
Contents

1	Linux	1
1.1	General	2
1.1.1	Shell	2
1.1.2	systemd	7
1.1.3	ssh	7
1.1.4	Snap	8
1.1.5	Security	8
1.2	GUI's	9
1.2.1	Cinamon	9
1.2.2	Xfce	9
1.2.3	Gnome	10
1.2.4	KDE	10
1.3	Distros	10
1.3.1	Debian	10
1.3.2	Mint	10
1.3.3	fedora	11
1.3.4	centos	11
1.4	Apps and Packages	12
1.4.1	ncat	12
1.4.2	unrar	12
1.4.3	Firefox	12
1.4.4	Google Chrome	12
1.5	Games!	12
1.5.1	SuperTuxCart	12
1.5.2	AssaultCube	13

1 Linux

```
1 rsync -avz -e ssh \  
2   --exclude='*.log' \  
3   --exclude='temp/' \  
4   user@remote:/var/www/html/ \  
5   /backup/local_html/
```

1.1 General

1.1.1 Shell

- install and config **zsh**
 - LINK

- create new file

```
1 touch <filename>
```

- create a range of files
 - example:

```
1 touch myfiles{1..10}.txt
```

- **grep** one of the most practical linux commands for filtering:
 - example:
- **grep** one of the most practical linux commands for filtering:
 - example:

```
1 ls | grep <wanted-folder>
```

you have very very practical flag **-v** that **exclude** files.

- how to see current user in cli

```
1 whoami
```

- recommended to don't install snap on linux!
- just try download the file from the internet.
- change the password to everything you want!

```
1 sudo passwd youruser
```

- how to see the hostname

```
1 hostname
```

- how to see the hostname ip in LAN

```
1 hostname -I
```

first ip is your ipv4 in LAN

- how to see the current port usage:

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

**** __ does not contains hidden files!!!****

- add `alias` permanently in one command

```
1 echo "alias aliasname='aliascommand'" >> ~/.bashrc && source ~/.bashrc
2 echo "alias aliasname='aliascommand'" >> ~/.zshrc && source ~/.zshrc
```

- one of the fastest way to define your commands as shortcut is to use from `aliases`, even in your running project!

- in your project folder

- ★ create a file like name `setup.sh`

- ★ define your function

```
1 yourname() {
2     yourcommand
3 }
```

- 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

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

- see the disk usage useful commands

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

- see the partitions

```
1 lsblk
```

-
- go to your flash removable

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

- find in file contents

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

- fast dns add/remove

```
1 sudo sed -i '2i\nameserver 192.168.200.168' /etc/resolv.conf
2 sudo sed -i '2d' /etc/resolv.conf
```

- issue with entering bios in fast boot

```
1 systemctl reboot --firmware-setup
```

- issue with opening drive in kde gui or arch linux

```
1 lsblk
2 sudo ntfs-3g /dev/sda2 .
```

- 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

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

if you saw large files in linux think about NTFS.

for large files(> 2GB) zipping you should use from zip splitting feature in linux if you dont use from ntfs

- network manager setting from terminal with ui

```
1 nmtui
```

- see which ports are used!

```
1 netstat -tunlp
```

-
- see the proxies

```
1 $ env | grep -i proxy
2 NO_PROXY=localhost,127.0.0.0/8,127.0.1.1
3 http_proxy=http://192.168.1.250:8080/
4 FTP_PROXY=ftp://192.168.1.250:8080/
5 ftp_proxy=ftp://192.168.1.250:8080/
6 all_proxy=socks://192.168.1.250:8080/
7 ALL_PROXY=socks://192.168.1.250:8080/
8 HTTPS_PROXY=https://192.168.1.250:8080/
9 https_proxy=https://192.168.1.250:8080/
10 no_proxy=localhost,127.0.0.0/8,127.0.1.1
11 HTTP_PROXY=http://192.168.1.250:8080/
```

- see port status

```
1 nmap -p 5432 172.19.0.3
```

- remove a file recursively

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

- increase the time login time

```
1 sudo visudo
```

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

- running and background and also saving logs!

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

- see your background processes

```
1 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.

```
1 fg %1
```

- resumes job [1] in the background.

```
1 bg %1
```

- kills job [1].

```
1 kill %1
```

- to beautify the json format

```
1 jsonoutputer | jq .
```

- see log online

```
1 tail -f output.log
```

- find and replace all command from out of text file

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

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

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

- just a process to up system

```
1 sleep 600 # 10 minutes
```

- run a command periodic to see the result realtime

```
1 watch -n 0.5 your_command
```

- set static ip

```
1 sudo nmcli con mod <connection-name> ipv4.method manual ipv4.  
addresses 192.168.56.100/24 ipv4.gateway 192.168.56.1 ipv4.dns  
8.8.8.8
```

-
- see network interfaces

```
1 ip a
```

- see network connections

```
1 nmcli con show
```

1.1.2 systemd

- start the service

```
1 systemctl start yourservice.service
```

- stop the service

```
1 systemctl stop yourservice.service
```

- restart the service

```
1 systemctl restart yourservice.service
```

- enable the service to run after boot

```
1 systemctl enable yourservice.service
```

- disable the service to run after boot

```
1 systemctl disable yourservice.service
```

1.1.3 ssh

- get a file from remote

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

- get a folder complete from remote

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

- send a file to destination ip

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

-
- send a folder to destination ip

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

- copy files with progress status

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

- sha2 hash of string

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

- send DHCP

```
1 sudo dhclient -r
2 sudo dhclient
```

- change date

```
1 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!

1.1.4 Snap

- installing

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

- removing

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

1.1.5 Security

1.1.5.1 File Encryption

1.1.5.1.1 GPG (GNUPG)

- generate keys

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


-
- see your keys

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

- export your keys

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

- import your keys

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

- delete a key

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

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

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

- asymmetric encryption with pair key

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

- now decrypt symmetric

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

1.2 GUI's

1.2.1 Cinammon

- file manager: nemo

1.2.2 Xfce

1.2.2.1 apps

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

1.2.2.2 tricks

- zooming (in normal mode :))

- `alt+scroll`

1.2.3 Gnome

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

1.2.4 KDE

- file manager: dolphin

1.3 Distros

1.3.1 Debian

- install `.deb` package

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

- remove `.deb` package

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

1.3.2 Mint

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

```
1 sudo passwd root
```

- install `.deb` package

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

- remove `.deb` package

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

-
- for becoming root user

```
1 sudo su
```

- install font
 - download the font and its typefaces
 - copy your `.ttf` or `.otf` or ... files in the `~/ .fonts` folder
 - create `~/ .fonts` if it does not exist.
 - the fonts loaded after you reopen the apps that you are used!

1.3.3 fedora

- see the all installed Packages

```
1 rpm -aq
```

- download your rpm with **DEPENDENCIES**

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

- export rpm file.

```
1 sudo yum install rpmrebuild
```

- create rpm

```
1 rpmrebuild your_app_name
```

1.3.4 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!

1.4 Apps and Packages

1.4.1 ncat

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

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

- sending data to server

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

1.4.2 unrar

- extract `.rar` file

```
1 unrar x yourfile.rar
```

1.4.3 Firefox

- open app

```
1 firefox
```

1.4.4 Google Chrome

- open app

```
1 google-chrome
```

1.5 Games!

1.5.1 SuperTuxCart

- racing
- open-source
- singleplayer

-
- multiplayer
 - online and lan possible
 - 700 MB :)

1.5.2 AssaultQube

- shooting
- open-source
- singleplayer
- multiplayer
- online and lan possible
- 50 MB :))