bash

# Nutzer:innen-Name einlesen
read -r MYNAME

# gewünschten String ausgeben
echo "Hallo $;MYUSER"
```

- ▶ Skript ausführbar machen: `chmod +x myscript.sh`
- ▶ Skript ausführen: `./myscript.sh`