

# Tutorat 1

## Einführung in Linux

# Übungsblatt

# Übungsblatt

## Kommandos im Zusammenhang mit Nutzern

- `w`, `who`: angemeldete Benutzer (`w`: momentaner Prozess und Laufzeiten)
- `whoami`: momentaner Benutzer
- `(finger`: Informationen zu lokalen und remote Nutzern)
- `id`: Benutzer-ID und die Gruppenmitgliedschaften des momentanen Benutzers
- `last -s yesterday`: zuletzt angemeldete Benutzer
- `df -ih`: eingehängte Dateisysteme (I-Node usage, humad readable)

## Ergänzungen

- `lsblk -f`: partition size (`-l`: list without the tree, `-f`: file system type)
- `lsblk -o NAME, FSTYPE, SIZE, MOUNTPOINT, LABEL`
- `sudo parted -l`, `sudo fdisk -l` (detailed)

# Übungsblatt

## Kommandos im Zusammenhang mit dem System

- `uptime`: wie lange Maschine schon läuft
- `date`: aktuelles Datum
  - `date '+Datum: %d.%m.%Y, Zeit: %H:%M:%S'`, `date +"Datum: %d.%m.%Y, Zeit: %T"`
- `(top: "Taskmanager")` → besser `htop`
- `hostname`: Hostname Maschine
- `free`: Menge verfügbarer freier Speicher, **Swap**: für sehr Arbeitsspeicherlastige Berechnungen)

## Ergänzungen

- `cat /etc/os-release`, `cat /etc/lsb-release` oder `lsb_release -a`: linux distro version (package lsb-release) oder am besten: `screenfetch` oder `neofetch`
- `cal -3`, `cal 2021`

# Übungsblatt

## Dateien Verzeichnisse 1 von 5

- `ls -alh`: Verzeichnisinhalt untereinander auf mit Zugriffsrechten, Nutzer, Gruppe, Dateigröße, Änderungsdatum, Namen usw. ( `h` für human readable, `a` für `.hidden_files`, `l` for a detailed listing with permissions)
- `cd path`: **changes** the shell's current working **directory**
  - only `cd`: changes to user's home directory (same as `cd ~`)
  - `cd ..`, `cd ../folder`, `cd folder`
- `cp file1 file2`: copy file1 to file2
  - `cp file1 ... fileN path`: **copy** a number of files to a directory
  - `cp ..`: one directory back
  - `./file` ist not needed, just `file`
- `mv file1 file2`: renames files
  - `mv file1 ... fileN path`: **move** a number of files to a different directory

# Übungsblatt

## Dateien Verzeichnisse 2 von 5

- `touch file1.txt file2.txt`: creates files
  - `touch *.txt`: overwrite files with empty files
  - in `ls -l` the date and time are when touch was run
- `rm file`: **remove** file
  - `(sudo) rm -r path`: delete file or folder at any place
  - `rm -rf path/dir`: to delete a directory and its contents, `-r` option specifies *recursive* delete to repeatedly delete everything inside dir, `-f` **forces** the delete
- `mkdir path/dir`: **makes** a new **directory** `dir`
  - `-p ~/dir1/dir2/dir3`: create fitting parent directories for `dir3`
  - `function mkcd() { mkdir $1; cd $1; }` in `~/.zshrc`, `~/.bashrc` uä.

# Übungsblatt

## Dateien Verzeichnisse 3 von 5

- `rmdir path/dir`: removes **dir**
  - if `path` isn't empty, this command fails, therefore use `rm -rf path/dir`
- `pwd`: (print working directory) simply outputs the name of the current working directory
  - `-P`: symbolic links can sometimes obscure the true full path of the current working directory
- `echo Hello World`: prints its arguments to the standard output
- `cat file1 file2`: outputs the contents of one or more files
  - is called cat because it performs concatenation
  - `cat > mail.txt` (with no filenames): input stream, exit with ctrl + d
- `tac`: like `cat` but other way round

# Übungsblatt

## Dateien Verzeichnisse 4 von 5

- `less`: page through a file, when a file is really big or when a command's output is long and scrolls off the top of the screen. See the contents of the file one screenful at a time
  - `<q>` to quit
  - enhanced version of an older program named `more`
  - `<spacebar>` to go forward in the file and the `<b>` key to skip back one screenful
  - `/word` to search forward, and `?word` to search backward. `<n>` to continue searching
- `file file`: system guesses format of file



# Übungsblatt

## Dateien Verzeichnisse 5 von 5

- `find dir -name file -print`: find file in dir
  - accepts special pattern-matching characters such as `*`, but you must enclose them in single quotes ( `'*'` ) to protect the special characters from the shell's own globbing feature
- `cat ./index.md | wc -l / -w / -m / -c`: count output (line, words, chars, bytes)
  - `find /usr/share/doc -name "*.pdf" | wc -l`
- `head file` and `tail file`: `head file` shows the first 10 lines of a file, `tail` shows the last 10 lines
  - `-n`: change the number of lines to display
  - `tail +n`: print lines starting at line n
  - `grep ie /usr/share/dict/words | less`: sending the output of a grep command to less

# Übungsblatt

## Ergänzungen 1 von 2

- `vifm`
- `grep word /path/file`: prints the lines from a file or input stream that match an expression
  - `grep word ~/dir/*`: check every file in `~/dir` that contains a word
  - `cat file1 file2 | grep -i word`: together with `cat` (but "don't cat into grep")
  - `-i`: for case-insensitive matches
  - `-v`: inverts the search, prints all lines that don't match
  - `egrep` just a synonym for `grep -E`, understands patterns known as regular expressions
- `fzf`, `rg` (ripgrep), `fd`
- `pdftopdf -in <schlagword> *.pdf`
- `which tesseract`, `whereis vim`: show path to the executable

# Übungsblatt

## Ergänzungen 2 von 2

- `file img.png`
- `clear` or `ctrl + l`, `reset`, `exit`
- `history`
- `tr [OPTION]... SET1 [SET2]`
  - `echo abcdefg | tr a-z A-Z`: translates characteres from SET1 to SET2
  - `echo abcefg | tr -d a-c`: deletetes characters ( `--delete` )
- `diff file1 file2`: differences between two text files
- `sudo -s`, get out of it with `exit`: no `sudo` each time (until timer runs out)
- `poweroff`, `shutdown now`: turn off (only `shutdown` will start a counter)
- `reboot`, `init 6`: restart

# Vertiefung

# Vertiefung

## Special Symbols

- `!vim` or `!<history-number>`: get last typed command starting with "vim" or by number in history with the `history`-command
- `sudo !!`: run last command as sudo (if one forgot typing sudo)
- `xed file.txt &`: open application in background
- `sudo apt update -y && sudo apt upgrade -y`: serveral commands and only execute if last one worked
- `cd; ls -l`: several commands, execute one after another
- **Pipe:** `cat ./index.md | wc -l / -w / -m / -c`

# Vertiefung

## Shell Globbing (Wildcards)

- `*`: glob character, which tells the shell to match any number of arbitrary characters and substitutes all matching filenames, called expansion
- `?` glob character, instructs the shell to match exactly one arbitrary character
- `'*'`: don't expand glob character
- problems with globs because `.*` matches `.` and `..`
- `.[^.]*` or `.??*`: get all dot files except the current and parent directories

# Vertiefung

## Shell

- program that runs commands the user enters
- Bourne **Sh**ell: a standard shell in `/bin/sh` developed at Bell Labs for early versions of Unix. Every Unix system needs the Bourne shell in order to function correctly
- **B**ourne-**a**gain **Sh**ell: enhanced version of the Bourne shell in `/bin/bash`, `/bin/sh` is normally a link to bash
- `$` for normal user and `#` for superuser / root user

# Vertiefung

## Vifm 1 von 3

- select multiple files with `v` and hit `enter`, `av` to continue selection again without replacing it, hit `t` to select single files, `esc` to undo everything again
- `cw`: to rename directory or file
- `:marks` to view all bookmarks, new bookmarki with `m + <any_key>` and `' + <any_key>` (h ist already set to the home directory by default)
- `za`: toogle invisible files
- `:sync`: make both windows open the same path
- `shift+s` or `:sort` and then type right letter from menu: show files in directory sorted after...
- `gA`: calculate filesize of folder
- `w` toggle file preview or `:view`



# Vertiefung

## Vim 2 von 3

- vim `D` für unwiederuflich löschen und `P` für ausschneiden und nicht nur kopieren
- vim `<C-w>w` or `shift+tab`, it switches to the other pane into to scroll up and down in file preview, exit with `q`
- `tab` switches with the other pane
- `!` and `!!`
- `:trashes?` same as `:trashes`, but also displays size of each trash directory
- `:trashes` list all valid trash directories in a menu. Only non-empty and writable trash directories are shown. This is exactly the list of directories that are cleared when vim- `:empty` command is executed.
- `:tree` for tree view

# Vertiefung

## Vim 3 von 3

- **open with:** `:f[ile]`, will open list to choose from with what to open file or files under cursor or selected files
- `=` for search-filtering
- vim `ctrl + g` or `q:`, `q/`, `q?`, `q=`, leave with `:q`
- vim quickstart: [https://wiki.vim.info/index.php/Quickstart\\_Tutorial](https://wiki.vim.info/index.php/Quickstart_Tutorial)
- `~/.config/vim/vimrc`:

```
" yank current directory path into the clipboard
nnoremap yd :!echo -n %d | xclip -selection clipboard %i<cr>:echo expand('%"d') "is yanked to clipboard"<cr>

" yank current file path into the clipboard
nnoremap yf :!echo -n %c:p | xclip -selection clipboard %i<cr>:echo expand('%"c:p') "is yanked to clipboard"<cr>

" yank current filename without path into the clipboard
nnoremap yn :!echo -n %c | xclip -selection clipboard %i<cr>:echo expand('%"c') "is yanked to clipboard"<cr>
```

# Vertiefung

## Kleine Spielerreihen

- `wget -A pdf,mp4 -m -p -E -k -K -np <url>`
  - `wget -A pdf -m -p -E -k -K -np --user=***** --password=***** <url>`
- `curl wttr.in/Berlin`, `curl ifconfig.co (IP ADDRESS)`, `curl ifconfig.co/city` or `country`, `curl -s http://tinyurl.com/api-create.php?url...`, `curl cheat.sh/btrfs (CHEAT SHEETS)`, `curl parrot.live`
- `tty-clock -c -C 1`
- `watch -n1 "date '+%D%n%T'|figlet -k", figlet -f block, ls /usr/share/figlet`
- `cmatrix`, `pipes ( sudo pacman -S bash-pipes )`
- `fortune | cowsay -f $(ls /usr/share/cowsay/cows | shuf -n 1) | lolcat`, `( -1 )`
- `youtube-dl -x --audio-format mp3 <url>`
- `mocp`, `neomutt`, `tizonia`
- `sudo apt install ascii`

# Vertiefung

## Hilfe zur Selbsthilfe 1 von 2

- `man command`: show **man**ual page to a command
  - `-k`: if the exact name of the command is unknown
  - often same as `comand --help` or `command -h`
  - manual pages are referenced by numbered sections: e.g. `man 5 passwd`

Section	Description
1	User commands
2	System calls
3	Higher-level Unix programming library documentation
4	Device interface and driver information
5	File descriptions (system configuration files)
6	Games
7	File formats, conventions, and encodings (ASCII, suffixes, and so on)
8	System commands and servers

# Vertiefung

## Hilfe zur Selbsthilfe 2 von 2

- `info` command: GNU Project decided that it didn't like manual pages very much and switched to another format called `info` (or `texinfo`)
- `curl cheat.sh/topic`
  - e.g. `curl cheat.sh/unziunzip`
- `tldr find`: easier to read manual
- some packages dump their available documentation into `/usr/share/doc`

# Vielen Dank für eure Aufmerksamkeit!

