An Open Source collection of common problems and solutions for *nix users

Arch Linux

Links useful during an arch install

1, 2, 3, 4, 5, 6,

Pacman commands

- `pacman -Syy`: update repositories
- `pacman -Su`: update packages
- `pacman -Qdtq`: list unused/orphan packages
- `pacman -R`: remove a package
- `pacman -S`: install a package
- `pacman -R $(pacman -Qdtq)`: remove orphaned packages recursively
- `pacman -Rns`: remove packages and its dependencies recursively
- `pacman -Rcs`: remove packages and its dependents recursively
- `pacman -Ql`: list all package's files and locations
- `pacman -Qqe`: list all installed packages
- `pacman -Qc`: view package changelog
- `pacman -Qm`: list packages not present in official repositories

If there are update related errors that you do not understand, do not panic issue a full update with the following commands: `sudo pacman -Syy` and then `sudo pacman -Su`

If you get an error similar to `could not unlock database` when trying to issue a command that means that another program is using the package manager. If you feel like taking the risk of breaking your system try manually (and forcefully) removing the transaction lock file by issuing `rm -rf /var/lib/pacman/db.lck`

In the unfortunate event that all of a sudden everything fucked up bad (like no xorg or wayland or login session ...), check the pacman log at `/var/log/pacman.log` to see what went wrong and what got removed

AUR pacman wrappers

- pikaur
- yay

Pacman GUI frontends (I advice you to not use them)

- `pamac`: based on gtk and it is the same GUI frontend in the gnome and KDE releases of manjaro, install `pamac-aur` or `pamac-aur-git` from the AUR.
- `octopi`: is a lightweight GUI frontend for pacman and it is one of the most used ones although it isn't as visually appealing as pamac and requires a gtk permission manager.

Network management

Sources

1 2 3

Arch uses netctl to manage connections, all the config files and examples are under `/etc/netctl/` and `/etc/netctl/examples`
NOTE: This is valid for every system that uses systemd(ick)

Void Linux

Hold package updates

- Hold: `xbps-pkgdb -m hold <pkg>`
- Unhold: `xbps-pkgdb -m unhold <pkg>`
- List packages on hold: `xbps-query --list-hold-pkgs`

Sysadmin stuff

Fixing mandoc.db permission denied

Source: link

If you ran `sudo makewatis ⋯` then it is possible that the file permissions of `/usr/share/man/mandoc.db` and `/usr/local/share/man/mandoc.db` were changed from 644 (rw-r—r--) to 600 (rw-------), giving the error when viewing man pages `⋯ mandoc.db permission denied` or `data changed but could not update mandoc.db` etc. To fix this simply do:

```
sudo chmod 644 /usr/share/man/mandoc.db
sudo chmod 644 /usr/local/share/man/mandoc.db
```

Important packages

- `acpi` gives information and control on battery and power status, it can also handle power related events (lid open/close, power button press, etc.)
- `thinkpad_acpi` acpi addon specific to thinkpads

Qt in wayland

To run QT prigrams in wayland install `qt5-wayland` and add the followind line somewhere in your profile (.profile .zprofile .bash_profile etc.) `export QT_QPA_PLATFORM=wayland`

Monitor Tor network usage

install the `arm` package and run it

Basic commands

- `grep`: find the lines which contain the input string:
  `ps ax | grep <query>` and it will output the line containing the query
  `grep <query> file.txt` search matches in file
  `grep -i <query> file.txt` case insensitive search
- `ps`: Ps is a UNIX tool used to get information about the current status of the system.
  The most basic and or useful usage is to view all the running processes and the corresponding UUIDs, that is done by appending `ax` to ps:
  `ps ax`
- `kill`: kill the specified process given its UUID:
  `kill <UUID>`
- `killall`: kill all the processes which name matches the arguments:
  `killall <proc name>`
- `|` and `>`: These are both "pipe" functions, they can pipe the output of a script or program into something (via std i/o), but they are used differently in the sense that `|` is used to pipe the output into another program, example:
  `ps ax | grep xorg` this outputs all the runnng processes into grep as input, which in turn filters the result.
  On the other hand `>` is used to pipe the output into a file, example:
  `ls -la > ls.txt` this puts the output of ls into ls.txt which we can then read. **NOTE**: `>` replaces everything that was in the file
- `<`: same thing as `>` but in "reverse"
- `>>` and `<<`: same as `<` and `>` but appends instead of replacing
- `whoami` and `groups`: they respectively output the current user and groups
  NOTE: one alternative to `whoami` is `echo $USER` which can be used in scripts
- `find`: Basic syntax:
  `find <dir> [OPTIONS]`

Some options:

- `-user <user>`: find files owned by a particular user
- `-group <group>`: find files owned by a particular group
- `-ls`: list results in "ls" format:
- `-name <filename>`: find a specific file or pattern (*.txt)

Group actions

- Adding a user to a group:
  `usermod -a -G group user` then reboot
- Listing all groups:
  `cut -d: -f1 /etc/group | sort`
- Deleting a group:
  `groupdel <group name>` then reboot

Users actions

- Creating a new user:
  `useradd -m user-name`
- Deleting an existing user:
  `userdel user-name`

Generating locales

1. Add locales by uncommenting them in `/etc/locale.gen`
2. generate locales by running `locale-gen`
3. Reboot

NOTE: some languages (russian, chinese, japanese) require specific fonts to be installed refer to: link for an incomplete list of specific fonts

Getting the graphics drivers

For Intel graphics cards

Source: link
Install `mesa` on arch or `mesa-intel-dri` on void and `xf86-video-intel`

For nvidia graphics cards

Follow the steps in: link

For AMD graphics cards

Install: `mesa` on arch or `mesa-ati-dri` on void and `xf86-video-ati` for Xorg hardware acceleration support

Adding entropy to your system (faster boot times)

Source: link
This is useful as the kernel's built-in random number generator is very slow, as such it makes tasks like loading the login manager a pain. To solve this problem you can install some pseudo-random number generators such as: * `rng-tools` secure but uses more CPU * `haveged` fast and lightweight but less secure **NOTE**: Keep in mind that these are not for secure systems and remember to start and enable the service rngd.service

NTFS partitions support

Source: link
Install `ntfs-3g`

Listing all installed packages

With apt/apt-get: `apt list --installed`
With pacman: `pacman -Qqe`
With xbps: `xbps-query -m`

Changing the default shell

- `chsh -l` list all installed shells and respective path
- `chsh -s <path-to-shell>` set the shell for the current user

Changing keyboard layout

Source: link
In a tty (no graphical envirnment line ssh) type: `layout <layout>`
In a graphical environment (or terminal emulator) type: `setxkbmap <layout>`

-To run .jar files in cli you must use "java -jar"

Disk and volume info

Graphical

- `gparted` graphical tool for managing disks and partitions

Terminal

Partitioning tools

- `fdisk`
- `parted`

Listing drives and mount points

- `fdisk -l` more info

- `lsblk`

Listing drive UUIDs

- `blkid`

Automatically mount volumes

Install `udevil` and start `devmon` at login/boot

Check disk health (if available)

Install `smartmontools`, then check the drive's compatibility with

```
smartctl -c /dev/sdX
```

Then either run a short test

```
smartctl -t short /dev/sdX
```

Or a long test

```
smartctl -t long /dev/sdX
```

lastly run `smartctl -H /dev/sdX` to get the results

Change default file openers

Source: link . Create a file (if not present) in `$HOME/.config` called `mimeapps.list` . Search your file extension in link . Add or modify the entry to that mimetype to be opened with the desired `<applications>.desktop` **IF** `<application>.desktop` **IS NOT PRESENT**: . Look again in `/usr/share/applications` . If it is not present make one and put it into: `~/.local/share/applications/` with the format:

```
[Desktop Entry]
Name=Xpdf
Comment=Views Adobe PDF (acrobat) files
Exec=xpdf %f
Terminal=false
Type=Application
Icon=xpdf
Categories=Office;
MimeType=application/pdf;
```

NOTE: `Exec=` are the execution parameters (-b, -l, etc.)

Recompiling compilers for zerynth

If you are facing compiling errors in zerynth, one option is to manually recompile the compiler for the board/platform. Compilers are locarted at `~/.zerynth2/sys`
For example to replace the `xtensa-lx106` compiler: (esp8266) . Move the old compiler folder (located at `~/.zerynth2/sys/xtensa-lx106`) somewhere safe . Clone and compile (as standalone) new compiler in a temporary folder, guide here link . Move the fresh compiler folder (xtensa-lx106-elf) in `~/.zerynth2/sys` and rename it as the original one (xtensa-lx106/) . Copy the old package.json in the new compiler folder

NOTE: Better yet is to wait for an official fix and report the bug on the troubleshooting section
Link for the `xtensa-lx6` compiler: (esp32) link

Various permission errors

Arduino permissions

Source: link
`ls -la /dev/tty*` and see which group TTYs are in, then add yourself to that group:
`usermod -a -G <group> <user>` or just use `$USER` to add current user, then reboot.

Wireshark permissions

```
usermod -a -G wireshark $USER
```

Pen drive in read only

Source: link

1. Unmount the pen drive (`/dev/sdXx`)
2. Run `dosfsck -a /dev/sdXx`
3. Remount

Cannot scroll down in VIM on st

Add `set ttymouse=sgr` to your .vimrc

DEL key not working in st

source: link
Add `tput smkx` in your zshrc/bashrc or equivalent

Theming gnome

download the themes packages here and unzip them then depending if it is an icon theme or an normal theme place them in `/usr/share/icons` or `/usr/share/themes` accordingly

Printing documents

Source: link
Install `cups` then start and enable the cupsd service/daemon or socket if you want on-demand activation.
To add and configure a printer either do it in your print manager or trough the cups web interface at the address: `localhost:631`
NOTE: adding and modifying printers requires administator (root) permissions, so in the web interface, when asked, insert the root credentials

Adding executables and shortcuts

Sources

- 1
- 2

You have to add them to your $PATH, example: "export PATH="$PATH:$HOME/esp/xtensa-esp32-elf/bin" Quote "If you just type export PATH=$PATH:</path/to/file> at the command line it will only last for the length of the session. If you want to change it permanently add export PATH=$PATH:</path/to/file> to your ~/.bashrc file (just at the end is fine)."

-Eagle CAD dark theme fix (KDE): Refers to: https://forum.kde.org/viewtopic.php?f=17&t=136316 https://forums.autodesk.com/t5/eagle-forum/kubuntu-18-04-kde-dark-theme/td-p/8188466 To fix this behavior you first have to apply the default light theme of KDE (breeze), then copy the kdeglobals file located under ~/.kde4/share/ create a folder named ~/.config_light/ and paste the kdeglobals in it, rename kdeglobals to config_light, then finally add to the eagle start command: "export XDG_CONFIG_HOME=/home/ale/.config_light/ ;" (where ale is your username) or add an alias to the eagle command where it becomes: "export XDG_CONFIG_HOME=/home/ale/.config_light/ ; eagle" NOTE: this will change the default settings folder to ~/.config_light, all previous setting will be lost but not the libraries or projects

-Use "clamav" as antivirus: "https://wiki.archlinux.org/index.php/ClamAV", to update use "freshclam" as sudo and to scan "clamscan --recursive --infected /path/to/something"

-Some git stuff: Creating a repo and pushing the first commit: "https://help.github.com/articles/adding-an-existing-project-to-github-using-the-command-line/" first init the repo with "git init" then if you want to sync with a remote repo basically it's a matter of defining the remote (origin) repo "git remote add origin <repo URL>" before committing add staged changes with "git add /roba" or everything with "git add ." committing the changes in local repo "git commit -m "message" " pushing before first pull "git pull origin master --allow-unrelated-histories" the last part is just to merge the repos (if needed) and finally pushing the changes "git push origin master", or in atom just publish Using 2 factor auth. in command line and atom: "https://help.github.com/articles/creating-a-personal-access-token-for-the-command-line/" basically you have to use a access token instead of the password, these are unique and you can only see them once, so be careful with them! Downloading other branches: In your folder repo open a git bash or a terminal (for linux masterrace) then, admitted that you have already downloaded and synced the master branch, and type "git checkout -t origin/branchname" this will download and sync the

branch

-Powertop usage: Arch page: https://wiki.archlinux.org/index.php/powertop To start powertop use "sudo powertop" For the first calibration use "sudo powertop --calibrate", NOTE: it takes a few minutes and during that time the screen may go (it does) black for a few minutes too, just let it run. To set everything to "good" do "sudo powertop --auto-tune", to make the auto-tune start at boot refer to the Arch wiki or "how to create a sysemd service" NOTE: in order to apply changes and stuff you have to leave it running for some time to let it take its measures and stuff

-Power management: Arch page: https://wiki.archlinux.org/index.php/Power_management#Power_management_with_systemd To change the actions to take when power button or lid switch events occur: modify the conf. file at /etc/systemd/logind.conf or /etc/systemd/logind.conf.d/*.conf

-Adding executables not located in /bin/ (downloaded from internet): Refers to: https://askubuntu.com/questions/322772/how-do-i-add-an-executable-to-my-search-path https://unix.stackexchange.com/questions/3809/how-can-i-make-a-program-executable-from-everywhere Adding them momentarily: Add them to your $PATH, example: "export PATH="$PATH:$HOME/esp/xtensa-esp32-elf/bin" Adding them at the start of the session: Using bashrc: Add export PATH=$PATH:</path/to/file> (the bin/ folder) to your ~/.bashrc file (just at the end is fine) Using cron: Add export PATH=$PATH:</path/to/file> to your crontab file: Open your crontab file using "crontab -e" for current user or "crontab -e -u username" fo others At the end of the file add "@reboot export PATH=$PATH:</path/to/file>" NOTE: probably it is best do to add the command in the "su" cron file

-Checking system errors: Refers to: https://wiki.archlinux.org/index.php/Systemd#Journal https://www.digitalocean.com/community/tutorials/how-to-use-journalctl-to-view-and-manipulate-systemd-logs If your distro users systemctl the easiest way to check errors is to look up the recent events in the system logs to do that type "journalctl"

-Getting the right dpi: (only applies to xorg) Resources: https://wiki.archlinux.org/index.php/Xorg#Display_size_and_DPI First of all get the optimal dpi for your screen: http://dpi.lv/ Then get the current dpi that the server is set on: (requires dpyinfo) xdpyinfo | grep -B2 resolution if it si already set to the right dpi you are good to go, else add to your xinitrc or xprofile (if using a display manager): xradr --dpi <your dpi>

-Using android devices: Source: https://wiki.archlinux.org/index.php/Media_Transfer_Protocol Android devices use the MTP interface, to be able to access them in your file manager it is usually required to install the packages: "libmtp" "gvfs-mtp" and "gvfs-gphoto2" If running a KDE installation with dolphin mtp support is integrated into kio-extras (dolphin dependency) NOTE: running multiple mtp tools can cause errors (gvfs-mtp and kio-extras) NOTE: Apple devices require the "gvfs-afc" package, and the "gamin" package is also recommended

-File previews on KDE (Dolphin): Please refer to: https://wiki.archlinux.org/index.php/Dolphin#File_previews for the list of packages suitable for different file types

-Windows: Adding programs to PATH: Source: https://www.howtogeek.com/118594/how-to-edit-your-system-path-for-easy-command-line-access/ Open: Control panel → system → advanced system settings → environment variables On windows 7 or 8: Set the variable name to "Path" Add the path to the program preceded by ";" ex. "...;C:\path\to\something" On windows 10: Click on "New" Enter the path to the program Retrieving the windows product key: Open a console with

admin privileges Type "wmic path softwarelicensingservice get OA3xOriginalProductKey"

-Creating a systemd (systemctl) service (aka how to make a program start at boot): Guides and references: https://askubuntu.com/questions/112705/how-do-i-make-powertop-changes-permanent https://wiki.archlinux.org/index.php/powertop https://askubuntu.com/questions/919054/how-do-i-run-a-single-command-at-startup-using-systemd Create a file under /etc/systemd/system/ and call it <whatever>.service Then follow the formatting { [Unit] Description=PowerTOP auto tune

```
[Service]
Type=idle                                 # Not necessary
Environment="TERM=dumb"                   # Not necessary
ExecStart=PATHTOEXECUTABLE --OPTIONS
```

```
  [Install]
  WantedBy=multi-user.target
}
And then enable it using "systemctl whatever.service enable"
Other method: using crontab
  Reference: https://wiki.archlinux.org/index.php/cron#Crontab_format
  Install cronie then put your command into the crontab file using
    "crontab -e"
  Then follow the formatting according to the reference to specify when the command
    is executed
```

LAPTOP OPTIMIZATION

-Undervolting the CPU: Refers to: https://wiki.archlinux.org/index.php/Undervolting_CPU https://wiki.archlinux.org/index.php/Stress_testing https://wiki.archlinux.org/index.php/benchmarking WARNING: Overvolting and overheating can result in permanent damage On arch install the package "intel-undervolt" from AUR and edit it's configuration file located in /etc/intel-undervolt.conf, changing the values from 0 to -number results in that number of millivolts taken from the CPU voltage. NOTE: on voidlinux use the pip3 package undervolt follow: https://wiki.voidlinux.org/Undervolting for more info. The following are the parts affected: "CPU" changes the CPU core voltage "GPU" changes the integrated GPU voltage "Cache" changes the cache voltage "System agent" changes the controller, RAM and PCI voltages "Analog IO" changes the various sensors voltage NOTE: changing the System Agent and Analog IO can lead to various errors such as inaccurate readings from sensors crashes and possibly corruption, so it is greatly discouraged One method to undervolt your CPU or GPU is to gradually step down the voltage in 5 or 10 millivolt steps stress testing the system in between for several minutes to ensure the system's stability Some tools used to stress test your system are: "stress" tool written in C used for memory, CPU and disk IO tests "mprime" AUR, tool for stress testing RAM and CPU "unigine-heaven" and the unigine series, although benchmarks they are a great way to stress test the integrated GPUs of laptops

-i915 (graphics) tweaks: Refers to: https://gist.github.com/Brainiarc7/aa43570f512906e882ad6cdd835efe57 There are a number of tweaks that can save some watts

regarding the integrated graphics card to enable them you can create a file called "/etc/modprobe.d/i915.conf" containing all of them, you can get the complete list of available parameters and their description by running "modinfo -p i915". The file must be filled following the format: options i915 "parameter" Some safe-to-use parameters are: "enable_fbc=1" enables frame buffer compression "enable_dc=1" enables power saving "disable_power_well=0" enables power well Other parameters may include "enable_psr=1" but that doesn't always work on pre-skylake hardware and on newer is straight up broken, it also depends on whether or not the display supports it, for further information refer to: https://hansdegoede.livejournal.com/18653.html NOTE: as every parameter has the potential to cause weird issues it is recommended to test them first by appending them to the boot command (either in GRUB, rEFInd or systemd-boot) using the syntax: i915.enable_dc=1 After having enabled the options you can check if they were correctly applied by running "systool -m i915 -av" as root (systool is part of sysfsutils) NOTE:on voidlinux the modprobe .conf files must go inside /lib/modprobe.d/ and instead of using mkinitcpio to regenerate the initramfs use "dracut --force"

-Screen tearing: Refers to: https://wiki.archlinux.org/index.php/Intel_graphics#Tearing Enable the option "TearFree" of the driver: File: /etc/X11/xorg.conf.d/20-intel.conf Content: Section "Device" Identifier "Intel Graphics" Driver "intel"

```
Option "TearFree" "true"
EndSection
```

-General tweaks: Use "haveged" instead of "rng-tools" Under KDE uninstall everything PIM or akonadi related Minimize the number of needed applications and daemons

-Better battery management: Refers to: https://wiki.archlinux.org/index.php/TLP TLP is a tool daemon which manages the power consumption of your laptop and automatically enables or disables power saving features. To use it install: "tlp", "tlp-rdw" (for radio devices), "acpi_call" (for thinkpads), "tlpui-git" (AUR, GUI frontend) After installing all of the beforementioned packeges be sure to start and enable tlp using systemctl: "sysytemctl start tlp.service" "systemctl enable tlp.service" "systemctl start tlp-sleep.service" systemctl enable tlp-sleep.service" and finally if you installed the radio davice wizard (tlp-rdw) run: "systemctl enable NetworkManager-dispatcher.service" "systemctl mask systemd-rfkill.service" "systemctl mask systemd-rfkill.socket" Reboot and you're all set

NETWORK SECTION

-Network managers: NetworkManager: default in many distros pretty good but resorce heavy (also depends on systemd?) void: https://docs.voidlinux.org/config/network/networkmanager.html arch: https://wiki.archlinux.org/index.php/NetworkManager Connman: small and fast, does all you need and doesn't hog on resources and has vpn support void: https://wiki.voidlinux.org/Connman arch: https://wiki.archlinux.org/index.php/ConnMan Wicd: outdated by now and written in python 2.7, still pretty good for simple configurations

-Gathering information on network cards ie IP, MAC, status, etc. With ifconfig, just type "ifconfig" and it will give: ether: the MAC address inet: the network ip address netmatsk: the netmask duh

inet6: the network ipv6 With iwconfig (specifically for wireless cards), just type "iwconfig" and it will give: Mode: monitor or managed ESSID: the ESSID (name) of the connected network Frequency Access Point: the MAC address of the AP Signal level: the power/quality of the signal Among other informations Sites for public IP discovery and information gathering: https://www.iplocation.net/ http://www.whatsmyip.org/ With ip (standard) ip link show

-Changing the MAC address: Using macchanger: The general usage is "macchanger [OPTIONS] interface", the most common options being -A for a random MAC, -r to reset to the original one, -a to change it to a random MAC of some type (same vendor), -b to set the "locally administered" bit set to false or -m XX:XX:XX:XX:XX:XX to manually set the MAC address. Using ifconfig: First disable the interface using "ifconfig <interface> down" Then change the MAC address using "ifconfig <interface> hw ether 00:XX:XX:XX:XX:XX" it has to begin with 00:* Then finally re-enable the interface with "ifconfig <interface> up" To prevent the MAC address from reverting to its original state you can configure the network manager to not scan for other networks using a random MAC address (that is done for security by many OSs and also by phones), to achieve this you have to change the conf. file for your network manager: For GNOME (kali) add { [device] wifi.scan-rand-mac-address=preserve

```
  [connection]
  ethernet.cloned-mac-address=preserve
  wifi.cloned-mac-address=preserve
}
to /etc/NetworkManager/NetworkManager.conf
```

-Changing interfaces to monitor mode: Not mandatory but best use is to first use "airmon-ng check kill", this kills all processes that could interfere with the process and later with the attacks Using airmon-ng: Simply use "airmon-ng start <interface>" to put it in monitor mode And "airmon-ng stop <interface>" to revert it to managed NOTE: when using airmon-ng it will change the interface name by adding *mon to its name for example wlan0 becomes wlan0mon Using iwconfig (useful since first method is not always reliable): Disable interface using "ifconfig <interface> down" or "ifdown <interface>" Change to monitor mode with "iwconfig <interface> mode monitor" Re-enable interface with "ifconfig <interface> up" or "ifup <interface>" To revert it back to managed disable the interface then do "iwconfig <interface> mode managed" then re-enable the interface

-Restarting network interfaces: Without ssh: Disable and re-enable the interface with "ifdown <interface>" and then "ifup <interface>" or "ifconfig <interface> down" and then "ifconfig <interface> up" On ssh: Non systemd "/etc/init.d/networking restart" Systemd (Arch) "systemctl restart NetworkManager"

-Enabling IP forwarding: Arch page: https://wiki.archlinux.org/index.php/Internet_sharing "echo 1 > /proc/sys/net/ipv4/ip_forward" To revert it back: "echo 0 > /proc/sys/net/ipv4/ip_forward"

-Manually changing ip address of a given network card: run: "ip addr add <ip>/<mask> dev <interface>" where ip is the desired ip and mask is the desired network mask in 2 digit format, ex: "192.168.1.2/24" where 24 expands to 255.255.255.0