

i'M DEVOPS @ gk lanit

visitors 404

visitors

[English/Английский](#) | [Русский/Russian](#) | [Translate](#)

What is Gentleman git repository:

- A Linux Beginner's Handbook for quick immersion, convenience and end-to-end learning skills for easy console operation. You will save a lot of time and reduced learning curve 🐱
- Linux Performance Tip's & Tools for Fast Work
- Instrument for reduce the time spent on comparison and selection among a variety of tools, providing a list of relevant solutions, trying to reduce the number of associated errors and increase a positive interaction experience
- Highlights for a long time way in OpenSource world
- Platform for start automated bootstrap with you working environment at new system without history
- Community for sharing experience that are common, affordable and time-saving

What is not Gentleman:

- Not [Wikipedia](#), [Google](#), [StackOverflow](#), [Man pages](#), new [GNU/Linux distributive](#)

Menu

- [gentleman minibook - \[Dive into Pinguin\]](#)
 - [Menu](#)
 - [Ubuntu](#)
 - [Terminal](#)
 - [Console](#)
 - [Shell](#)
 - [Bash](#)
 - [Zsh](#)
 - [History](#)
 - [Color](#)
 - [Autocomplete](#)
 - [Aliases](#)
 - [Modern console utils](#)
 - [File Manager](#)
 - [Search](#)
 - [Disk](#)
 - [Network](#)
 - [Monitoring](#)
 - [Log](#)

- [Man](#)
- [git](#)
 - [git-lifehacks](#)
 - [Git-flow](#)
- [Managers](#)
- [Automation](#)
 - [Text processor](#)
 - [Parsers:](#)
 - [Ansible](#)
 - [Interactive](#)
- [Text editor](#)
 - [nano](#)
 - [emacs](#)
 - [vim \(vi\)](#)
- [Archive](#)
- [Web Browser](#)
- [Solution](#)
- [Naked](#)
- [Tip's](#)
- [Scripts](#)
 - [sed](#)
- [Configs](#)
- [Filesystem](#)
- [Soft](#)
 - [Packages](#)
 - [APT/deb](#)
 - [Snap](#)
 - [Repos](#)
 - [Store](#)
 - [GUI/Desktop](#)
 - [Tools](#)
 - [App](#)
 - [cli](#)
 - [Web](#)
 - [Browser](#)
 - [Addons](#)
 - [Games](#)
 - [Server](#)
- [Performance](#)
 - [Kernel](#)
- [Hardware](#)
 - [Info](#)
 - [Drivers](#)
 - [CPU Microcode](#)
 - [GPU](#)
 - [Support & Compability](#)
 - [Manufacturers](#)
 - [Intel](#)
 - [Dell](#)
 - [Asus](#)
 - [Lenovo](#)
 - [HP](#)

- [Raspberry Pi](#)
 - [System76](#)
 - [KDE Slimbook](#)
- [Routers](#)
- [Cheatsheets](#)
- [Education](#)
 - [OpenSource](#)
 - [LFS](#)
 - [GNU](#)
 - [Philosophy](#)
- [Docker](#)
 - [docker-compose](#)
- [Kubernetes](#)
- [Windows user](#)
 - [Package Manager](#)
 - [PowerShell Core \(PS Core\)](#)
 - [Repository](#)
 - [PowerShell Core \(PS Core\)](#)
 - [Windows Terminal](#)
 - [Remote client](#)
 - [KiTTY](#)
 - [MobaXterm](#)
 - [KRDC](#)
 - [Reminna](#)
 - [Linux way](#)
 - [Git Bash](#)
 - [Problems](#)
 - [Advantages](#)
 - [Console](#)
 - [WSL](#)
 - [Hotkeys](#)
 - [App](#)
 - [Tools](#)
 - [Server](#)
- [Explanations and interesting facts](#)
- [Thinking out](#)
- [Top Bugs](#)
- [Criticism](#)
- [Requests/Wishes/Ideas](#)
- [Participation](#)
 - [Translating](#)
 - [Testing](#)
 - [Develop OpenSource](#)
 - [GitHub](#)
 - [GitLab](#)
 - [Promotion](#)
 - [Wiki](#)
- [History](#)
- [Forums](#)
- [People](#)
 - [Fathers](#)
 - [Developers](#)

- [Guru](#)
- [Popular](#)
- [Specified](#)
- [Other](#)
 - [Statistic](#)
 - [Poll](#)
 - [Infographics](#)
 - [Presentation](#)
 - [Installation](#)
 - [Boot](#)
 - [Restoring](#)
 - [Stabilization](#)
 - [Security](#)
 - [Telemetry](#)
- [Glossary](#)
- [ToDo](#) Need fork and finish: <https://github.com/willklein/markdown-menu>

Ubuntu

One of most popular Linux distributive for servers and desktop (see [Kubuntu with KDE](#)). Best for typical server infrastructure without graphics, Docker practice and recommended for beginners. Largest package repository based on Debian + [PPA](#) + [snap](#) and flatpak + any popular software support only Ubuntu.

Developed by: [Canonical](#) Creator: [Mark Shuttleworth](#) (also Space Adventures Tourist) Support ([LTS](#)): 5 year base + 5 extended Users: Goobuntu with KDE in Google, Wikipedia at servers, NASA, MIT, CERN, LIGO, DARPA, Uber, Tesla, Audi, Mercedes, etc.

Terminal

Console interface where input commands and see outputs.

Ctrl+Alt+F1-F9 - change real TTY output to monitor. Default F1 (RHEL) or F7 (Ubuntu). Can be helpfull if console busy or frozen GUI.

- [Magic SysRq](#):

```
echo 1 > /proc/sys/kernel/sysrq
echo "kernel.sysrq = 1" >> /etc/sysctl.conf
```

- Magickeys:
-

Alt+SysRq+k - ****K****ills all programs on the current virtual console, including X.

Alt+SysRq+b - re**B**oot. Or command `echo "b" > /proc/sysrq-trigger`

Console

```
Ctrl+D # exit, close most popular consoles
cat /etc/*-release # print OS release information
hostnamectl # show hostname info with virtualization info
uname -a # show linux kernel version
cd - # return to previous directory
whereis ls # return binary path
which ls # return path to file
groups # list groups of current user
cat /etc/passwd # list system users
./file.sh # run file.sh in current dir
```

Parameters for human use utils with most popular cases:

```
util -v # view util version or verbose output
util --version
util version
util -h # or -? # print help or human format view: ls -hal; df -h; free -h
util --help # or --usage
util help # often no print help
```

Shell

Change user default shell:

```
chsh -s /bin/bash user
```

Bash

Most popular unix-shell for execute command language. Bash scripts starts with `#!` (shebang) symbols in file and have `.sh` extension.

```
chmod +x script.sh # make file executable
bash -x script.sh # debug script with print runned command
```

color, l, ll, hh, enhanced history, recursive search, autocomplete

- `~/.bashrc` - init file for bash session. Mora about [bash](#) and [Prompt](#) (console invitation). Default `~/.bashrc` file with human and colored output system standard utils:

```
sed -i "s/#force_color_prompt/force_color_prompt/; s/#export  
GCC_COLORS/export GCC_COLORS/; s/#alias dir/alias dir/;  
s/#alias vdir/alias vdir/; s/-alF'/-alFh'/; s/-A'/-Ah'/;  
s/-CF'/-CFh'/" ~/.bashrc
```

Zsh

Powerful shell that operates as both an interactive shell and as a scripting language interpreter.

- Installation:

```
sudo apt install zsh powerline fonts-powerline -y # zsh-syntax-  
highlighting zsh-theme-powerlevel9k  
sh -c "$(wget -O- https://raw.githubusercontent.com/ohmyzsh/  
ohmyzsh/master/tools/install.sh)"
```

```
# Oh My Zsh plugin installation (better use zinit plugin  
manager):  
#git clone https://github.com/zsh-users/zsh-syntax-  
highlighting.git ${ZSH_CUSTOM:-~/.oh-my-zsh/custom}/  
plugins/zsh-syntax-highlighting  
#git clone https://github.com/zsh-users/zsh-autosuggestions $  
{ZSH_CUSTOM:-~/.oh-my-zsh/custom}/plugins/zsh-  
autosuggestions  
#echo "source /usr/share/powerlevel9k/powerlevel9k.zsh-theme" >>  
~/.zshrc
```

- [Plugin Manager](#)

```
sh -c "$(curl -fsSL https://raw.githubusercontent.com/zdharma/  
zinit/master/doc/install.sh)"
```

Restart shell for install zinit.

- [Theme](#)

```
echo "zinit ice depth=1; zinit light romkatv/powerlevel10k" >>  
~/.zshrc
```

To run Powerlevel10k configuration wizard:

```
p10k configure
```

- [Plugins](#)

```
zinit load zsh-users/zsh-autosuggestions  
zinit load zdharma/fast-syntax-highlighting
```

- [Syntax Hightlighting](#)

History

```
history
history -c # clean
    command # with space at first symbol command will not write to
             history
!<number> - will run command from history with selected number
!! - run last command in history
sudo !! - run last command in history with substitute user
        (default as root with admin privileges)
```

- Arrow Up & Down - command history scroll
- Ctrl+R - recursive command history search
- Add timestamp, remove duplicates, sync with any console session:

```
cat >> ~/.bashrc << EOF
export HISTTIMEFORMAT='%F %T '
export HISTIGNORE='ls -l:ls -la:pwd:date:ll:ls:l:'
export HISTCONTROL=ignoreboth
export PROMPT_COMMAND="{PROMPT_COMMAND:+$PROMPT_COMMAND$'\n'}
    history -a; history -c; history -r;"
EOF
```

Color

- Prompt - add red color for root session, print command execute result code:

```
cat >> ~/.bashrc << EOF
if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
fi
```

```
Last_Command=$? # Must come first!
Blue='\[\e[01;34m\]'
White='\[\e[01;37m\]'
Red='\[\e[01;31m\]'
Green='\[\e[01;32m\]'
Reset='\[\e[00m\]'
FancyX='\342\234\227'
Checkmark='\342\234\223'
## Add a bright white exit status for the last command
PS1="$White$? "
```

```

## If it was successful, print a green check mark. Otherwise,
    print
## a red X.
if $Last_Command == 0 ; then
    PS1+="$Green$Checkmark "
else
    PS1+="$Red$FancyX "
fi
## If root, just print the host in red. Otherwise, print the
    current user
## and host in green.
if $EUID == 0 ; then
    PS1+="$Red\\h "
else
    PS1+="$Green\\u@\\h "
fi
## Print the working directory and prompt marker in blue, and
    reset
## the text color to the default.
PS1+="$Blue\\w \\$Reset "
EOF

```

- [grc](#) - add more color to console output:

```
sudo apt install grc -y
```

```

cat >> ~/.bash_aliases << EOF
alias cvs='grc --colour=auto cvs'
alias diff='grc --colour=auto diff'
alias esperanto='grc --colour=auto esperanto'
alias gcc='grc --colour=auto gcc'
alias irclog='grc --colour=auto irclog'
alias ldap='grc --colour=auto ldap'
alias log='grc --colour=auto log'
alias netstat='grc --colour=auto netstat'
alias ping='grc --colour=auto ping'
alias proftpd='grc --colour=auto proftpd'
alias traceroute='grc --colour=auto traceroute'
alias wdiff='grc --colour=auto wdiff'
alias dig='grc --colour=auto dig'
alias ll='grc --colour=auto ls -laFh --color=always'
alias cat='grc --colour=auto cat'
alias zcat='grc --colour=auto zcat'
alias make='grc --colour=auto make'
alias gcc='grc --colour=auto gcc'

```



```
alias g++='grc --colour=auto g++'
alias head='grc --colour=auto head'
alias mount='grc --colour=auto mount'
alias ps='grc --colour=auto ps'
alias mtr='grc --colour=auto mtr'
alias zgrep='grc --colour=auto zgrep'
alias l='grc --colour=auto ls -lah --color=always'
EOF
```

Autocomplete

Check in your system:

```
cat ~/.bashrc | grep completion
```

Default already in Ubuntu:

```
sudo apt install bash-completion -y
cat >> ~/.bashrc << EOF
if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
fi
EOF
```

- For [kubectl]: see [Kubernetes](#) partition.
- <https://github.com/cloudnativelabs/kube-shell>
- kail

Aliases

Add alias for command to ~/.bash_aliases:

```
cat >> ~/.bash_aliases << EOF
alias upd="sudo apt update && sudo apt full-upgrade -y"
alias tcp="ss -altp"
EOF
```

Add more useful aliases based on all you shell command history with [topalias](#):

```
pip install --user topalias
topalias # add -z flag for zsh
```

Modern console utils

- [bat](#) - cat clone with syntax highlighting and Git integration + [delta](#) for Git and diff + [bat-extras](#) (batwatch, batdiff, prettybat).

```
sudo apt install bat delta -y
```

- [peep](#) - less and more alternative
- [exa](#) - better than ls
- [f***](#) - magnificent app, that corrects errors in previous console commands. You can add safe fast [aliases](#) 'f' and 'F' to force -y without confirmation.

```
sudo apt install python3-dev python3-pip python3-setuptools -y
pip3 install --user thefuck
```

```
puthon #example
#No command 'puthon' found, did you mean
fuck
#python [enter/↑/↓/ctrl+c]
```

- Run any command in one line without check execute result - separate command with ; (semicolons): touch \$(date).txt; ls -lah
- Run next command in line if previous was successfully completed (exit code: 0) - separate command with && (ampersand): sudo apt update && sudo apt full-upgrade -y

File Manager

- [mc](#) - dual-panel visual (working from console without GUI) file manager with SFTP/SMB client, mceditor, archive navigator. Like hotkeys:

```
Ctrl+O - switch panels to console and back to mc
Ctrl+\ - open path bookmarks
Ctrl+X+S - create simlink
Ctrl+X+T - print filename to bottom console line
F3 - view file
F4 - edit file
F4 (in mcedit) - find and replace
Shift+Mouse select - copy selected text text
Shift+RightClick or Shift+Insert - paste clipboard
Shift+Del - cut selected text
F10 - exit
Ctrl+S - search file
Insert - mark file for multiple action
F5 - copy selected or marked files
```

- Full list hotkeys/shortcuts: [panels](#), [actions](#), [text editor](#) (mcedit).

- [Midnight Commander hotkeys](#).
- Switch to mcedit for default editor (optional):

```
select-editor
sudo update-alternatives --config editor
EDITOR=mcedit mc
cat >> ~/.bashrc << EOF
EDITOR=mcedit mc
export EDITOR
EOF
```

In panels mode:

- **F9** Activates the top menu.
- **o** Selects the Option menu.
- **c** Opens the configuration dialog.
- **i** Toggles the use internal edit option.
- **s** Saves your preferences.

Search

- [fd](#) - simple, fast and user-friendly alternative to find

```
sudo apt install fd-find -y
```

- [fzf](#) - fuzzing search

```
sudo apt install fzf -y
```

- [mlocate](#)

```
sudo apt install mlocate -y
sudo updatedb # update file index, autoupdate every night
sudo locate root | grep txt # search all system
locate root | grep txt # search with user permissions
```

Disk

```
df -h # list volumes, free space, mount points
mount # list mounted points
```

- [pydf](#) - better than df

```
pip install --user pydf
```

- [ncdu](#) # or nnn

```
sudo apt install ncdu -y
```

- [vifm](#)

- [diskonaut](#) # better than ncdu

Network

```
ip a # show network address, instead legacy "ipconfig" - short
      and doesn't need sudo
ss -alt # show open tcp ports, instead "netstat -lnp" - short
        and doesn't need sudo
```

Monitoring

- [htop](#) - console interactive process viewer with resources load visualisation (analog Windows task manager)

```
sudo apt install htop
```

View IOWAIT: F2 -> display options → detailed CPU time

- [bashtop](#)

```
sudo add-apt-repository ppa:bashtop-monitor/bashtop
sudo apt update
sudo apt install bashtop -y
```

- Search process:

```
ps -A aux | grep <proc>
```

Log

```
tail -f /var/log/dpkg.log # print real-time log or other file
sudo tail -f /var/log/* # printing multiple file logs
```

- [lnav](#) TODO: log rotation TODO: systemd-journald

Man

```
man -k login # search mans with login context
PgDown/PgUp # scroll page
/<word> # in opened man search text <word> in all man (need Enter
        for command).
/ # search next (Enter)
q # exit
```

- [tldr](#) - short manual from internet
- [ealdeer](#) - fast implementation of tldr in Rust
- pinfo - curses based lynx-style info browser, better than man

cargo install tealdeer

- batman # see [Console](#)

git

git add -A # add to index all files in folder/subfolder and hidden (name start with dot) too

- lazygit - console client
- [tig](#) - text-mode interface for git

git-lifefhacks

- [BFG](#) - removes large or troublesome blobs like git-filter-branch does
- git commit -amend -m "an updated commit message"
- rebase -i

Git-flow

TODO: Presentation

Managers

- screen

screen pstree # start new session with command pstree

screen -S build ninja # start session "build" with command "ninja"

screen -r build
restore session "build", remove session name to restore last session

screen -ls # list opened session

- tmux (default installed in Ubuntu)

tmux new -s session_name # create new named session

tmux ls # show opened session

tmux attach-session -t 0 # attach for session with number 0

Ctrl+b (Action mode) **c** # create a new window (with shell)

Ctrl+b **w** Choose window from a list

Ctrl+b **0** Switch to window 0 (by number)

Ctrl+b **%** Split current pane horizontally into two panes

Ctrl+b **%** Split current pane vertically into two panes

Ctrl+b **o** Go to the next pane

Ctrl+b **x** Close the current pane

Session manager: [tmuxp](#)

- byobu (default installed in Ubuntu)

byobu - run new or restore last session
Shift+F6 - detach session without exit
F2 - add new tab window
F3 - previous tab window
F4 - next tab window
Ctrl+F2 - split vertical windows
Shift+F2 - split horizontal windows
Shift+F3 - next window zone
Ctrl+a+k - close current window
Ctrl-A [- scroll window

Automation

- cron:

```
crontab -e # edit cron jobs for current user
sudo crontab -e -u root # edit root cron jobs
## min (0-59) # hour (0-23) # day of month (date 1-31) # month
    (1-12) # day of week (1-7) # use '*' in these fields (for
    'any')
30      12      22      31
    12      7      # cron job format
example
```

```
cat /etc/crontab # print scheduled tasks
```

TODO: apt autoupdate TODO: e-mail notification

Text processor

- ack - better than grep
- [ag](#) - better than ack awk, sed (ed), grep
- [ripgrep](#) - better than ag
- [batgrep](#) - better than grep, ack, ag, ripgrep (best)

```
grep -H # print search string with file path
```

Parsers:

- JSON

```
sudo apt install jq -y
jq '.parameter[0].next_param'
```

[jqed](#) - interactive query generator for jq

```
pip3 install --user urwid urwid-readline
cd ~/Downloads
git clone https://github.com/Wazzaps/jqed
sudo install -o root -g root ./jqed/jqed /usr/bin/
```

[fx](#) - better than jq

- XML

```
sudo apt install xmlstarlet -y # user xml command for XPath
                                action
xml val file.xml
xml sel -t -v "/xml/..xpath.." file.xml
```

Ansible

Automation and orchestration.

- Installation:

```
pip3 install --user ansible
```

Interactive

- [percol](#)
- aptitude - ncurses Apt shell

Text editor

nano

- When you open file without Write-access privileges (root file from user without sudo) - on bottom printed warning "unwritable".

```
nano +42 file.txt # open text file on 42 line number
F2 - save and exit from nano
Ctrl+O - save file
Ctrl+X - exit from editor
Ctrl+W - search
Alt+R - find and replace
Ctrl+C - show current position line/row
```

- Add syntax highlight without internet and extended formats:

```
find /usr/share/nano/ -iname "*.nanorc" -exec echo include {} \;
>> ~/.nanorc
```

Improved [Nano Syntax Highlighting](#):

```
wget https://raw.githubusercontent.com/scopatz/nanorc/master/
install.sh -O- | sh
#wget https://raw.githubusercontent.com/scopatz/nanorc/v2.9/
install.sh -O- | sh # for nano <= v2.9
```

emacs

- [spacemacs](#)

F10 - Menu

vim (vi)

- [neovim](#)

Exit:

- Without saving changes: Esc (for Command mode) -> :q! -> Enter
- With saveing changes: Esc (for Command mode) -> :q -> Enter or ZZ (capital with Shift) in Text mode for Save and Exit.
- Edit Mode: I (without Shift)
- Text Mode: Esc from Edit Mode or Command mode)
- Search: in Command mode type /text -> Enter

Plugin Manager:

- <https://github.com/junegunn/vim-plug>
- <https://github.com/Shougo/dein.vim>

Plugins:

vimv - better than vim rename & thameera

```
sudo apt install neovim -y
sudo apt install python3-neovim -y
pip3 install --user neovim
nvim
```

Archive

```
tar xvfz archive.tar.gz # unpack tar.gz archive
unzip archive.zip # unpack ZIP-archive
sudo apt install p7zip p7zip -y
7z x myfile.7z; 7z x archive.rar # uncompress an 7-zip/RAR
archive
```


Web Browser

```
sudo apt install elinks -y
sudo apt install w3m -y # with emacs integration
```

Solution

TODO: Useful complex commands with explanation.

Naked

Helpful command to look around

```
help help # reference
man man # manuals
env # environment variables
<double>-<TAB> # list all executable command from autocomplite
ps; ls -la; cat; top
```

Tip's

```
sudo hostnamectl <short hostname> # correct change hostname, need
                                argument short hostname, not fqdn
apt # use one instead apt-get & apt-cache commands
sudo hwclock # real hardware time on machine
```

Scripts

1. Mass rename files - replace "name_start" and "name_finish":

```
for f in *.name_start; do
    mv -- "$f" "${f%.name_start}.name_finish"
done
```

sed

- sed1line: <http://sed.sourceforge.net/sed1line.txt>

Configs

Filesystem

- ext4 - default for Ubuntu and for most volumes on linux OS devices (mobile, network, multimedia, etc.).
- XFS - default for RHEL/CentOS
- Btrfs - default for [openSUSE](#). zypper (snapper) and apt ([apt-btrfs-snapshot](#)) with Btrfs on system volume can save installed system

package snapshot and rollback if system was damaged. Supported in Ubuntu, Oracle Linux, Fedora. Active developer: Facebook.

- exFAT - filesystem for flash-drive memory. Limited support on some devices. Can't be used for boot device.
- [NTFS](#) - Linux support NT-filesystem on read/write.

Soft

Packages

APT/deb

```
dpkg -l | grep <package> # find installer package and version:
```

```
apt search <package> | grep <package> # find package in repository
```

```
apt show <package> # package about info
```

```
sudo apt -f install # fix broken install packages
```

- [apt-fast](#) - shellscript wrapper for apt-get and aptitude that can drastically improve apt download times by downloading packages in parallel

```
sudo add-apt-repository ppa:apt-fast/stable
```

```
sudo apt update
```

```
sudo apt install apt-fast -y
```

- apt-file

```
sudo apt install apt-file -y
```

```
sudo apt-file update
```

```
apt-file search libmp3lame.so.0
```

- aptitude - console interactive package manager
- synaptic - GUI package manager

Snap

Package management without dependency hell as well as Docker for services. Not replacement deb/rpm system, supported in [20+ popular distributives](#). Use latest GUI app version in any distros and developers no need to compile a separate package for each distribution. Provide All-in-One containerised app will work on [most Desktop Linux OS](#). [Secure](#) and other isolation ([AppArmor & Seccomp](#)) features.

- Install on Kubuntu:

```
sudo apt update
```

```
sudo apt install snapd -y
```

```
sudo apt install plasma-discover-snap-backend -y # KDE Discover
integration
sudo snap install snap-store
sudo snap set system refresh.retain=2 # storage limit for
previous app versions on disk
```

- Install Snap Application from web [Snap Store](#) (One Click install), KDE Discovery, GNOME Software or [console](#).
- Install snapd in other Linux distributives (similar simple): [snapd](#) and [Snap Store](#).

Repos

- Google Chrome:

```
sudo bash -c 'cat >> /etc/apt/sources.list.d/google-chrome.list
<< EOF
deb [arch=amd64] http://dl.google.com/linux/chrome/deb/ stable
main
EOF'
```

Store

- Steam

```
sudo add-apt-repository multiverse
sudo apt update
sudo apt install steam -y
```

GUI/Desktop

Tools

- [CopyQ](#) - advanced clipboard manager with editing and scripting features.

```
sudo add-apt-repository ppa:hluk/copyq
sudo apt update
sudo apt install copyq
```

App

- WPS Office - proprietary free office suite with high MS Office compability and similar design from China.

```
sudo snap install wps-2019-snap
sudo snap connect wps-2019-snap:cups-control :cups-control
sudo snap connect wps-2019-snap:alsa :alsa
```

```
sudo snap connect wps-2019-snap:pulseaudio :pulseaudio
sudo snap connect wps-2019-snap:removable-media :removable-media
```

- [Telegram Desktop](#) - cross-platfrom quick online messenger, works with poor connection and low internet speed.

cli

- [neofetch](#)

```
sudo apt install neofetch -y
```

- [telegram-cli](#) - cross-platfrom quick online messenger, works with poor connection and low internet speed.

```
sudo snap install telegram-cli
telegram-cli -k <public-server-key>
telegram-cli msg <peer> text - sends message to this peer
```

Web

- ExplainShell parse linux command: <https://explainshell.com/>
- [SpellCheck](#) - check bash script: <https://www.shellcheck.net/>

Browser

Addons

Games

Server

- Lab - Source Code Management (vcs:git), DevOps platform (CI/CD, Monitoring, Kubernetes integration), Project Management and other with more service support (YouTrack, JIRA, oAuth, Gmail, Trello, Slack, Mattermost, HipChat, TeamCity, Jenkins, Bamboo, GitHub, [other](#))
- Sourcegraph - code search system (GitHub, GitLab, other)

Performance

Kernel

- [Liquorix](#) - distro kernel replacement built using custome configuration and kernel sources for desktop, multimedia, and gaming workloads. May be not working with proprietary drivers and not really stable. Installing (not recommended, at your peril):

```
sudo add-apt-repository ppa:damentz/liquorix
sudo apt update
sudo apt install linux-image-liquorix-amd64 linux-headers-liquorix-amd64
```

zram, filesystem, kernel (liquorix), grub, initramfs, preload, sysctl (vm.swappiness)

Hardware

Info

Drivers

[fwupd](#) - Linux Vendor Firmware Service for Autoupdates BIOS and drivers, integrated with KDE Discover and GNOME Software. [Vendors support](#) lot of [devices](#).

CPU Microcode

GPU

Proprietary vs OpenSource

Support & Compability

Manufacturers

Intel

NUC

Dell

XPS 13 Developers Edition

Asus

Laptops

Lenovo

Thinkpad

HP

Servers

Raspberry Pi

System76

[KDE Slimbook](#)

Routers

- Prometheus
- OpenSource firmware

Cheatsheets

Print hotkey/command/syntax tips of the product, that you are currently studying/use more, add magnetically or tape attach to the Hardware system unit or wall. In few months you can give to someone who has not yet had time to learn hotkeys or place the smoking room.

- [Midnight Commander]
- [Linux commands cheat sheet](#) – popular linux commands
- [Markdown](#)

Education

OpenSource

LFS

GNU

Philosophy

Docker

- lazydocker
- Portainer

docker-compose

```
pip3 install --user docker-compose
```

Kubernetes

- Installer: Rancher/Kubespray
- Enable kubectl autocompletion:

```
echo 'source <(kubectl completion bash)' >> ~/.bashrc
kubectl completion bash >/etc/bash_completion.d/kubectl
```

- [kube-shell](#)

Windows user

Package Manager

- [Chocolatey](#) - [repository](#) package manager with console and [GUI client](#).
Run in CMD.exe as Administrator:

```
@powershell -NoProfile -ExecutionPolicy Bypass -Command
"[Net.ServicePointManager]::SecurityProtocol =
[Net.SecurityProtocolType]::Tls12; iex ((new-object
net.webclient).DownloadString('https://chocolatey.org/
install.ps1'))" && SET PATH=%PATH%;%ALLUSERSPROFILE%
\chocolatey\bin
choco feature enable -n allowGlobalConfirmation
choco install ChocolateyGUI
choco install nuget.commandline --pre
```

Usage:

```
choco search # find package, alias for choco list
choco install notepadplusplus # install package
choco uninstall <package> # delete/remove
choco upgrade <package> # update program
choco info <package> # show about
choco -h # help
```

- [winget](#) - Microsoft powered cli tools for package management, distributed with [App Install](#). In development and gathering packages ([manifest list](#)).
- [Microsoft Store](#) - applicaion software & themes marketpalace for Windows.

Free program examples:

- Slack: <https://www.microsoft.com/ru-ru/p/slack/9wzdnck3wp>
- DBeaver CE: <https://www.microsoft.com/store/productId/9PNKDR50694P>
- Python 3.9: <https://www.microsoft.com/store/productId/9P7QFQMJRFP7>
- Windows Terminal: <https://www.microsoft.com/store/productId/9N0DX20HK701>
- Sticky Notes: <https://www.microsoft.com/store/productId/9NBLGGH4QGHW>

- Power BI Desktop: <https://www.microsoft.com/store/productId/9NTXR16HWNW1T>
- Ubuntu Linux (WSL): <https://www.microsoft.com/store/productId/9NBLGGH4MSV6>
- App Installer: <https://www.microsoft.com/store/productId/9NBLGGH4NNS1>
- AIDA64: <https://www.microsoft.com/store/productId/9NBLGGH2WNWH>
- Rufus: <https://www.microsoft.com/store/productId/9PC3H3V7Q9CH>

Fonts:

- Verdana Pro: <https://www.microsoft.com/store/productId/9N8D67VHHDC2>
- Arial Nova: <https://www.microsoft.com/store/productId/9NS5CT1MZ7M8>

Hardware:

- Intel Graphics Command Center: <https://www.microsoft.com/store/productId/9PLFNLNT3G5G>
- OpenCL & OpenGL: <https://www.microsoft.com/store/productId/9NQPSL29BFFF>
- Kyocera Print Center: <https://www.microsoft.com/store/productId/9WZDNCRFJ2V4>
- Samsung Printer Experience: <https://www.microsoft.com/store/productId/9WZDNCRFHWGG>
- Epson Print and Scan: <https://www.microsoft.com/store/productId/9WZDNCRFJ4P8>
- Realtek Audio Control: <https://www.microsoft.com/store/productId/9P2B8MCSVPLN>
- Dolby Access: <https://www.microsoft.com/store/productId/9N0866FS04W8>
- MyASUS: <https://www.microsoft.com/store/productId/9N7R5S6B0ZZH>

Codecs:

- Raw Image Extension: <https://www.microsoft.com/en-us/p/raw-image-extension/9nctdw2w1bh8>
- AV1 Video Extension: <https://www.microsoft.com/en-us/p/av1-video-extension/9mvzqvxbq9v>
- Web Media Extension: <https://www.microsoft.com/store/productId/9N5TDP8VCMHS>
- Webp Image Extension: <https://www.microsoft.com/store/productId/9PG2DK419DRG>
- MPEG-2 Video Extension: <https://www.microsoft.com/store/productId/9N95Q1ZZPMH4>
- VP9 Video Extension: <https://www.microsoft.com/store/productId/9N4D0MSMP0PT>
- Photos Media Engine Add-on: <https://www.microsoft.com/store/productId/9PLK42WD0RC0>

Microsoft Office - Templates: <https://templates.office.com> - Plug-in apps: <https://store.office.com>

PowerShell Core (PS Core)

`choco install powershell-core`

Repository

Open PowerShell as Administrator:

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned
Install-PackageProvider -Name NuGet -Force
Register-PSRepository -Name PSGallery -SourceLocation https://
www.powershellgallery.com/api/v2/ -PublishLocation https://
www.powershellgallery.com/api/v2/package/ -ScriptSourceLocation
https://www.powershellgallery.com/api/v2/items/psscript/ -
ScriptPublishLocation https://www.powershellgallery.com/api/v2/
package/ -InstallationPolicy Trusted -PackageManagementProvider
NuGet # Add PSGallery for Powershell with version less than v5
Set-PackageSource -Name PSGallery -Trusted # If registered, but
Untrusted
Register-PSRepository -Default -InstallationPolicy Trusted #
Register PSGallery for PS5+
Register-PackageSource -Name Nuget -Location "http://
www.nuget.org/api/v2" -ProviderName Nuget -Trusted
Install-Module -Name PowerShellGet -Force
Install-Module -Name PackageManagement
```

PowerShell Core (PS Core)

`choco install powershell-core`

Windows Terminal

<https://www.microsoft.com/en-us/p/windows-terminal/9n0dx20hk701>

- Fonts: <https://github.com/microsoft/cascadia-code/releases>
- Themes: <https://windowsterminalthemes.dev/>

PowerShell Core:

```
Install-Module posh-git -Scope CurrentUser
Install-Module oh-my-posh -Scope CurrentUser
```

Run:

`notepad $PROFILE`

Add and save:

```
Import-Module posh-git
Import-Module oh-my-posh
Set-Theme Paradox
```

Tips:

```
cd D:\ # if doesn't change directory try:
D:
```

Remote client

Install command in cmd as Administrator:

```
choco install kitty winscp mobaxterm
```

KiTTY

PuTTY-based SSH client.

- Installation path: C:.exe
- If you uses SSH keys - run SSH-agent and you no need enter passphrase every new session: C:.exe
- WinSCP - SFTP/FTP client with Kitty integration

MobaXterm

Proprietary free integrated SSH/RDP-client with session manager. Multi-protocol client & server All-in-One combine.

Client support network protocols: RDP, SSH, VNC, SFTP, FTP, AWS S3, WSL, XDMCP, Telnet, Mosh, Rsh (Rlogin) and Serial COM-port. Server tools (all it run in Windows): HTTP server, FTP server, SFTP server, TFTP server, NFS server, SSH server, Telnet server, VNC server, Iperf server, X server for Windows (display Linux X11 GUI app). Integrated tools: ZSH shell for Windows, Cron sheduler and other Network, System, Office, Linux and Console utilities

- X server for Windows - view Linux GUI X11 app in Windows over network. Command for Linux after start X server in MobaXterm:

```
DISPLAY=<IP>:0 gedit # IP - network address Windows machine
```

KRDC

RDP/VNC client for [KDE](#).

Remmina

RDP/VNC client, default in Ubuntu.

```
sudo snap install remmina
```

Linux way

Console utilities, Git/Bash for Windows, cmder shell (based on ConEmu & clink & FAR), Chocolatey/winget.

- Commandline utils installation command for chocolatey (as Administrator):

```
choco install git.install bat ag fzf delta clink cmder curl  
openssl.light python3 wget less neovim
```

Git Bash

```
ipconfig /all # doesn't work  
ipconfig -all # parameter  
ipconfig //all # duplicate slash  
ipconfig \all # escape slash
```

- [Sampler](#)

Problems

Advantages

Console

WSL

Hotkeys

Win+D - hide all windows
Win+E - open my computer folder
Win+V - clipboard manager
Ctrl+Shift+` - open Task Manager

App

- [7-zip] <https://www.microsoft.com/en-us/p/7-zip-file-manager-unofficial/9mz81rmk8jfd> - [Open Source](#) de facto standard all format archiver.
- [Notepad++](#) - [Open Source](#) fast modern text editor with syntax highlight and plugins.
- [Sticky Notes](#) (UWP app from MS Store) - text notes on screen sticks with online sync.
- CopyQ - [Open Source](#) advanced clipboard manager with editing and scripting features (or you can enable modern clipboard in Start -> Settings -> System -> Clipboard -> Clipboard history -> Win+V).

choco install copyq

- PowerToys - [Open Source](#) Microsoft utilities for power users to tune and streamline their Windows experience for greater productivity (Open Source alternative: [ImageGlass](#)).

Long press Win - show hotkeys

Shift+Arrow - snap window to FancyZones.

- digiKam - [Open Source](#) KDE powered image organizer.
- FS Image Viewer - [proprietary](#) free all-screen image viewer with 4 screen-sides tools panel and integrated editor.
- VLC - [Open Source](#) cross-platform video player.
- qBittorrent - [Open Source](#) Qt-based bittorrent client.
- Telegram - [Open Source](#) cross-platfrom quick online messenger, works with poor connection and low internet speed.
- Bitwarden - [Open Source] cross-platform password manager with online sync.
- PyCharm - [Open Source](#) JetBrains powered IDE with [plugins](#).
- Visual Studio Code - [Open Source] cross-platfrom Microsoft powered modern Atom-based code editor with [extensions](#). Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. More languages, debuggers, and tools in [Marketplace](#).
- SourceTree - [proprietary](#) Atlassian powered Git/Mercurial GUI client with Git-flow and Bitbucket integration, visual merge conflic and graph, large commit information, interactive rebase.
- Fiddler - [proprietary](#) cross-platform debugging proxy HTTP(S) traffic sniffer for web developers.
- Rufus - [OpenSource](#) utility that format and create from ISO bootable USB flash drives with Linux/Windows/other installation & live disk image.
- Slack - [proprietary](#) cross-platform business communication messenger platform with [GitHub](#) integration.

Tools

Autoruns, Fiddler, Hunspell

Server

Docker Heroku pgAdmin4

Explanations and interesting facts

- RPM5
-

Thinking out

Top Bugs

Criticism

Requests/Wishes/Ideas

Participation

Translating

Testing

Develop OpenSource

GitHub

Repository list, star, fork, pet

GitLab

Promotion

Wiki

History

Forums

People

Fathers

Developers

Guru

Popular

Specified

Other

Statistic

- [Ubuntu internal statistics](#)
- [All GNU/Linux distributives list.](#)

Poll

Infographics

Presentation

Installation

Boot

Restoring

Stabilization

Security

- Disable read/executable flags for other at home directory:

chmod 750 ~

- sudo vs su
-

Telemetry

Glossary

ToDo