CLI tools

- find, search, grep
- Cronjob, tar, autostart, sudo, ln
- CLI Editor
- user management
- disk/file management
- tmux
- basic

find, search, grep

find

use cases:

- `find ./subfolder -name sample.txt` - Search a file with specific name
- `find ./subfolder -name *.txt` - Search a file with pattern
- `find ./GFG -name sample.txt -exec rm -i {} \;` - find and delete a file with confirmation
- `find ./subfolder -empty` - Search for empty files and directories
- `find ./subfolder -perm 664` - Search for file with entered permissions
- `find ./ -type f -name "*.txt" -exec grep 'some Phrase'  {} \;` - Search text within multiple files

more examples

```
find /path -name *.txt
find /path -type f -name test.txt
find /path -name failed*.* -type f
find /path -type f -not -name "*.html"
find / -name "file.txt" -size +4M
find /dev/ -type b -name "sda*"
find ./*file*
```

grep

`grep [OPTION]... PATTERNS [FILE]...`

- options
  - `-B <numb>` - show numb lines before match
  - `-A <numb>` - show numb lines after match
  - `-i` - ignore case distinctions in patterns and data
  - `-r`, `--recursive` - like --directories=recurse
  - `-v`, `--invert-match` - To display only the lines that do not match a search pattern
  - `--exclude-dir=<foldername>` - exclude folder from search

    - `-n`, `--line-number` - Prefix each line of output with the 1-based line number within its input file.

examples

- `grep -ir --exclude-dir=vendor skeleton .` - find all occurences of "skeleton" in the current working dir
- `grep -i "some string" path/**/files.log` - search string in log files

sed

`sed -i 's/SEARCH_REGEX/REPLACEMENT/g' INPUTFILE`

- `-i` - By default, sed writes its output to the standard output. This option tells sed to edit files in place. If an extension is supplied (ex -i.bak), a backup of the original file is created.
- `s` - The substitute command, probably the most used command in sed.
- `/ / /` - Delimiter character. It can be any character but usually the slash (/) character is used.
- `SEARCH_REGEX` - Normal string or a regular expression to search for.
- `REPLACEMENT` - The replacement string.
- `g` - Global replacement flag. By default, sed reads the file line by line and changes only the first occurrence of the SEARCH_REGEX on a line. When the replacement flag is provided, all occurrences are replaced.
- `INPUTFILE` - The name of the file on which you want to run the command.

Cronjob, tar, autostart, sudo, ln

cronjob

- execute all listed cronjobs: `crontab -l | grep -v '^#' | cut -f 6- -d ' ' | while read CMD; do eval $CMD; done`
- send output to mail address
  - `MAILTO="empfänger@adresse.de"`
  - https://de.godaddy.com/blog/behalten-sie-ihre-cron-jobs-unter-kontrolle/

tar

useage: `tar [OPTION...] [FILE]...`

- options:
  - `-c` Ein neues Archiv erzeugen.
  - `-d` Dateien im Archiv und im Dateisystem miteinander vergleichen.
  - `-f` Archiv in angegebene Datei schreiben / Daten aus angegebener Datei lesen. Diese Option muss die letzte sein, da die nachfolgende Zeichen als Datei interpretiert werden. Z.B. würde -cfv zu einer Fehlermeldung führen. Korrekt wäre -vcf.
  - `-x` Dateien aus einem Archiv extrahieren.
- create archive:
  - unzipped
    - `tar -czf archiv.tar.gz file` Create archiv.tar.gz
  - zipped
    - `tar -cf archiv.tar path/` Create archiv.tar with all subdir
    - `tar -cf archiv.tar datei_1.txt *.pdf` Create archiv.tar with 1 file and all pdfs
- anhängen
  - `tar -rf archiv.tar datei_1.txt` Add 1 specific file to archiv (unconpressed)
- extract archive: ACHTUNG - tar überschreibt existierende automatisch
  - `tar -xf archiv.tar` Extract files
  - `tar -xzf archiv.tar.gz -C /` Extract gzip archives
  - `tar -xzf archiv.tar.gz -C /path` Extract gzip archives to /path
- anzeigen `tar -tvf archiv.tar` Show which files are in archiv.tar

autostart

- create new file
  - user only `sudo nano ~/.config/autostart/<some_name>.desktop`
  - global `sudo nano /etc/xdg/autostart/<some_name>.desktop`
- insert content like:

```
[Desktop Entry]
Type=Application
Name=Musterprogramm
Exec=Auszuführendes Kommando
```

sudo

add user to sudo

```
su -
usermod -aG sudo <username>
```

no password sudo

- edit sudoers file with `sudo visudo`
- add line at **end of file** (important for not be overriten):

```
username  ALL=(ALL) NOPASSWD:ALL
```

links

- `ln -s /Pfad/zur/Datei /Pfad/zum/symlink`
  - `-s` (symbolic) erstellt einen symbolischen Link statt eines Hardlinks.
  - `-f` (force) aktualisiert den Link und entfernt existierende Ziele.
  - `-i` (interactive) fragt nach, bevor Ziele entfernt werden (setzt -s voraus).
  - `-r` (relative) erstellt symbolische Links relativ zum Link-Speicherort.

CLI Editor

nano

cli parameters/flags

- `-c`, `--const` - Constantly display the cursor position and line number on the statusbar. Note that this overrides -U.
- `-U`, `--quickblank` - Do quick statusbar blanking. Statusbar messages will disappear after 1 keystroke instead of 25. Note that -c overrides this.
- `-S`, `--smooth` - Enable smooth scrolling. Text will scroll line-by-line, instead of the usual chunk-by-chunk behavior.
- `-m`, `--mouse` - Enable mouse support, if available for your system. When enabled, mouse clicks can be used to place the cursor, set the mark (with a double click), and execute shortcuts. The mouse will work in the X Window System, and on the console when gpm is running.

set defaults by editing `/etc/nanorc` or in `~/.nanorc`: (some recommendations)

```
## Constantly display the cursor position in the status bar or minibar.
set constantshow

## Display line numbers to the left (and any anchors in the margin).
set linenumbers

## Enable mouse support, if available for your system.  When enabled,
## mouse clicks can be used to place the cursor, set the mark (with a
## double click), and execute shortcuts.  The mouse will work in the
## X Window System, and on the console when gpm is running.
set mouse

## Make the Home key smarter: when Home is pressed anywhere but at the
## very beginning of non-whitespace characters on a line, the cursor
## will jump to that beginning (either forwards or backwards).  If the
## cursor is already at that position, it will jump to the true start
## of the line (the left edge).
```

```
# set smarthome

## Spread overlong lines over multiple screen lines.
# set softwrap
```

userful short cuts

shortcuts are displayed on the bottom, maximize to see more

- `Ctrl` + `k` cut selection or current line
- `Ctrl` + `u` paste at cursor possition
-

vim

tutorials

- VimTutor
- https://danielmiessler.com/study/vim/

vim modes

- command mode (default)
  You can switch to any mode from this mode. You can't do this in any other modes. To switch from one mode to another, you have to come to Command Mode first and then navigate to the other mode. The commands that you run without any prefix (colon) indicate that you're running the command in command mode.
- `i` - insert mode
  This mode is used to edit the contents of the file. You can switch to insert mode by pressing `i` from command mode. You can use the `Esc` key to switch back to command mode.
- `:` - command-line mode
  You can use this mode to play around with some commands. But the commands in this mode are prefixed with a colon (`:`). You can switch to this mode by pressing `:` (colon) in command mode.
- `v` - visual mode
  You use this mode to visually select some text and run commands over that section of code. You can switch to this mode by pressing v from the command mode.

base commands

- create/edit and save/quit
  - `:edit sample.txt` create/edit (new file)
  - `:w` - save and continue
  - `:q` - quit
  - `:q!` - quit without save
  - `:wq` - save and quit
- modes
  - `i` - insert
  - `v` - visual
  - `ESC` - return to command mode (default start point)
- undo/redo
  - `u` - undo
  - `U` - undo a whole line
  - `3u` - undo 3 times
  - `:undolist` - list available undo options
  - CTRL + `R` - redo

movement

- general

```
          ^
          k□□    Hint:  The h key is at the left and moves left.
     < h    l >□□   The l key is at the right and moves right.
          j□□□   The j key looks like a down arrow.
          v
```

- jump to lines
  - `gg` - first line
  - `G` - end of file
  - `42G` - to line 42
- counted motions
  - `2w` move cursor to words forward
  - `3e` move cursor to the end of the third word
  - `0` move to line start

text editing

- motion list
  - `w` - until the start of the next word, EXCLUDING its first character.

    - `e` - to the end of the current word, INCLUDING the last character.
    - `$` - to the end of the line, INCLUDING the last character.
- inserting
    - `i` - insert before cursor
    - `A` - append after the line
- deletion
    - `x` - delte current character
    - `dw` - delete from cursor up to the next word
    - `de` - delete from cursor up to the end of the current word
    - `d$` - delete from cursor to the end of the line
    - `d0` - delete to the beginning of line
    - `dd` - delete whole line
    - `d2w` - delete two words
    - `d2d` - delete two lines
- copy/cut/paste
    - `y` - copy selected textpaste
    - `yy` - copy current line
    - `y5y` - copy 5 lines
    - `d` - cut selected text
    - `p` - paste copied text
    - `:t.` - duplicate current line
    - `yy`, `p` - duplicate current line

find and replace

searching only

- `/` search forward (forward slash)
- `?` search backward
- case insensitiv: append `\c`

example: `/Linux\c` + Enter

- after first search press `n` to search for the next occurrence or uppercase `N` to search backwards

find and replace

`:[range]s/{pattern}/{string}/[flags]`

- `[range]` indicates that you can pass the range of lines. The range is separated by a comma.
    - Pass `%` to find and replace in all lines
    - `5,10` To find and replace between lines 5 to 10
    - `.` current line

    - `$` the last line of the file
- `{pattern}` indicates the pattern to find the text. You can pass regex patterns here.
- `{string}` is the string to replace in the found text.
- `[flags]` indicates if you wish to pass any additional flags. By default, this does a case-sensitive search.
    - `i` case-insensitive search
    - `c` confirm before replacing
    - `g` indicates making the change globally

example

`:2,3s/Hi/Hello and Welcome/gci` -> replace with Hello and Welcome (y/n/a/q/l/^E/^Y)?

- `y` - Replace the match
- `n` - Skip the match
- `a` - Substitutes the match and all remaining occurrences of the match
- `q` or `Esc` - Quit substitution
- `l` - Replace the match and quit
- `CTRL+Y` - Scroll the screen down
- `CTRL+E` - Scroll the screen up

user management

create user

- `useradd john`
- set password `passwd john`
- create user with root privilages: `useradd -ou 0 -g 0 john`
  - check if id are set correctly:

    ```
    $ grep john /etc/passwd
    john:x:0:0::/home/john:/bin/sh
    ```

  - add user to root group `usermod -a -G root john`
- delete user `userdel john`

disk/file management

disk

show disk size

- check disk space: `df`
  - `-h` - human readable

show folder size

- show folder size: `sudo du -shc ./*`

find biggest folders of current directory

`du -hs * | sort -rh | head -5`

file movements

rsync

Links

- https://wiki.ubuntuusers.de/rsync/

options

- `a` - contains **highly recommended**
  - `-r` copy subfolder
  - `-l` copy symbolic links
  - `-p` keep permissions of source files
  - `-t` keep timestamps of source files
  - `-g` keep group permissions of source files
  - `-o` keep user permissions of source files (only root)

    - `-D` keep device permissions of source files (only root)
- `v` - verbose
- `h` - human readable
- `z` - with compression
- `-e` - remote shell auswählen
    - `-e 'ssh -p 222'` ändert ssh port auf e.g. 222

local to remote server

```
rsync [option] [source] user@hostname-or-ip:[destination path]
rsync -avhz /home/source-rsync/ user@your-remote-server.com:/home/dest-rsync/
```

remote to local server

```
rsync -avhz user@your-remote-server.com:/home/dest-rsync/ /home/source-rsync/
```

scp

`scp <Optionen> <Quellpfad> <Zielpfad>`

- Quellpfad: `<Nutzer>@<Host>:<Verzeichnis/Datei.Endung>`
- optionen:
    - `-C` Kompression nutzen (compression)
    - `-p` Datei-Attribute auf Zieldateien übertragen (permissions)
    - `-r` Verzeichnisse rekursiv kopieren (recursive)
    - `-v` Erweiterte Ausgabe anzeigen (verbose)
    - `-q` Ausgabe unterdrücken (quiet)
    - `-3` Daten durch lokales System senden (third party)

example:

```
scp -r srv01-local_draab:/var/lib/docker/volumes/wireguard_wireguard-config/_data/
/home/danielraab/wireguard/
```

tmux

links

tutorial cheatsheet

configs

edit file `~/.tmux.conf` as you like

-> gpakosz github

-

keyboard shortcuts

`Ctrl + b` `?` - show keyboard shortcut overview

pane

- `Ctrl+b "` — split pane horizontally.
- `Ctrl+b %` — split pane vertically.
- `Ctrl+b arrow key` — switch pane.
- `Ctrl+b z` - maximize pane (and undo maximize)
- Hold `Ctrl+b`, don't release it and hold one of the `arrow keys` — resize pane.

windows

- `Ctrl+b c` — (c)reate a new window.
- `Ctrl+b w` — List all windows.
- `Ctrl+b n` — move to the (n)ext window.
- `Ctrl+b p` — move to the (p)revious window.

modes

CTRL+b then [Enter copy mode. q Exit copy mode. SPACE Start text selection in copy mode. ENTER Copy the selected text. ESC Clear the selected text and exit the copy mode. CTRL+b then] Paste the copied text. h Move the cursor left. j Move the cursor down. k Move the cursor up. l Move the cursor right. w Move the cursor one word forward. b Move the cursor one word backward.

CTRL+u Scroll up half a page. CTRL+d Scroll down half a page. PgUp Scroll up full page. PgDn Scroll down full page.

configs

adaption for zsh powershell10k

Your tmux has two issues.

It cannot display 256 colors. To fix this, create ~/.tmux.conf with set -g default-terminal screen-256color in it and reboot your machine.

It cannot display non-ascii characters. To fix this, install and enable a UTF-8 locale in your OS and reboot your machine.

You can verify that you've fixed tmux by running the following command:

```
print -P -- '--> %F{70}\u276E\u276F%f <--'
```

basic

suspend and continue

- `Ctrl - z` suspend
- command `fg` to continue