Linux

General

Shell

- install and config **zsh**
  - LINK
- create new file

  `touch <filename>`
- create a range of files
  - example:

    `touch myfiles{1..10}.txt`
- `grep` one of the most practical linux commands for filtering:
  - example:

    `ls | grep <wanted-folder>`

    you have very very practical flag `-v` that **exclude** files.
- how to see current user in cli

  `whoami`
- recommended to don't install snap on linux!
- just try download the file from the internet.
- how to see the hostname

  `hostname`
- how to see the hostname ip in LAN

  `hostname -I`

  first ip is your ipv4 in LAN
- how to see the current port usuage:

  `sudo lsof -i -P -n | grep LISTEN`

  **** does not contains hidden files!!!***
- add `alias` permanently in one command

  ```
  echo "alias aliasname='aliascommand'" >> ~/.bashrc && source ~/.bashrc
  echo "alias aliasname='aliascommand'" >> ~/.zshrc && source ~/.zshrc
  ```
- one of the fastest way to define your commands as shortcut is to use from `alias`es, even in your running project!

  - in your project folder
    * create a file like name `setup.sh`
    * define your function
      ```
      yourname() {
          yourcommand
      }
      ```
      · you can use from input arguments by `$1`, `$2`, ...
      · you can call the function like `yourname firstarg secondarg`
    * now use from your `command1`, `command2` easily

- add app to path

  ```
  echo 'export PATH=/your/directory/path:$PATH' >> ~/.bashrc && source ~/.bashrc
  ```

- see the disk usage useful commands

  ```
  df -h yourpath
  df -h /
  du -shx * | sort -rh | head -10
  sudo du -h --max-depth=1
  ```

- see the partitions

  ```
  lsblk
  ```

- go to your flash removable

  ```
  cd /run/<username>/<medianame>
  ```

- find in file contents

  ```
  grep -rnw yourpath -e "yourregex"
  ```

- go to terminal before login to linux shortcut:
  - `ctrl + alt + f2`
  - for back to graphical:
    * `ctrl + alt + f1` (centos worked)
    * `ctrl + alt + f7` (mint worked)

- how to run `.jar` file

  ```
  java -jar yourjarfile.jar
  ```

- network manager setting from terminal with ui

  ```
  nmtui
  ```

- see which ports are used!

  ```
  netstat -tunlp
  ```

- see the porxies

```
$ env | grep -i proxy
NO_PROXY=localhost,127.0.0.0/8,127.0.1.1
http_proxy=http://192.168.1.250:8080/
FTP_PROXY=ftp://192.168.1.250:8080/
ftp_proxy=ftp://192.168.1.250:8080/
all_proxy=socks://192.168.1.250:8080/
ALL_PROXY=socks://192.168.1.250:8080/
HTTPS_PROXY=https://192.168.1.250:8080/
https_proxy=https://192.168.1.250:8080/
no_proxy=localhost,127.0.0.0/8,127.0.1.1
HTTP_PROXY=http://192.168.1.250:8080/
```

- see port status

  ```
  nmap -p 5432 172.19.0.3
  ```

- remove a file recursively

  ```
  find . -name "FILENAMEWILDCARD" -delete
  ```

- increase the time login time

  ```
  sudo visudo
  ```

  ```
  Defaults        timestamp_timeout=240 # in minutes (4 hours)
  ```

- running and background and also saving logs!

  ```
  yourcommand >output.log 2>&1 &
  ```

- see your background procceses

  ```
  jobs
  ```

  you can have the latest runned background job in the session with `$i` command!

- The `+` symbol indicates the current job. This is the job that will be affected by commands like fg or bg if you don't specify a job ID.
- The `-` symbol indicates the previous job (the one before the current job).
- Other jobs do not have a symbol.
- Running
  - The job is actively running in the background.
- Stopped
  - The job has been paused (e.g., with Ctrl+Z).
- Done
  - The job has completed execution.
- brings job [1] to the foreground.

`fg %1`

- resumes job [1] in the background.

`bg %1`

- kills job [1].

`kill %1`

- to beautify the `json` format

  `jsonoutputer | jq .`

- see log online

  `tail -f output.log`

- find and replace all command from out of text file

  `sed 's/pattern/replacement/g' file`

  `sed -i 's/pattern/replacement/g' file`

  `sed -i.bak 's/pattern/replacement/g' file`

- just a process to up system

  `sleep 600 # 10 minutes`

- run a command periodic to see the result realtime

  `watch -n 0.5 your_command`

- set static ip

  `sudo nmcli con mod <connection-name> ipv4.method manual ipv4.addresses 192.168.56.100/2`

- see network interfaces

  `ip a`

- see network connections

  `nmcli con show`

systemd

- start the service

`systemctl start yourservice.service`

- stop the service

`systemctl stop yourservice.service`

- restart the service

`systemctl restart yourservice.service`

- enable the service to run after boot

```
systemctl enable yourservice.service
```

- diable the service to run after boot

```
systemctl diable yourservice.service
```

ssh

- get a file from remote

  ```
  scp remoteuser@remotehostip:/path/to/file localdirectory
  ```

- get a folder complete from remote

  ```
  scp -r remoteuser@remotehostip:/path/to/directory localdirectory
  ```

- send a file to destination ip

  ```
  scp filepath remoteuser@remotehostip:/remote/to/directory
  ```

- send a folder to destination ip

  ```
  scp -r folderpath remoteuser@remotehostip:/remote/to/directory
  ```

- copy files with progress status

  ```
  rsync -ah --progress source destination
  ```

- sha2 hash of string

  ```
  echo -n "your text" | sha256sum
  ```

- send DHCP

  ```
  sudo dhclient -r
  sudo dhclient
  ```

- see all process

  ```
  ps aux | grep whatyouwant
  ```

- kill process

  ```
  sudo kill -9 pid
  ```

- see app for port usage

  ```
  sudo lsof -i :5432
  ```

- change date

  ```
  date -s "date_output_like"
  ```

- while linux is up it should create a swap in size of your ram, if it can't (for example for having full disk drive), the system does not goes up, (if you can)decrease the amount of ram for solving from problem temporary!

Snap

- installing

  ```
  sudo snap install <your-package>
  ```

- removing

  ```
  sudo snap remove <your-package>
  ```

Security

File Encryption

GPG (GNUPG)

- generate keys

  ```
  gpg --gen-key
  ```

- see your keys

  ```
  gpg --list-secret-keys
  ```

- export your keys

  ```
  gpg --export-secret-keys [key-id | email] > private_key.asc
  ```

- import your keys

  ```
  gpg --import private_key.asc
  ```

- delete a key

  ```
  gpg --delete-key [KeyID]
  gpg --delete-secret-key [KeyID]
  ```

- symmetric encryption with just passphrase! (***very practical***)

  ```
  gpg -c --no-symkey-cache <filename>
  ```

- assymetric encryption with pair key

  ```
  gpg -e -r <key-email> <filename>
  ```

- now decrypt symmetric

  ```
  gpg -d <filename>.gpg > <filename>
  ```

GUI's

Cinammon

- file manager: nemo

Xfce

apps

- file manager: thunar
- disk usage analyzer: baobab
- pdf reader: xreader
- text editor: xed

tricks

- zooming (in normal mode :))
  - `alt` + `scroll`

Gnome

- file manager: nautilus
- disk usage analyzer: baobab
- pdf reader: xdg-open
- text editor: gedit

Distros

Debian

- install `.deb` package

  ```
  sudo dpkg -i <packagepath>
  ```

- remove `.deb` package

  ```
  sudo dpkg -r <packagepath>
  ```

Mint

- mint has has no default root password!, you should set for it

  ```
  sudo passwd root
  ```

- install `.deb` package

  ```
  sudo dpkg -i <packagepath>
  ```

- remove `.deb` package

  ```
  sudo dpkg -r <packagepath>
  ```

- for becoming root user

  ```
  sudo su
  ```

fedora

- see the all installed Packages

```
rpm -aq
```

- download your rpm with **DEPENDENCIES**

```
yumdownloader --resolve <package_name>
```

- export `rpm` file.

```
sudo yum install rpmrebuild
```

- create `rpm`

```
rpmrebuild your_app_name
```

centos

- if the root disk usage become 100% your operating system does not boot completely!!!
  - you probably go to your virtual terminal with `ctrl+alt+f2` shortcut to remove some of your data for booting completely!

Apps and Packages

ncat

- a package for general-purpose command-line tool for reading, writing, redirecting and encrypting data across a network.
- run server

  ```
  nc -lv <port>
  ```

- sending data to server

  ```
  nc -v localhost <port>
  ```

Firefox

- open app

  ```
  firefox
  ```

Google Chrome

- open app

  ```
  google-chrome
  ```