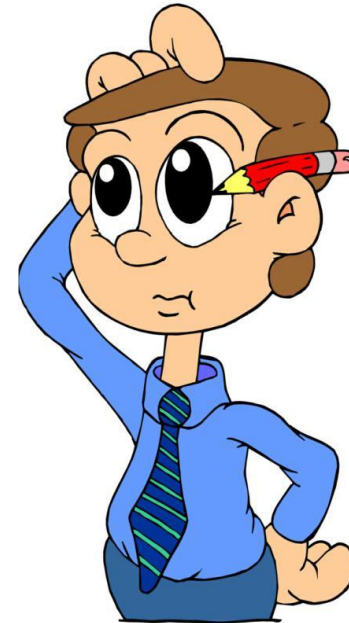
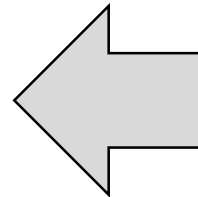


Bash Programmierung

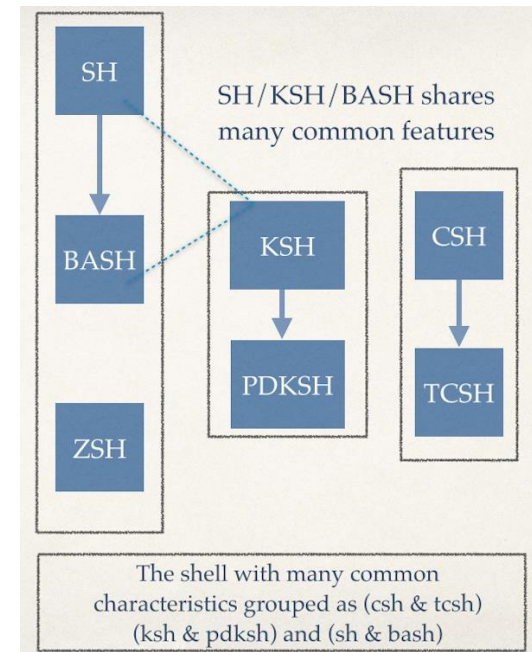


Shells

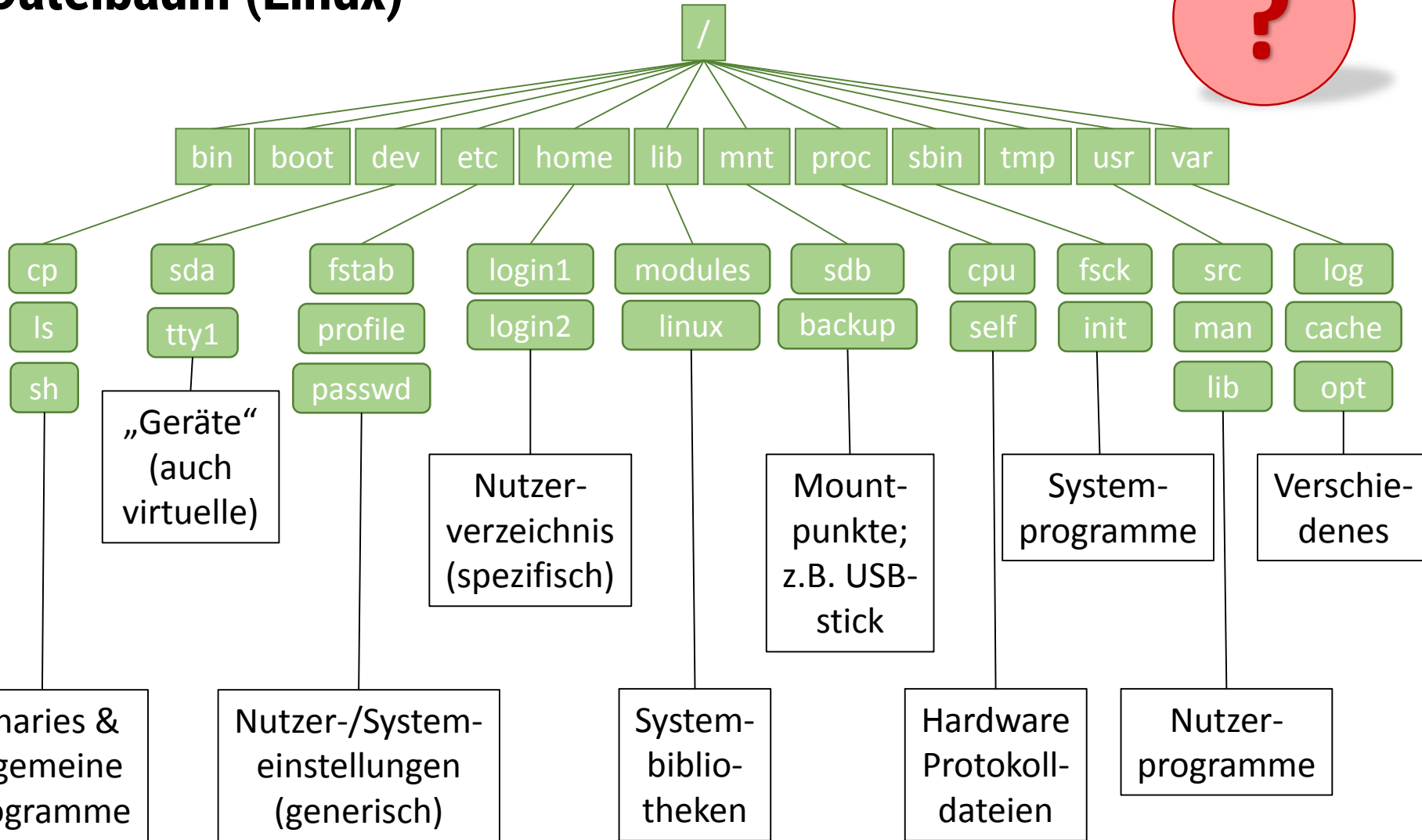
- https://en.wikipedia.org/wiki/List_of_command-line_interpreters

- Fokus hier: Bourne-again shell (Bash)
- Nützliche Links:

- <https://devhints.io/bash>
- https://www-user.tu-chemnitz.de/~hot/unix_linux_werkzeugkasten/bash.html



Dateibaum (Linux)



Shell und Linux-Programme

- Hinter jedem Befehl in der shell steckt ein Linux-Programm
- Befehle können Optionen (-a) und/oder Argumente aufnehmen
- Grundsätzliche *Systembefehle*:
 - `cd folder` # folder = Argument
 - `ls --color` # --color = Option
 - `cp a b` # a, b = Argumente
 - `ln -s a` # Option und Argument (erstellt einen symbolischen Link)
 - `mkdir ab` # Argument
- Hilfen
 - `www.google.com`
 - `man <Befehl>` (Beenden mit „q“)
 - `info <Befehl>`
 - `<Befehl> -h/-H/--help`



Praktischer Teil im Terminal

- Terminal öffnen:

Strg + Alt + T

- Neuer Tab im Terminal:

Strg + Shift + T

- Einen weiteren Terminal ausgehend von einem bestehenden:

Strg + Shift + N

Schleifen

- `For i in <LIST>; do ...; done`
 - `<LIST>` kann sein
 - Bash-expansion (`/path/to/files/*`)
 - Range (`{1..5}` oder `{1..10..2}`)
- `while <condition>; do ...; done`
 - `<condition>` kann jede bash-condition sein (`true`, `false`, `[?]`, `test ?`)
- `< file.txt | while read line; do echo $line; done`
 - Dateien zeilenweise auslesen (im Vergleich zu `cat`)

Weitere nützliche Werkzeuge

- sed
- awk
- top
- zip
- tar
- screen
- kill
- date
- time

Bash Texteditoren

- Vi(m)
- nano/pico
- emacs

- sed
- tr

Grafische Editoren

- Gedit (gnome)
- Mousepad (xfce)
- Kwrite (KDE)
- Atom, sublime

IDE

- Eclipse
- Visual Studio
- Spyder
- ...

Nützliche Hotkeys

- Befehl abbrechen: `Strg+C`
- Den letzten Befehls nochmal ausführen `!!`
- Das letzte Argument nochmals nutzen `!$`
- Vervollständigung in der Kommandozeile `Alt+Shift+*`
- Durchsuchen der Historie `Strg+R`
- Prozess anhalten `Strg+Z`
- Angehaltenen Prozess wieder fortsetzen `fg`
- Blockieren des Terminals `Strg+S`
- Fortführen des Terminals `Strg + Q`
- Schließen des Inputs `Strg+D`
- Den Terminal leeren `Strg + L` (auch über Befehl „clear“)

Und vieles mehr, siehe z.B. <https://ss64.com/bash/syntax-keyboard.html>

Reguläre Ausdrücke (regular expression, Regex)

- Definition von Suchmustern
- Anwendbar in vielen Programmiersprachen oder Texteditoren (z.B. Suchen und Ersetzen)
- Teils Unterschiede im Syntax
- Nützliche Links:
 - <https://www.rstudio.com/wp-content/uploads/2016/09/RegExCheatsheet.pdf>
 - <https://www.rexegg.com/regex-quickstart.html>
 - <https://remram44.github.io/regex-cheatsheet/regex.html>

Praktischer Teil im Terminal

- Terminal öffnen:

Strg + Alt + T

- Neuer Tab im Terminal:

Strg + Shift + T

- Einen weiteren Terminal ausgehend von einem bestehenden:

Strg + Shift + N

Beispiel: doi-extraction (.sh)

- Suchen eine doi aus einem pdf-file
- Definition einer DOI:
https://en.wikipedia.org/wiki/Digital_object_identifier
- Regex für doi:
`re_doi='10[.]\d{4,}[.][\d+][\/][?!.]'`
- Umwandlung pdf -> Text:
`pdftotext any.pdf`
- Nun doi extrahieren mit grep (any.txt sollte existieren)
`grep -oP "$re_doi" any.txt`
- Nun Abfrage Crossref nach .bib-string:
`getbib $doi >> bibfile.bib`

prefix	/	suffix
10.NNNN.N	/	anything

