

Copyright (C) 2025 Shaon. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

----- Programs -----

```
$ sudo pacman -Syu i3 i3status i3lock i3blocks rofi lf nsxiv ffmpeg feh picom network-manager-applet volumeicon unzip unrar p7zip tar xz scrot pulseaudio audacity nvim lf nsxiv redshift --needed fish fzf htop lm_sensors tldr lshw fastfetch ncd u keepassxc figlet kdenlive newsboat udisks2 cryptsetup exfatprogs ntfs-3g imagemagick unzip p7zip libarchive xarchiver --needed zathura zathura-pdf-mupdf ghostscript enscript mat2 perl-image-exiftool
```

```
$ which i3 i3status i3lock i3blocks rofi lf nsxiv ffmpeg feh picom volumeicon unzip unrar tar xz scrot pulseaudio audacity nvim lf nsxiv redshift fish fzf htop sensors tldr lshw fastfetch ncd u keepassxc figlet kdenlive newsboat udisksctl cryptsetup ntfs-3g magick unzip xarchiver zathura ghostscript enscript mat2 exiftool
```

```
$ sudo pacman -S base-devel git
$ cd ~ && git clone https://aur.archlinux.org/yay.git && cd yay && makepkg -si
$ yay --version
$ sudo curl -L https://github.com/yt-dlp/yt-dlp/releases/latest/download/yt-dlp -o /usr/local/bin/yt-dlp
$ sudo chmod a+rx /usr/local/bin/yt-dlp
$ yt-dlp --version
$ yay -S vscode-bin
```

----- i3 -----

```
logout, select i3 and login
$ nano ~/.config/i3/config
refresh i3 --- mod+shift+r
$ sudo nano /etc/lightdm/lightdm.conf
inside there find user-session=xfce & autologin-session=xfce. Replace
"xfce" with "i3" and save
```

----- redshift -----

```
$ redshift -O 3333
To Reset : $ redshift -x
```

----- fish -----

```
$ fish
$ chsh -s /usr/bin/fish
then log out from xfce session, log back in and then
$ set -eg fish_greeting
$ set -U fish_greeting
```

----- fzf -----

```
$ fzf
preview a file while searching -
$ fzf --preview='cat {}'
open a file in nvim after searching inside fzf -
$ nvim $(fzf --preview='cat {}')
```

----- lf -----

```
$ mkdir -p ~/.config/lf
$ nano ~/.config/lf/lfrc
$ mkdir -p ~/.config/nvim
$ nvim ~/.config/nvim/init.vim
```

----- ncd u -----

```
$ ncd u
```

----- lshw -----

```
$ lshw
$ sudo lshw -C cpu
```

----- lf -----

```
$ figlet "hi there"
$ figle "Yo fam"
$ figlet -f script "Yo fam"
$ figlet -f banner "Yo fam"
$ figlet -c -f banner "Yo fam"
```

----- newsboat -----

```
$ mkdir ~/.config/newsboat
$ cd ~/.config/newsboat
$ nvim urls (paste your urls)
$ nvim cnfig (paste your config)
```

----- Imagemagick -----

```
$ magick image1.jpg -background black -vignette "100x333" image1_output.jpg
$ magick image1.jpg -solarize 44% image1_output.jpg $ magick image1.jpg
-blur 0x4 image1_output.jpg $ magick image1.jpg -negate image1_outasaput.jpg
$ magick image1.jpg -charcoal 4 image1_out33ut.jpg $ magick image1.jpg
-colorspace gray image1_11ut.jpg $ magick image1.jpg -bordercolor red
-border 33 image1t.jpg $ magick image1.jpg -modulate 100,266 image1t.jpg $
magick image1.jpg -paint 6 it.jpg
```

- Display Server: yet dunno but Xorg! / X11! Hell NOO!!! -

----- p7zip -----

```
Archive a folder: $ 7z a web_search_tools.7z web_search_tools
To unarchive: $ 7z x web_search_tools.7z
```

--- usdisks2 + cryptsetup ---

note: first check the targeted USB drive by running the command lsblk ---

```
# Rename (label) an exFAT drive
$ sudo exfatlabel /dev/sdX1 NewName
```

```
# Rename (label) an NTFS drive
$ sudo ntfslabel /dev/sdX1 NewName
```

```
# Only Apply LUKS Encryption (no format)
$ sudo cryptsetup luksFormat /dev/sdX1 && \ sudo cryptsetup open /dev/sdX1
secure_usb
```

```
# Only Format as exFAT (no encryption)
Way # 1i,\217 Unmount the correct mountpoint sudo umount /run/media/zzz/mysam
```

```
#Close the LUKS mapper
$ sudo cryptsetup close luks-d13f7771-6d20-4707-aa85-ec55188e239a
```

```
#Force kernel + udev to refresh before
# formatting
$ sudo udevadm settle sudo partprobe /dev/sda
```

```
#Verify closure
$ lsblk -fp
```

```
#Format safely as exFAT
$ sudo mkfs.exfat -n MyB /dev/sda1
```

```
Way # 2 Unmount the correct mountpoint sudo umount /run/media/zzz/mysam
```

```
#Close the LUKS mapper
$ sudo cryptsetup close luks-d13f7771-6d20-4707-aa85-ec55188e239a
```

```
#Verify closure (mapper should disappear)
$ lsblk -fp
```

```
#Now it's safe to format as exFAT
$ sudo mkfs.exfat -n MyB /dev/sda1
```

```
# Only Format as NTFS (no encryption)
$ sudo mkfs.ntfs -f -L MyUSB /dev/sdX1
```

note: while formatting + luks is being attempted, if anything it says like - Cannot exclusively open /dev/sda1, device in use - then simply run these command below ---

```
$ lsblk-fp
```

identify your drive and see if mounted and showing its capacity, is showing then

```
$ sudo umount /run/media/zzz/FD86-090D (your drive name) then, $ lsblk -fp
```

now it should not show the drive's capacity or like that & now it's all okay to go on...

```
# formatting a USB drive
--- "exFAT + LUKS"
$ sudo cryptsetup luksFormat /dev/sdX1 && sudo cryptsetup open /dev/sdX1
secure_usb && sudo mkfs.exfat -n mydrivesam /dev/mapper/secure_usb && sudo
cryptsetup close secure_usb
```

```
# formatting a USB drive
--- "NTFS + LUKS"
$ sudo cryptsetup luksFormat /dev/sdX1 && sudo cryptsetup open /dev/sdX1
secure_usb && sudo mkfs.ntfs -f -L Mydrivesand /dev/mapper/secure_usb &&
sudo cryptsetup close secure_usb
```

```
# To unlock and mount the drive
--- "exFAT"
$ sudo cryptsetup open /dev/sdX1 secure_usb && sudo mkdir -p /mnt/usb &&
sudo mount -t exfat /dev/mapper/secure_usb /mnt/usb
```

```
# To unlock and mount the drive
--- "NTFS"
$ sudo cryptsetup open /dev/sdX1 secure_usb && sudo mkdir -p /mnt/usb &&
sudo mount -t ntfs /dev/mapper/secure_usb /mnt/usb
```

```
# To lock and mount --- "both exFAT & NTFS"
$ sudo umount /mnt/usb && sudo cryptsetup close secure_usb
```

```
# PARTITION MANAGEMENT TOOLKIT 2024
```

```
PART 1 2024 CREATE NEW PARTITIONS (WIPING OLD TABLE)
Identify your target drive
lsblk -fp
2024 Example:
/dev/sda 128G 2024\2024\200/dev/sda1 vfat MyOldUSB
`/dev/sda` = the *whole* drive. Double-check before continuing.
```

```
Wipe old signatures (optional but safest)
$ sudo wipefs --all /dev/sda
```

Removes any old partition tables/filesystem traces. Destroys all data 2024 verify the device name first. Create new GPT partitions (modern layout)

```
$ sudo parted /dev/sda --script mklabel gpt \ mkpart primary 0% 28GB \ mkpart
primary 28GB 100%
```

```
Creates two partitions:
/dev/sda1` 2024 0 2024\22328 GB /dev/sda2` 2024 28 GB 2024\223 end (~100 GB)
(For BIOS/MBR systems)
$ sudo parted /dev/sda --script mklabel msdos \ mkpart primary 0% 28GB \
mkpart primary 28GB 100%
```

Same result using an MBR label. Verify -

```
$ bash lsblk -fp
```

Expected:

```
/dev/sda 224\234\224\200/dev/sda1 28G 224\224\224\200/dev/sda2 100G
```

(Optional) Format

****exFAT:****

```
$ sudo mkfs.exfat -n Part1 /dev/sda1 sudo mkfs.exfat -n Part2 /dev/sda2
```

****NTFS:****

```
$ sudo mkfs.ntfs -f -L Part1 /dev/sda1 sudo mkfs.ntfs -f -L Part2 /dev/sda2
```

(Optional) Encrypt with LUKS

Example 200\223 28 GB partition:

```
$ sudo cryptsetup luksFormat /dev/sda1
```

```
$ sudo cryptsetup open /dev/sda1 secure_usb1
```

```
$ sudo mkfs.exfat -n MySecure1 /dev/mapper/secure_usb1
```

```
$ sudo cryptsetup close secure_usb1
```

Example 200\223 100 GB partition:

```
$ sudo cryptsetup luksFormat /dev/sda2
```

```
$ sudo cryptsetup open /dev/sda2 secure_usb2
```

```
$ sudo mkfs.exfat -n MySecure2 /dev/mapper/secure_usb2
```

```
$ sudo cryptsetup close secure_usb2
```

CREATE NEW PARTITION WITHOUT LOSING EXISTING NTFS DATA**

(Non-destructive, NTFS-only; exFAT cannot be safely resized.)

```
$ lsblk -fp
```

```
206\222 Find something like '/dev/sda1 ntfs MyDrive'.
```

Unmount -

```
$ sudo umount /dev/sda1
```

```
Check filesystem sudo ntfsfix /dev/sda1
```

Shrink filesystem to 100 GB sudo ntfsresize --size 100G /dev/sda1

Shrinks only the filesystem, not the partition boundary.

Resize partition boundary to match

```
$ sudo parted /dev/sda (parted) print (parted) resizepart 1 100GB (parted)
```

quit

Create new partition in free space (~28 GB)

```
$ sudo parted /dev/sda --script mkpart primary 100GB 100%
```

Format the new partition

****exFAT:****

```
$ sudo mkfs.exfat -n NewPart /dev/sda2
```

****NTFS:****

```
$ sudo mkfs.ntfs -f -L NewPart /dev/sda2
```

Verify final layout

```
$ lsblk -fp
```

Expected:

```
/dev/sda1 ntfs MyDrive /dev/sda2 exfat NewPart
```

Important Notes

* Always back up important data first. * **exFAT cannot be resized non-destructively.** * Power loss during resizing may corrupt data on any filesystem.

FULL COMMAND SUMMARY

Identify lsblk -fp

--- If wiping & repartitioning

```
sudo wipefs --all /dev/sda sudo parted /dev/sda --script mklabel gpt mkpart primary 0% 28GB mkpart primary 28GB 100%
```

--- If keeping data (NTFS shrink)

```
sudo umount /dev/sda1 sudo ntfsfix /dev/sda1 sudo ntfsresize --size 100G
```

```
/dev/sda1 sudo parted /dev/sda resizepart 1 100GB sudo parted /dev/sda
```

```
--script mkpart primary 100GB 100%
```

--- Format new partition

```
sudo mkfs.exfat -n NewPart /dev/sda2 # or mkfs.ntfs -f -L NewPart /dev/sda2
```

--- Verify

```
lsblk -fp
```

```

----- yt-dlp -----
for the maximum best quality possible -
$ yt-dlp --no-playlist --no-call-home --no-cache-dir --no-cookies -f
bestvideo+bestaudio/best --merge-output-format mp4 --write-description
--write-info-json -P ~/Downloads -o "%(title).100s.%(ext)s" "video URL"

for specific resolution/quality , say to say 1080p (feel free to just change
the value of the video resolution/quality)-
$ yt-dlp --no-playlist --no-call-home --no-cache-dir --no-cookies -f
"bestvideo[height=1080]+bestaudio/best[height=1080]" --merge-output-format
mp4 --write-description --write-info-json -P ~/Downloads -o
"%(title).100s.%(ext)s" "VIDEO_URL"

and in order to download an entire playlists videos also for the maximum
best quality possible type this -
$ yt-dlp --no-call-home --no-cache-dir --no-cookies -f
bestvideo+bestaudio/best --merge-output-format mp4 --write-description
--write-info-json -P ~/Downloads -o "%(playlist_title)s/%(playlist_index)03d
- %(title).100s.%(ext)s" "PLAYLIST_URL"

and in order to download an entire playlists videos and for specific
resolution/quality, say to say 1080p (feel free to just change the value of
the video resolution/quality)-
$ yt-dlp --no-call-home --no-cache-dir --no-cookies -f
"bestvideo[height=1080]+bestaudio/best[height=1080]" --merge-output-format
mp4 --write-description --write-info-json -P ~/Downloads -o
"%(playlist_title)s/%(playlist_index)03d - %(title).100s.%(ext)s"
"PLAYLIST_URL"

for downloading audio Single files type this (just replace wav, mp3 with
flac for your targeted audio formats) -
$ yt-dlp --no-playlist --no-call-home --no-cache-dir --no-cookies -f
bestaudio --extract-audio --audio-format flac --audio-quality 0
--write-description --write-info-json -P ~/Downloads -o
"%(title).100s.%(ext)s" "VIDEO_URL"

for downloading audio playlists files type this (just replace wav, mp3 with
flac for your targeted audio formats) -
$ yt-dlp --no-call-home --no-cache-dir --no-cookies -f bestaudio
--extract-audio --audio-format flac --audio-quality 0 --write-description
--write-info-json -P ~/Downloads -o "%(playlist_title)s/%(playlist_index)03d
- %(title).100s.%(ext)s" "PLAYLIST_URL"

----- ffmpeg -----

without audio
$ ffmpeg -f x11grab -i :0.0 test1.mkv

with system's media audio
$ ffmpeg -f x11grab -i :0.0 -f alsa -i default test1.mkv

without microphone
$ ffmpeg -f x11grab -framerate 25 -video_size 1280x1024 -i :0.0 -f pulse -i
default -c:v libx264 -preset ultrafast -crf 25 -pix_fmt yuv420p -c:a aac
-b:a 128k ~/Desktop/screen_record_$(date +%F_%H-%M-%S).mkv

with microphone
$ ffmpeg -f x11grab -framerate 25 -video_size 1280x1024 -i :0.0 -f pulse -i
alsa_input.pci-0000_00_1f.3.analog-stereo -ac 1 -c:v ffv1 -level 3 -g 1
-pix_fmt yuv422p -c:a flac ~/Desktop/screen_record_$(date
+%F_%H-%M-%S)_lossless.mkv

with microphone+media's sound (kinda reaction sort of setup)
$ ffmpeg -f x11grab -framerate 25 -video_size 1280x1024 -i :0.0 -f pulse -i
alsa_input.pci-0000_00_1f.3.analog-stereo -f pulse -i
alsa_output.pci-0000_00_1f.3.analog-stereo.monitor -filter_complex
"[1:a][2:a]amix=inputs=2:duration=longest:dropout_transition=2[aout]" -map
0:v -map "[aout]" -ac 2 -c:v ffv1 -level 3 -g 1 -pix_fmt yuv422p -c:a flac
~/Desktop/screen_record_$(date +%F_%H-%M-%S)_mic+sys_lossless.mkv

```

Converting files from one format to another (MP4 to WebM).

```
$ ffmpeg -i input.mp4 output.webm
```

Extract the audio stream from a video file.

```
$ ffmpeg -i input.mp4 -vn output.mp3
```

Resize videos to specific dimensions for various playback requirements.

```
$ ffmpeg -i input.mp4 -vf scale=1280:720 output.mp4
```

Add subtitles to a video for better accessibility.

```
$ ffmpeg -i input.mp4 -vf "subtitles=subtitles.srt" output.mp4
```

Create a video slideshow from a series of images.

```
$ ffmpeg -framerate 1 -i img%03d.jpg output.mp4
```

----- Zathura -----

```
$ set cmd (ls /usr/bin /usr/local/bin /bin | rofi -dmenu -p "man page"); and man $cmd | g  
roff -mandoc -Tps | ps2pdf - "$HOME/$cmd.pdf"; and zathura "$HOME/$cmd.pdf" &
```

now after rofi pops up, type which manual you want as pdf and hit enter

To convert a txt file to pdf ---

```
$ sudo pacman -S enscript
```

1. Centered PDF ---

```
$ awk '{ pad=int((80-length($0))/2); if(pad<0) pad=0; printf "%*s\n", pad+length($0), $0  
}' "GNU.txt" \  
| enscript -B -f Courier10 -M A4 -p - | ps2pdf - "GNU_centered.pdf"
```

2. Left-aligned PDF ---

```
$ enscript -f Courier10 -B --no-header "GNU.txt" -p - | ps2pdf - "GNU_left_align.pdf"
```

----- PDF Arranger -----

```
$ sudo pacman -S pdfarranger
```

---- Some tweaks ----

```
### On Artix linux if you get any problem regarding keyring,  
namely keyring not found, transaction failed while updating /upgrading just  
right after freshly installed Artix linux!
```

```
$ sudo rm -rf /var/lib/pacman/local/gnupg $ sudo pacman-key --init $ sudo  
pacman-key --populate artix $ sudo pacman -Sy artix-keyring --noconfirm $  
sudo rm -rf /usr/lib/firmware/nvidia $ sudo pacman -Syu
```

```
### After installing adn opening at the very first time do these tweaks must
```

```
### ---
```

```
*****about:config settings start*****
```

Disable WebRTC:

WebRTC can give up your real IP even when using VPN or Tor. Type in

```
media.peerconnection.enabled = false
```

Enable fingerprint resistance: With this alone we pretty much negate the need for canvas defender, or any other fingerprint blocking addon.

```
privacy.resistfingerprinting = true
```

3DES Cypher:

3DES has known security flaws

```
security.ssl3.rsa_des_ede3_sha = false
```

Require Safe Negotiation: Optimize SSL

```
security.ssl.require_safe_negotiation = true
```

Disable TLS 1.0, 1.1

```
security.tls.version.min = 3
enables TLS 1.3
tls.version.max = 4
```

Disable 0 round trip time to better secure your forward secrecy

```
security.tls.enable_0rtt_data = false
```

Disable Automatic Formfill

```
browser.formfill.enable = false
```

Disable disk caching

```
browser.cache.disk.enable = false
browser.cache.disk_cache_ssl = false
browser.cache.memory.enable = false
browser.cache.offline.enable = false
browser.cache.insecure.enable = false
```

Disable geolocation services

```
geo.enabled = false
```

Disable plugin scanning.
Can improve functionality, as some sites scan for adblockers and script blockers. Should be used even on non-hardened firefox.

```
plugin.scan.plid.all = false
```

Disable ALL telemetry

```
browser.newtabpage.activity-stream.feeds.telemetry browser.newtabpage.activity-stream.tel
emetry = false
browser.pingcentre.telemetry = false
devtools.onboarding.telemetry-logged = false
media.wmf.deblacklisting-for-telemetry-in-gpu-process = false
toolkit.telemetry.archive.enabled = false
toolkit.telemetry.bhrping.enabled = false
toolkit.telemetry.firstshutdownping.enabled = false
toolkit.telemetry.hybridcontent.enabled = false
toolkit.telemetry.newprofileping.enabled = false
toolkit.telemetry.unified = false
toolkit.telemetry.updateping.enabled = false
toolkit.telemetry.shutdownpingsender.enabled = false
```

Disable WebGL
Allows direct access to GPU.

```
webgl.disabled = true
```

Enable first-party isolation
Prevents browsers from making requests outside of the primary domain of the website. Prevents supercookies. may cause websites that rely on 3rd party scripts and libraries to break, however those are generally only used for tracking so fuck em anyway.

```
privacy.firstparty.isolate = true
```

Disable TLS false start

```
security.ssl.enable_false_start = false
*****about:config settings end*****
```

Uninstall any packages/Programes ---

```
$ sudo pacman -Rns (package name)
```

--- create bootable drive ---

```
$ lsblk (make sure the targeted usb drive is present/mounted) $ sudo dd
if=/home/az/Desktop/artix-xfce-runit-20250407-x86_64.iso of=/dev/sda bs=4M
status=progress oflag=sync $ sync
```

--- Bunch of cool commands ---

```
# disable sshrootlogin---
$ nvim /etc/ssh/sshd_config
inside, look for a line named - "#PermitRootLogin"
there it may set to yes or prohibit-password.
just set it to no, and save and maybe reboot maybe!

### Uninstall any packages/Programes ---
$ sudo pacman -Rns (package name)

### view installed programs/packages
$ echo "==== PACMAN (explicit user installs) ====" pacman -Qe

echo echo "==== AUR (foreign packages via yay) ====" yay -Qm

echo echo "==== FLATPAK (installed apps) ====" flatpak list --app

echo echo "==== APPIMAGE (portable executables) ====" find ~ -type f -iname
"*.AppImage"

echo echo "==== MANUAL (/usr/local, /opt per FHS) ====" ls /usr/local /opt
2>/dev/null

.....
# get package details ---
$ sudo pacman -Ss package_name

# remove package including its dependencies but not those that are being
used on other programmes ---
$ sudo pacman -Rs package_name

# remove package including its dependencies wheather those that are being
used on other programmes or not ---
$ sudo pacman -Rns package_name
$ sudo pacman -Rns $(pacman -package_name)    --- yet dunno what it actually does!

# to install packages ---
$ sudo pacman -S package_name

$ ls -a | wc -l
the less, the better (below 20 is kinda ideal)

$ history

$ grep -i now Itâ\200\231s\ GNU_Linux\ btw.txt -wo
$ grep -i now Itâ\200\231s\ GNU_Linux\ btw.txt -wo | wc -l

$ free -h
also,
$ free -h | awk '/^Mem:/ {print $3 "/" $2}'

$ sudo pacman -S wget $ sudo pacman -S curl

$ wget -O cheatsh_1.txt "https://cheat.sh/magick?T"

also,

man magick
man ffmpeg
info magick
info ffmpeg

1) Show your most-frequently used shell commands (counts, sorted) awk
'{cmd[$1]++} END {for (c in cmd) print cmd[c], c | "sort -rn"}'
~/.bash_history | head -n 20
```

2) Show most frequently installed or removed packages (Debian/Ubuntu APT history) [-r /var/log/apt/history.log] && sudo awk '/COMMAND/ {print \$1}' /var/log/apt/history.log | sort | uniq -c | sort -rn | head -n 20 || echo "/var/log/apt/history.log not found or not readable"

3) List commands you typed wrong most often (from your shell history) grep -F 'command not found' ~/.bash_history 2>/dev/null | sed 's/.*command not found: */' | sort | uniq -c | sort -rn | head -n 20

4) Show the 10 most recently changed config files under ~/.config find ~/.config -type f -printf '%T@ %p\n' 2>/dev/null | sort -n | tail -n 10 | awk '{sub(/^[0-9]+\.[0-9]+ /,""); print}'

5) List the 10 largest files in your home directory (by size), paths only find "\$HOME" -xdev -type f -printf '%s %p\n' 2>/dev/null | sort -nr | head -n 10 | awk '{sub(/^[0-9]+ /,""); print}'

6) Find files in your home modified in the last 24 hours with timestamps find "\$HOME" -type f -mtime -1 -printf '%TY-%Tm-%Td %TH:%TM %p\n' 2>/dev/null | sort

7) Show all mounted block devices with filesystem type, size and mountpoint lsblk -o NAME,SIZE,FSTYPE,MOUNTPOINT | grep -v '^loop'

8) Show your public/external IP address (single line) curl -s https://ipinfo.io/ip || curl -s https://ifconfig.co

9) List currently active (running) systemd services systemctl list-units --type=service --state=running

10) Show the top CPU-consuming processes (PID, command, %CPU) ps -eo pid,comm,%cpu --sort=-%cpu | head -n 11

files related-

\$ pwd --- print working directory -shows where you are at now \$ cd ---change directory \$ ls \$ ls --color \$ ls -a --- shows hidden files namely .config, .local .gnupg \$ ls -al \$ ls .config \$ mkdir ~/newfolder \$ rmdir ~/newfolder \$ mkdir -p ~/aaa/bbb \$ mkdir -p ~/Music/new

Get manual of anything-

\$ man -(programme name) \$ man man \$ man cd \$ info \$ info ls \$ info newsboat \$ sensors

Know system's OSs specs or info:

\$ uname -a \$ uname --help

then,

\$ uname -(there will be mentioned many letters to know a specific info in the help section)

Storage related-

\$ free \$ free -h \$ free -h | awk '/^Mem:/ {print \$3 "/" \$2}' \$ df \$ df -h \$ ps axch -o cmd,%mem --sort=-%mem \$ ps axch -o cmd,%mem --sort=-%mem | head \$ ps axch -o cmd:15,%cpu --sort=-%cpu | head

Remove .trash1000 folders items from your USB drive---

\$ sudo rm -rf /run/media/\$USER/YOUR_USB_NAME/.Trash-1000/*

replace "YOUR_USB_NAME" with your actual USB name, to know exactly run the command ---

\$ ls /run/media/\$USER/

For secure file deletion use -

shred (file name) command also can be used shred -u (file name) for both shredding and then delete the file. Additionally,

\$ sudo dd if=/dev/urandom of=/mnt/usb/junk status=progress (gotta check again)

```
### Remove BG from pretty much any images- (gotta verify again)
$ convert -transparent (BG color, namenly white, red, blue, black) (file
name)
```

```
$ convert -transparent (BG color, namenly white, red, blue, black) -fuzz
(60/50...%)w(or without w) (file name)
```

```
$ convert -trim -transparent (BG color, namenly white, red, blue, black)
-fuzz (file name)
```

```
### CLI method for stabilizing any videos -
Source--- https://github.com/georgmartius/vid.stab
```

.....

Shift+PageUp or Shift+PageDown - Scroll up or down through shell output Home or End - Move to the start or end of a line, respectively Tab - Autocomplete a file name, directory name or command name. Ctrl+c - End a running process Ctrl+d - End-Of-File (EOF) character (usually ends a process or signifies the end of input data) Ctrl+z - Send the currently running process to the background Ctrl+l - Clear the screen, same as running the clear command If you're using BASH as your shell (most distros default to this) you can customize it to your liking.

Common commands pwd - Print Working Directory. Outputs the full path of the current directory. cd - Change Directory. Used to switch to a different directory. ls - List. Lists files and directories in the current directory. ls -a Lists "hidden" files and directories also. Hidden files in linux are preceded with a full stop. ls -l Lists further information about the files, including size, modify date, owner and permissions. ls -t List by modify date, with the most recently modified files at the top. cp - Copy a file/directory. cp -r Descend into directories (recursively copy all directory contents). mv - Move or rename a file. rm - Delete a file. rm -r Delete directory contents. (Never rm -r /) chmod - Changes file permissions for the Owner, Group and Others. chown - Changes the owner of a file. chgrp - Changes the group of a file. date - Display current date and time. who - Displays a list of currently logged in users. echo - Prints to stdout. cat - Concatenates text to stdout. grep - Searches for strings in a file or stdin. su - Switch user. Defaults to root. sudo - Similar to su but uses your own password instead. Lets you configure access in /etc/sudoers. passwd - Change password. Defaults to current account. rsync - Synchronizes files and directories. halt - halt the system halt -p - also shut off the power systemctl - Interface to systemd. systemctl poweroff - Shutdown. systemctl reboot - Reboot. systemctl start service Start a service. systemctl stop service Stop a service. journalctl - Interface to systemd logging. Piping and redirection There are a number of little quirks that the shell has that gives it more functionality. Piping takes the stdout of the left program and connects it (i.e. pipes it) into stdin of the right program with the pipe operator.

For example:

```
Count number of words in helloworld.txt $ cat helloworld.txt Redirection
directs data in and out of files, i.e.:
Redirect stdout to file $ echo "Hello world" > helloworld.txt Redirect
stdout to the end of a file $ echo "world." >> helloworld.txt Redirect a file to
stdin $ cat < helloworld.txt
```

----- Steganography -----

```
### Steghide:
```

With Images

To embed a secret file in an Image file use this Command:

```
$ steghide embed -cf (image file's full name within extension) -ef
(secretfile, the file that wanna hide inside the image file)
```

To extract the secret file use this command:

```
$ steghide extract -sf (image file's full name within extension)
```

With Audio

To embed a secret file in an Audio file use this Command:

```
$ steghide embed -cf (Audio file's full name within extension) -ef  
(secretfile, the file that wanna hide inside the image file)
```

To extract the secret file use this command:

```
$ steghide extract -sf (Audio file's full name within extension)
```

Screen mirroring:

for wireless open Terminal and ---

```
$ adb devices
```

it should shows like this:

List of devices attached 07302331BT003558device

now, type and run-

```
$ adb shell ip -f inet addr show wlan0
```

it should shows like this:

```
32: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP  
group default qlen 3000 inet 192.168.0.101/24 brd 192.168.0.255 scope global  
wlan0 valid_lft forever preferred_lft forever
```

now, type and run-

```
$ adb tcpip 5555
```

it should shows like this:

restarting in TCP mode port: 5555

```
$ now, type and run- adb connect (device IP):5555
```

it should shows like this: connected to 192.168.0.101:5555

Offline Internet !!!

<https://download.kiwix.org/zim/wikipedia/> <https://download.kiwix.org/zim/>

<https://dumps.wikimedia.org/other/kiwix/zim/wikipedia/>

<https://medlineplus.gov/xml.html> <https://medlineplus.gov/xml.html> Download

as much as you want use kiwix app or software and boom enjoy

Slow-scan television (SSTV)

.....

---- Worth looking at ----

https://wiki.archlinux.org/title/Main_page

<https://stackoverflow.com/questions> <https://unix.stackexchange.com>

<https://www.reddit.com/r/commandline> <https://www.reddit.com/r/Artix>

<https://github.com/mayfrost/guides/blob/master/CHECKLIST.md>

<https://igwiki.lyci.de/wiki/Security>

.....

site:instagram.com "linux and open source"

"united states" "@gmail.com"