

Arch Linux – Setup Guide

v2024-11-01

Table of Contents

0 INTRO.....	4
0.1 Recommended prerequisites.....	4
0.2 Key features – Realization of goals.....	4
0.3 Styles & Meaning.....	6
0.4 Notes.....	6
1 Pre-boot.....	7
1.1 Acquire an installation image.....	7
1.2 Prepare the USB flash installation medium.....	7
1.3 Set up UEFI.....	8
1.4 Notes for MS Windows dual booters.....	8
1.5 For Oracle VM VirtualBox: Port Forwarding.....	8
2 Pre-installation.....	9
2.1 Boot from USB flash installation medium.....	9
2.2 Set the keyboard layout.....	9
2.3 Verify the EFI-boot mode.....	9
2.4 Connect to the internet.....	9
2.5 Set temporary root password (for ssh root login).....	9
2.6 Connect via ssh (as <i>root</i>).....	9
2.7 Update the system clock.....	10
2.8 Preparing the disk.....	11
2.9 Partition the disk using gdisk – ESP + Root.....	14
2.10 Format & Mount the partitions – Btrfs on LUKS + ESP.....	15
3 Install essential packages.....	18
3.1 Basic.....	18
3.2 Disk management.....	18
3.3 Firmware.....	18
3.4 Other crucial packages.....	18
3.5 OPT: Dynamic Kernel Module Support (DKMS).....	18
4 System configuration.....	19
4.1 Generate an fstab file.....	19
4.2 Chroot.....	19
4.3 Time.....	19
4.4 Locale.....	19
4.5 Network.....	20
4.6 Set root password.....	20
4.7 Allow ssh root login (for next boots).....	20
4.8 Swap (also for hibernation).....	21
4.9 Mkinitcpio (for UKI).....	22
4.10 Kernel command line (for mkinitcpio).....	23
4.11 Create UEFI boot entries using efibootmgr.....	25
4.12 Check if everything is working.....	25
4.13 Enter crypt-password & Login as <i>root</i>	25

4.14 For next steps.....	25
5 Secure Boot.....	26
5.1 Install SB tools.....	26
5.2 Backing up current SB variables.....	26
5.3 Creating keys in /root/efi-keys/.....	26
5.4 Signing the UKIs (using mkinitcpio post hook).....	28
5.5 Putting firmware in "Setup Mode".....	29
5.6 Enrolling keys using sbkeysync.....	29
5.7 Completing Secure Boot.....	30
5.8 Verify Secure Boot status.....	30
5.9 FYI: Disable SB by removing the PK using noPK.auth.....	30
6 System configuration (cont.).....	31
6.1 Locale (cont.): Set locale variables.....	31
6.2 Normal user.....	32
6.3 Pacman configuration.....	33
6.4 Auto update mirrors – reflector.....	33
6.5 Discard unused packages weekly – paccache.....	34
6.6 Limit journal size – systemd/Journal.....	34
6.7 Define missing environment variables.....	34
6.8 Improving compile times – Makepkg.....	34
6.9 Enable Periodic TRIM (if TRIM is supported).....	35
7 Install Backend & DE.....	36
7.1 Graphics driver.....	36
7.2 Fonts.....	39
7.3 Multimedia frameworks.....	39
7.4 WM/DE – KDE Plasma + KDE Gear.....	41
7.5 Misc.....	42
7.6 After installation.....	42
8 Display Manager – SDDM (for KDE).....	43
8.1 Configuration.....	43
8.2 Enable SDDM & Reboot into SDDM/KDE.....	43
8.3 Congratulations on reaching this point!.....	43
8.4 Change SDDM theme to "Breeze".....	43
9 Btrfs snapshots.....	44
9.1 Setup snapshots using snapper & snap-pac.....	44
9.2 Restoring / (subvolume @) to its previous snapshot.....	46
10 Security – Hardening Arch Linux.....	48
10.1 Open interactive shell with root prompt for this chapter.....	48
10.2 Restrict programs' capabilities – AppArmor.....	48
10.3 Sandboxing applications – Firejail.....	50
10.4 More Kernel Hardening.....	52
10.5 Firewall – FirewallD.....	53
10.6 Restricting root login.....	53
10.7 Mount hardening of the ESP.....	53
10.8 fwupd (Firmware updater & Verify platform security).....	54
10.9 Harden yourself.....	54
11 User-specific configurations.....	55
11.1 kdesu: Use "doas" instead of "su" (for KDE).....	55
11.2 AUR helper & Pacman wrapper – paru.....	55
11.3 Reduce output latency.....	56

11.4 PipeWire optimizations.....	57
12 Hardware specific.....	60
12.1 Sensors.....	60
12.2 OPT: Fan speed control.....	60
12.3 OPT: Stress testing.....	60
12.4 Printer & Scanner.....	61
12.5 Fingerprint reader.....	61
12.6 RGB Control.....	62
12.7 Peripherals.....	62
12.8 Desktop-PC (DDR5) – Hardware Recommendations.....	63
13 Software specific.....	64
13.1 Copy config files (Here: using sftp).....	64
13.2 Zsh.....	64
13.3 Install your packages from text files (using zsh).....	66
13.4 Firefox.....	66
13.5 LibreOffice.....	70
13.6 Gamemode (for low latency).....	71
13.7 Monitoring system performance – MangoHud.....	72
13.8 Gaming.....	74
13.9 KDE Configuration.....	76
13.10 Virtualization.....	78
13.11 Podman.....	85
13.12 mpv (media player).....	86
13.13 Hardware Security Key – Nitrokey 3.....	87
13.14 Windows 11 – Minimal Setup.....	93
13.15 Waydroid.....	94
13.16 Rust (using rustup, for devs).....	95
13.17 OPT: Easy Effects (for EQ & more).....	96
14 Disks & Data.....	97
14.1 Preparing the disk (as <i>root</i>).....	97
14.2 Setup encrypted internal disk (as <i>root</i>).....	97
14.3 TODO – OPT: Directory structure of your private data.....	100
14.4 TODO: Create a backup plan.....	101
15 Maintenance.....	102
15.1 Removing unused packages.....	102
15.2 Pacnew and Pacsave.....	102
15.3 Free up disk space.....	103
15.4 Btrfs.....	104
16 Troubleshooting.....	105
16.1 Downgrading packages.....	105
16.2 Check for errors (also for maintenance).....	105
16.3 Useful keyboard shortcuts.....	106
16.4 Common problems.....	107
17 TODOs.....	108
17.1 Additions.....	108
17.2 Deprecations.....	108
17.3 Current bugs & misbehaviors.....	108
17.4 Changelog.....	109

0 INTRO

This **extensive & modular guide** focuses on how to install and set up Arch Linux with the following goals in mind. The system should:

- be as **stable** as possible,
- be sufficiently **secure** (without compromising usability too much),
- have acceptable **low latency** (for real-time applications, including games),
- be **automated** as much as possible (also for a small maintenance of the system),
- be **well set up for** KDE, Gaming, Firefox, mpv, Virt-Manager, your hardware, ...

This guide should be very helpful especially – but not only – for Arch beginners. Many sections (esp. after installation) should be adaptable to other distributions.

0.1 Recommended prerequisites

- Having an [UEFI](#) system & a whole drive
- **View:** [FAQ](#) & [Help:Reading](#)
- For a better understanding of certain steps, **view:**
 - the [file system hierarchy overview](#) &
 - the [essential core utilities](#) of a GNU/Linux system

0.2 Key features – Realization of goals

0.2.1 [Ideal setup](#) of [Secure Boot](#) using

- [Unified kernel image \(UKI\)](#) – a single executable which will be booted directly from UEFI firmware (no extra boot manager). Using [mkinitcpio](#) as the UKI generator with [systemd init](#).
- [Full-disk encryption \(FDE\)](#) using [dm-crypts](#) LUKS2 extension
- [Own UEFI keys](#)
- [Kernel lockdown](#) (but disables hibernation)

0.2.2 File system: [Btrfs](#) ([Feature overview](#))

- **System rollback** if greater system upgrade issues occur
> **Creation of snapshots** during pacman transactions using: [Btrfs](#) > [Snapper](#) > [snap-pac](#)
- [Transparent and automatic zstd compression](#)
- [Hibernation support](#) using [swapfile](#)

0.2.3 Low latency

- [linux-zen](#), [Enabling real-time](#), Reducing [DRI](#) & [PipeWire](#) latency, ...

0.2.4 System Hardening

- Restrict programs' capabilities: [AppArmor](#)
- Sandboxing applications: [Firejail](#) (AA is required)
- Firewall: [Firewalld](#)
- [Kernel Hardening](#)

0.2.5 Setup of the following software

... to make them "best" in their category.

- Desktop environment: [KDE Plasma](#)
- AUR helper & Pacman wrapper: [Paru](#) | Shell: [Zsh](#)
- Web browser: [Firefox](#) | Media player: [mpv](#)

0.2.6 Disk layout

Partition	1: EFI system partition (ESP)	2: Root partition
File systems & Labels	└ FAT32 (Unencrypted)	└ "Arch" – LUKS2 encrypted └ "Root" (/dev/mapper/root) – Btrfs
Mount points	/efi	/, /.snapshots, /home, /swap, ...

0.3 Styles & Meaning

Style	Meaning
BRICK	<something you probably want to change>
GRAY	<OPT = optional, not always a good option>, <ALT = alternative>, <not necessary>
DARK RED 2	<ATTENTION>
RED	<BAD>, <You can really mess up>, <deletions>
GREEN	<OK>, <already good adjustment>, <additions>, <uncomment>
GOLD, BLUE, BOLD	For highlighting
 command ;	
> ... (one line)	Insert ...; Next step(s)
> ... (more lines)	Insert ...; Next step(s)

0.4 Notes

- [Get the current version](#) of this guide
- This guide is modular. For example, if the "Ideal Secure Boot Setup" is not desirable, skip certain steps and rename `/dev/mapper/root` to `/dev/nvme0n1p2` throughout this document.
- You should first perform the installation in a VM in e.g. [VirtualBox](#) or [QEMU/KVM](#).
- **Disclaimer:** This document is often adapted from the [official ArchWiki](#) => [General disclaimer](#)

1 Pre-boot

1.1 Acquire an installation image

<https://archlinux.org/download/> > See: HTTP Direct Downloads

1.1.1 Download: Image, Signature & Checksum file

- ISO: archlinux-*version*-x86_64.iso
- PGP signature: archlinux-*version*-x86_64.iso.sig
- Checksum file: b2sums.txt (ALT: sha256sums.txt)

1.1.2 Verify checksum

```
b2sum --ignore-missing -c b2sums.txt  
> <ISO>: OK
```

1.1.3 Verify signature (using sequoia-sq)

<https://gitlab.com/sequoia-pgp/sequoia>

ALT: https://wiki.archlinux.org/title/Installation_guide#Verify_signature

```
sq network wkd fetch pierre@archlinux.org -o release-key.pgp;  
sq verify --signer-file release-key.pgp --detached archlinux-version-  
x86_64.iso.sig archlinux-version-x86_64.iso;
```

1.2 Prepare the USB flash installation medium

Other tools: https://wiki.archlinux.org/title/USB_flash_installation_medium

Note: Plug the USB stick directly into the mainboard.

1.2.1 In GNU/Linux – Example (using dd)

Note: It is highly recommended to use tools like [isoimagewriter](#).

Note: Replacing /dev/*sdX* with your unmounted USB drive (See: [lsblk](#))!

```
dd bs=4M if=/path/to/archlinux-version-x86_64.iso of=/dev/sdX conv=fsync  
oflag=direct status=progress
```

1.2.2 In MS Windows – Example (using Rufus)

[Download](#) & Start Rufus:

- **Select:** ISO & USB drive
- **Target system:** UEFI
- **Partition scheme:** GPT

Note: If the USB drive does not boot properly using the default ISO Image mode, **DD Image mode** should be used instead.

1.3 Set up UEFI

1.3.1 Update UEFI firmware

Usual procedure:

- **Download** [latest stable](#) UEFI firmware & **Copy** file to a FAT32 formatted USB stick
- **Boot into the UEFI Boot manager** to start the update procedure
- **Update** the UEFI firmware & wait for completion
- **OPT:** Reset UEFI settings

1.3.2 UEFI settings

- **Boot into UEFI firmware setup utility** often by pressing [F2] after [POST](#)

1.3.2.1 *Basic*

- **Disable Secure Boot.** Later we will set that up
- **Disable CSM** (Compatibility Support Module) for "UEFI only"
- **OPT: Disable Fast Boot** for full initialization

1.3.2.2 *Save settings & Reboot*

- **Save settings & Reboot** (often [F10])

1.4 Notes for MS Windows dual booters

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

- **First install Windows** on a separate drive, otherwise Windows will use Archs ESP.
ALT: Remove the Linux drive during the Windows installation.
- **Disable Fast Startup & Hibernation**
- [Use UTC](#) instead of localtime
- **Disable your Linux disks** in the "Disk Management utility"! **Do not create a (new) GPT!**

1.5 For Oracle VM VirtualBox: Port Forwarding

Network settings > Advanced: Port Forwarding:

- **Host port:** 3022
- **Guest port:** 22

2 Pre-installation

2.1 Boot from USB flash installation medium

Note: Press vendor specific key after [POST](#) to open the UEFI boot menu (often [F8] or [F12]).

ALT: Set boot order in UEFI firmware setup utility.

2.2 Set the keyboard layout

Note: Default locale is **Germany (de_DE)**.

Available layouts:

```
| ls /usr/share/kbd/keymaps/**/*.*.map.gz
```

Note – en ← de: | **y** ← **z** | **_** ← **ß** | **/** ← **-** | **** ← **#** | **:** ← **ö** | **^** ← **&** |

```
| loadkeys de_latin1
```

2.3 Verify the EFI-boot mode

Check UEFI firmware bitness:

```
| cat /sys/firmware/efi/fw_platform_size
```

> **Output:** **64** or **32**

2.4 Connect to the internet

```
| ping -c 4 archlinux.org
```

> **Error?:** https://wiki.archlinux.org/title/Installation_guide#Connect_to_the_internet

2.5 Set temporary root password (for ssh root login)

```
| passwd
```

> **myTmpRootPassword**

2.6 Connect via ssh (as root)

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

... to set up Arch Linux via SSH to copy & paste (esp. longer) text without failures.

This guide is very copy & paste friendly!

Note: Repeat these steps later to login as **root** or as **myUsername** ("as user").

2.6.1 Later: Start ssh daemon

```
| systemctl start sshd.service;
```

2.6.2 For Oracle VM VirtualBox: Connect

```
| ssh -p 3022 root@localhost
```

2.6.3 Connect to (new) IP address

Note: In LAN often 192.168.X.Y, else see [here](#).

Note: You may want to delete the fingerprint(s) from last session in ~/.ssh/known_hosts.

```
| ssh root@<ip> -br addr>
```

2.7 Update the system clock

https://wiki.archlinux.org/title/System_time#Time_zone

Available zones: `timedatectl list-timezones`

```
| timedatectl set-timezone Europe/Berlin;  
| timedatectl set-ntp true
```

2.8 Preparing the disk

Note: The drive should be connected directly to a SATA/NVMe/... interface. Issuing the Secure Erase/Format/Sanitize command on a drive via USB or a SAS/RAID card could brick the drive!

2.8.1 Identify your disk/drive

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

Note: [NVMe SSDs](#) are named: nvme**D**n**N**p**P** (**D** ∈ {0, 1, 2, ...}, **N** ∈ {1, 2, ...}).
Other drives are often named: sd**D****P** (**D** ∈ {a, b, ...}).
... **D**evice, **N**amespace, **P**artition ∈ {1, 2, ...}

```
| lsblk -f  
> /dev/nvme0n1 is our disk < CHANGE this accordingly in this whole document
```

Example: Your disk is **vda** (or **sda**, **sdX**, ...)

- Replace **nvme0n1p** with **vda** **AND THEN**
- Replace **nvme0n1** with **vda**

2.8.2 OPT: Update firmware of the drive

SSD: https://wiki.archlinux.org/title/Solid_state_drive#Firmware

NVMe SSD: https://wiki.archlinux.org/title/Solid_state_drive/NVMe#Firmware_update

2.8.3 Check health of the drive – S.M.A.R.T.

2.8.3.1 For NVMe

https://wiki.archlinux.org/title/Solid_state_drive/NVMe#SMART

```
| nvme smart-log -H /dev/nvme0n1
```

2.8.3.2 For non-NVMe (using smartmontools)

<https://wiki.archlinux.org/title/S.M.A.R.T.#smartctl>

2.8.3.2.1 Check if SMART support is "Available" & "Enabled"

```
| smartctl --info /dev/sdb | grep 'SMART support is:'  
> Available but not enabled?:  
| smartctl --smart=on --device=ata /dev/sdb
```

2.8.3.2.2 Run a self-test

Available self-tests: `smartctl -c /dev/sdb`

```
| smartctl -t short /dev/sdb
```

OPT – Check transportation damage of the HDD:

```
| smartctl -t conveyance /dev/sdb
```

2.8.3.2.3 View test results

```
| smartctl -l selftest /dev/sdb
```

2.8.4 (Secure) Erasure of the drive

https://wiki.archlinux.org/title/Dm-crypt/Drive_preparation

2.8.4.1 Wipe old LUKS header (if LUKS encrypted before)

https://wiki.archlinux.org/title/Dm-crypt/Drive_preparation#Wipe_LUKS_header

```
| cryptsetup erase /dev/nvme0n1p2
```

2.8.4.2 Erase all old available signatures

```
| wipefs --all /dev/nvme0n1
```

2.8.4.3 Setting the sector size to 4 KiB

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

Drives often report a **logical sector size of 512 B** (for stone age compatibility), but use a **larger physical sector size**. Many filesystems (including btrfs) default to a sector size of 4 KiB.

=> To avoid mapping, **change the sector size to 4 KiB**, which should also be closer to the physical sector size and thus **improve performance**.

2.8.4.3.1 Check if there is a better sector size available (Here: NVMe)

```
| nvme id-ns -H /dev/nvme0n1 | grep "Relative Performance"
```

> E.g. 2 supported sector sizes:

```
lbaf 0 : ms:0 lbads:9 rp:0x2 (in use)
lbaf 1 : ms:0 lbads:12 rp:0x1
```

1. ms (Metadata Size) should be 0 (ms is not well supported under Linux)
2. rp (Relative Performance) should be the smallest (Here: rp:0x1)
=> **Here:** Change the LBA format with its sector size for better performance

Note: "Data Size" = 2^{lbads} (LBA data size) [bytes] (Here: $2^9 = 512$ B; $2^{12} = 4096$ B)

=> **Here:** Change the sector/data size from 512 B to 4 KiB

2.8.4.3.2 Change the LBA format with its sector size (Here: NVMe)

```
| nvme format --lbaf=<lbaf> /dev/nvme0n1 (Here: 1)
```

2.8.4.4 Memory cell clearing (for SSD)

https://wiki.archlinux.org/title/Solid_state_drive/Memory_cell_clearing

... thus restoring it to its factory default write performance.

Note: Do this only if you have not changed the sector size.

```
| E.g.: nvme format /dev/nvme0n1
```

2.8.4.5 Prevent cryptographic attacks or file recovery (for dm-crypt)

https://wiki.archlinux.org/title/Dm-crypt/Drive_preparation#dm-crypt_wipe_on_an_empty_disk_or_partition

Wipe the disk with crypto-grade randomness **if** the disk contains non-random or unencrypted data.

2.8.4.5.1 Create a temporary encrypted container

https://wiki.archlinux.org/title/Dm-crypt/Device_encryption#Encryption_options_for_plain_mode

Note: Change **cipher** according to \$ **cryptsetup benchmark**

```
| cryptsetup -d /dev/urandom --cipher aes-xts-plain64 open --type plain  
/dev/nvme0n1 to_be_wiped
```

2.8.4.5.2 Wipe with (encrypted) zeros

```
| dd bs=1M if=/dev/zero of=/dev/mapper/to_be_wiped status=progress  
> WAIT:| 16,67 min/TB @1 GB/s | 1,5 h/TB @185 MB/s | 2 h/TB @139 MB/s |
```

2.8.4.5.3 Close the temporary container

```
| cryptsetup close to_be_wiped
```

2.9 Partition the disk using gdisk – ESP + Root

Modified: [LUKS on a partition with TPM2 and Secure Boot](#)

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

https://wiki.archlinux.org/title/EFI_system_partition#Create_the_partition

Num	Size	gdisk's code	Partition type
1	1 GiB	EF00	EFI system
2	100%FREE	8304	Linux x86-64 root (/)

```
| gdisk /dev/nvme0n1  
>
```

2.9.1 Create the GUID Partition Table (GPT)

https://wiki.archlinux.org/title/Partitioning#GUID_Partition_Table

```
o
```

2.9.2 Create the EFI System Partition (ESP)

```
n  
enter (default: 1)  
enter (default: ...)  
+1g  
ef00 (EFI System)
```

2.9.3 Create the Root Partition

```
n  
enter (default: 2)  
enter (default: ...)  
enter (default: [Free Space])  
8304 (Linux x86-64 root (/); also for LUKS)
```

2.9.4 Commit changes

```
w
```

2.10 Format & Mount the partitions – Btrfs on LUKS + ESP

<https://wiki.archlinux.org/title/Partitioning>

https://wiki.archlinux.org/title/Security#Mount_options

https://en.wikipedia.org/wiki/Comparison_of_file_systems

2.10.1 Preparing the LUKS container

2.10.1.1 Create the LUKS encrypted container

https://wiki.archlinux.org/title/Dm-crypt/Device_encryption#Encryption_options_for_LUKS_mode

```
| cryptsetup luksFormat --label Arch /dev/nvme0n1p2  
> YES  
> crypt-password
```

2.10.1.2 OPT: Enable TRIM/discard support (for encrypted SSD)

READ: [Discard/TRIM support for solid state drives \(SSD\)](#)

Warning: Negative security impact because it can make filesystem-level operations visible on the physical device (information leaking filesystem type, used space, etc.). If in doubt, do not enable it.

2.10.1.3 Verify the sector size used by the LUKS2 volume

https://wiki.archlinux.org/title/Advanced_Format#dm-crypt

... esp. if you have changed the sector size earlier.

```
| cryptsetup luksDump /dev/nvme0n1p2 | grep sector
```

2.10.1.4 Open the LUKS container

```
| cryptsetup open /dev/nvme0n1p2 root  
> crypt-password
```

2.10.2 Preparing btrfs (also for snapper & swap)

<https://wiki.archlinux.org/title/Btrfs#Subvolumes>

https://wiki.archlinux.org/title/Snapper#Suggested_filesystem_layout

2.10.2.1 Format the (unlocked LUKS) device

```
| mkfs.btrfs -L Root /dev/mapper/root;
```

2.10.2.2 Mount the top-level btrfs subvolume

```
| mount /dev/mapper/root /mnt;
```

2.10.2.3 Create nested btrfs subvolumes (for excludes of snapshots)

... using a *flat* layout (all subvolumes are direct descendants of the toplevel one).

Note: "Snapshotting is not recursive, so a subvolume or a snapshot is effectively a barrier and no files in the nested appear in the snapshot."

```
| btrfs su cr /mnt/@;                # /
| btrfs su cr /mnt/@home;            # /home
| btrfs su cr /mnt/@snapshots;       # /.snapshots
| btrfs su cr /mnt/@srv;             # /srv
| btrfs su cr /mnt/@swap;            # /swap
| btrfs su cr /mnt/@var_cache_pacmanPkg; # /var/cache/pacman/pkg
| btrfs su cr /mnt/@var_lib_containers; # /var/lib/containers
| btrfs su cr /mnt/@var_lib_libvirtImages; # /var/lib/libvirt/images
| btrfs su cr /mnt/@var_log;         # /var/log
| btrfs su cr /mnt/@var_tmp;         # /var/tmp
```

2.10.2.4 Unmount the top-level btrfs subvolume

```
| umount /mnt;
```

2.10.2.5 Mount the nested subvolumes

```
| mount -o compress=zstd,subvol=@ /dev/mapper/root /mnt;
```

```
| mount -m -o compress=zstd,subvol=@home /dev/mapper/root /mnt/home;
| mount -m -o compress=zstd,subvol=@snapshots /dev/mapper/root /mnt/.snapshots;
| mount -m -o compress=zstd,subvol=@srv /dev/mapper/root /mnt/srv;
| mount -m -o compress=zstd,subvol=@swap /dev/mapper/root /mnt/swap;
| mount -m -o compress=zstd,subvol=@var_cache_pacmanPkg /dev/mapper/root
| /mnt/var/cache/pacman/pkg;
| mount -m -o compress=zstd,subvol=@var_lib_containers /dev/mapper/root
| /mnt/var/lib/containers;
| mount -m -o compress=zstd,subvol=@var_lib_libvirtImages /dev/mapper/root
| /mnt/var/lib/libvirt/images;
| mount -m -o compress=zstd,subvol=@var_log /dev/mapper/root /mnt/var/log;
| mount -m -o compress=zstd,subvol=@var_tmp /dev/mapper/root /mnt/var/tmp;
```


2.10.3 Preparing the EFI system partition (ESP)

https://wiki.archlinux.org/title/EFI_system_partition#Typical_mount_points

Note: The *esp* will be mounted on */efi* (for UKI).

```
mkfs.fat -F32 /dev/nvme0n1p1;  
mount -m -o umask=0077,noexec,nosuid,nodev /dev/nvme0n1p1 /mnt/efi;
```

3 Install essential packages

https://wiki.archlinux.org/title/Installation_guide#Install_essential_packages

| `pacstrap -K /mnt <packages in this chapter>`

3.1 Basic

<https://wiki.archlinux.org/title/Kernel>, [Benchmarks \(2023-01\)](#)

- Base: [base](#) [base-devel](#) [linux-firmware](#)
- UKI generator: [mkinitcpio](#) [systemd-ukify](#)
- Recommended Kernel: [linux-zen](#) ([Feature List](#))
- ALT: [linux](#) [linux-hardened](#) [linux-rt](#) [linux-rt-lts](#)
- OPT – Fallback: [linux-lts](#) (Major kernel upgrades *can* cause problems. ALT: Rollback system)

3.2 Disk management

- NVMe: [nvme-cli](#)
- Dm-crypt: [cryptsetup](#)
- Btrfs: [btrfs-progs](#)
- FAT: [dosfstools](#)
- Ext4: [e2fsprogs](#) | exFAT: [exfatprogs](#) | NTFS: [ntfs-3g](#)

3.3 Firmware

- Processor Microcode (not for VM): [amd-ucode](#) or [intel-ucode](#)
- For some audio devices (≥2020): [sof-firmware](#)
- For some audio devices: [alsa-firmware](#)

3.4 Other crucial packages

- Documentation: [man-db](#) [man-pages](#) [man-pages-de](#)
- Network manager: [networkmanager](#) ([Wiki](#)) (dep. of [plasma-meta](#))
- Pacman tools: [pacman-contrib](#) [rebuild-detector](#) [pkgstats](#)
- Console text editor: [vim](#) ([Wiki](#))

3.5 OPT: Dynamic Kernel Module Support (DKMS)

[dkms](#) + Headers according to the kernels to be installed:

[linux-zen-headers](#) [linux-lts-headers](#) [linux-headers](#) [linux-hardened-headers](#)

4 System configuration

4.1 Generate an [fstab](#) file

```
| genfstab -U /mnt >> /mnt/etc/fstab;
```

4.2 Chroot

https://wiki.archlinux.org/title/Chroot#Using_arch-chroot

```
| arch-chroot /mnt
```

4.3 Time

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

4.3.1 Set time zone

Available [time zones](#) (on host): `timedatectl list-timezones | grep -i search_term;`

```
| ln -sf /usr/share/zoneinfo/Europe/Berlin /etc/localtime;
```

4.3.2 Set hardware clock from system clock

```
| hwclock --systohc;
```

4.3.3 Time synchronization (NTP) – systemd-timesyncd

<https://wiki.archlinux.org/title/Systemd-timesyncd>

```
| systemctl enable systemd-timesyncd.service;
```

4.4 Locale

<https://wiki.archlinux.org/title/Locale>

4.4.1 Generating locales

Note – vim: Press [i] to go into **insert mode** to type; and [ESC] to go out into **normal mode**.

Note – vim (normal mode): | :x = :wq = Save & Close | / = Search (n=Next; N=Previous) |

```
| vim /etc/locale.gen
```

> **Uncomment** (using [De] over # in normal mode):

de_DE.UTF-8 UTF-8	# Region based
en_US.UTF-8 UTF-8	# Also for user interface
ja_JP.UTF-8 UTF-8	# Japanese locale support
ko_KR.UTF-8 UTF-8	# Korean locale support

```
| locale-gen
```

4.4.2 Set persistent keymap layout (for crypt-password in 1st boot)

https://wiki.archlinux.org/title/Linux_console/Keyboard_configuration

... because `localectl set-keymap` only works after a reboot.

Available keymaps (on host): `localectl list-keymaps | grep -i search_term`;

```
| echo KEYMAP=de-latin1-nodeadkeys > /etc/vconsole.conf;
```

4.5 Network

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

4.5.1 Set the hostname

https://wiki.archlinux.org/title/Network_configuration#Set_the_hostname

```
| echo myHostname > /etc/hostname;
```

4.5.2 Local hostname resolution (for some software)

https://wiki.archlinux.org/title/Network_configuration#localhost_is_resolved_over_the_network

```
| vim /etc/hosts
>
127.0.0.1 localhost
::1      localhost
127.0.1.1 myHostname
```

4.5.3 Set the regulatory domain (for Wi-Fi)

https://wiki.archlinux.org/title/Network_configuration/Wireless#Respecting_the_regulatory_domain

```
| pacman -S wireless-regdb;
```

```
| vim /etc/conf.d/wireless-regdom
```

> Uncomment:

```
WIRELESS_REGDOM="DE"
```

4.5.4 Enable NetworkManager

```
| systemctl enable NetworkManager.service;
```

4.6 Set root password

```
| passwd;
> myRootPassword
```

4.7 Allow ssh root login (for next boots)

```
| pacman -S openssh;
| vim /etc/ssh/sshd_config
> PermitRootLogin yes (Later: Change to "no")
```

4.8 Swap (also for hibernation)

<https://wiki.archlinux.org/title/Swap>

https://wiki.archlinux.org/title/Btrfs#Swap_file

<https://btrfs.readthedocs.io/en/latest/Swapfile.html>

https://man.archlinux.org/man/btrfs.5.en#SWAPFILE_SUPPORT

... to extend the virtual memory beyond the installed physical memory (RAM) or to be able to hibernate.

4.8.1 Create a swapfile (for btrfs) (Here: 32 GiB)

Note – For hibernation: If the swap size is smaller than the RAM size and the RAM is not fully utilized, you still have a chance of hibernating successfully.

```
| btrfs filesystem mkswapfile --size 32g --uuid clear /swap/swapfile;
```

4.8.2 Activate the swapfile

```
| swapon /swap/swapfile;
```

4.8.3 Add an entry to /etc/fstab

```
| vim /etc/fstab
```

> **Append:**

```
# /swap/swapfile
```

```
/swap/swapfile      none          swap          defaults 0 0
```

4.8.4 Lower swappiness value (for low latency)

... to avoid swapping for better system responsiveness.

```
| echo "vm.swappiness = 10" > /etc/sysctl.d/99-swappiness.conf;
```

4.9 Mkinitcpio (for UKI)

4.9.1 HOOKS (for systemd init & fsck, dm-crypt & hibernation)

<https://wiki.archlinux.org/title/Mkinitcpio#HOOKS>

<https://wiki.archlinux.org/title/Fsck>

Note – If not using dm-crypt: Don't add sd-encrypt or shift keyboard.

```
| vim /etc/mkinitcpio.conf
> HOOKS=(base udev systemd keyboard autodetect microcode modconf kms keyboard
keymap-consolefont block sd-vconsole sd-encrypt filesystems-fsck)
```

4.9.2 Enable Unified kernel images (UKI)

https://wiki.archlinux.org/title/Unified_kernel_image#mkinitcpio

https://wiki.archlinux.org/title/Arch_boot_process#Boot_loader

Note: Modify other presets accordingly to installed kernels (*linux*, *linux-lts*, ...).

Primary place for UKIs:

```
| mkdir -p /efi/EFI/Linux;
```

```
| vim /etc/mkinitcpio.d/linux-zen.preset
```

> **Change:**

```
#default_image="/boot/initramfs-linux-zen.img"
default_uki="/efi/EFI/Linux/archlinux-zen.efi"
default_options="--splash=/usr/share/systemd/bootctl/splash-arch.bmp"

#fallback_image="/boot/initramfs-linux-zen-fallback.img"
fallback_uki="/efi/EFI/Linux/archlinux-zen-fallback.efi"
fallback_options="-S autodetect"
```

4.9.2.1 Generate all UKIs

```
| mkinitcpio -P
```

4.9.2.2 TMP: Remove all "default_images"

```
| rm /boot/initramfs-linux*.img;
```

4.9.3 Install possibly missing firmware

https://wiki.archlinux.org/title/Mkinitcpio#Possibly_missing_firmware_for_module_XXXX

When generating the **default** UKI using mkinitcpio, you may get the warning:

```
==> WARNING: Possibly missing firmware for module: 'module_name'
```

> **Install** [the corresponding packages for your modules](#)

4.10 Kernel command line (for mkinitcpio)

https://wiki.archlinux.org/title/Unified_kernel_image#mkinitcpio

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

<https://docs.kernel.org/admin-guide/kernel-parameters.html>

... with your kernel parameters.

Note: *<Output of <command> in vim (normal mode) using `:r! <command>`>*

Create the drop-in config directory:

```
| mkdir /etc/cmdline.d;
```

4.10.1 root.conf (for btrfs, hibernation)

https://uapi-group.org/specifications/specs/discoverable_partitions_specification/

Note: For btrfs, the default subvolume is set with rootflags. “btrfs subvolume set-default” is not used because after each system rollback you would have to re-set the new “@” subvolume.

```
| vim /etc/cmdline.d/root.conf
>
# Btrfs: GPT partition automount
rootflags=subvol=@

# Hide OEM logo & Reduce boot messages
bgrt_disable quiet loglevel=4
```

4.10.1.1 Regenerate all UKIs

```
| mkinitcpio -P
```

4.10.2 performance.conf

```
| vim /etc/cmdline.d/performance.conf  
>
```

4.10.2.1 For low latency

<https://gitlab.freedesktop.org/pipewire/pipewire/-/wikis/Performance-tuning#kernel>

https://wiki.archlinux.org/title/Professional_audio#System_configuration

```
# Low latency  
preempt=full threadirqs
```

4.10.2.2 Disable staggered spin-up

https://wiki.archlinux.org/title/Improving_performance/Boot_process#Staggered_spin-up

... otherwise, the ATA interfaces are probed serially > Thus slower boot speed.

Check if SSS is being used > Yes (output)?:

```
| dmesg | grep SSS
```

```
# Disable staggered spin-up  
libahci.ignore_sss=1
```

4.10.2.3 For AMD Ryzen CPUs (\geq Zen 2): CPU power scaling driver

https://wiki.archlinux.org/title/CPU_frequency_scaling#Scaling_drivers

<https://docs.kernel.org/admin-guide/pm/amd-pstate.html>

Benchmarks: [Mobile](#), [EPYC](#) (2023-04)

- In UEFI: Enable "CPPC" & "CPPC Preferred Cores" for amd_pstate_epp CPU power scaling driver. Usually found in: AMD CBS > NBIO > SMU > CPPC
- Add: `amd_pstate=active`
- Verify after reboot: `doas cpupower frequency-info`

4.11 Create UEFI boot entries using efibootmgr

https://wiki.archlinux.org/title/Unified_Extensible_Firmware_Interface#efibootmgr

ALT – Boot manager: https://wiki.archlinux.org/title/Unified_kernel_image#systemd-boot

... to boot directly from UEFI firmware.

```
| pacman -S efibootmgr;
```

Do the following for all installed kernels: (*linux*, *linux-lts*, ...)

```
| efibootmgr --create --disk /dev/nvme0n1 --part 1 --label "ArchLinux-zen-  
fallback" --loader EFI/Linux/archlinux-zen-fallback.efi --unicode;
```

```
| efibootmgr --create --disk /dev/nvme0n1 --part 1 --label "ArchLinux-zen" --  
loader EFI/Linux/archlinux-zen.efi --unicode;
```

4.12 Check if everything is working

4.12.1 Reboot

```
| exit (OR [Ctrl]+[D])
```

```
| OPT: umount -R /mnt
```

> Busy partitions?: Ignore (swap), double-check configs AND/OR use [fuser\(1\)](#).

```
| systemctl reboot
```

> UEFI boot entry "*ArchLinux-zen*" should be visible.

Otherwise, it *should* be possible to add an entry manually using the UEFI firmware setup utility.

ALT: Use [systemd-boot](#).

4.12.2 Boot problems?

Open LUKS > Mount root partition > chroot > Mount other partitions > Fix typo/problem:

```
| loadkeys de-latin1;  
| cryptsetup open /dev/nvme0n1p2 root;  
| mount -o compress=zstd,subvol=@ /dev/mapper/root /mnt;  
| arch-chroot /mnt
```

> Mount all partitions listed in /etc/fstab:

```
| mount -a
```

> Fix typo/problem

4.13 Enter crypt-password & Login as root

Type root & Enter *myRootPassword*. Don't forget to [Connect via ssh \(as root\)](#).

4.14 For next steps

```
| mkdir /etc/pacman.d/hooks;  
| pacman -S --needed wget;
```

5 Secure Boot

https://wiki.archlinux.org/title/Unified_Extensible_Firmware_Interface/Secure_Boot#Using_your_own_keys

5.1 Install SB tools

```
| pacman -S efitools sbsigntools;
```

5.2 Backing up current SB variables

```
| mkdir -p efi-keys/0_old; cd efi-keys/0_old/;  
| for var in PK KEK db dbx ; do efi-readvar -v $var -o old_${var}.esl ; done  
| cd ...;
```

5.3 Creating keys in /root/efi-keys/

https://wiki.archlinux.org/title/Unified_Extensible_Firmware_Interface/Secure_Boot#Manual_process

5.3.1 Create a GUID for owner identification

```
| uuidgen --random > GUID.txt;
```

5.3.2 Platform Key (PK)

```
| openssl req -newkey rsa:4096 -nodes -keyout PK.key -new -x509 -sha256 -days  
| 3650 -subj "/CN=my Platform Key/" -out PK.crt;  
| openssl x509 -outform DER -in PK.crt -out PK.cer;  
| cert-to-efi-sig-list -g "$(< GUID.txt)" PK.crt PK.esl;  
| sign-efi-sig-list -g "$(< GUID.txt)" -k PK.key -c PK.crt PK PK.esl PK.auth;
```

5.3.3 Sign an empty file to allow removing PK when in "User Mode"

```
| sign-efi-sig-list -g "$(< GUID.txt)" -c PK.crt -k PK.key PK /dev/null  
| noPK.auth;
```

5.3.4 Key Exchange Key (KEK)

```
| openssl req -newkey rsa:4096 -nodes -keyout KEK.key -new -x509 -sha256 -days  
| 3650 -subj "/CN=my Key Exchange Key/" -out KEK.crt;  
| openssl x509 -outform DER -in KEK.crt -out KEK.cer;  
| cert-to-efi-sig-list -g "$(< GUID.txt)" KEK.crt KEK.esl;  
| sign-efi-sig-list -g "$(< GUID.txt)" -k PK.key -c PK.crt KEK KEK.esl KEK.auth;
```

5.3.5 Signature Database (db)

```
| openssl req -newkey rsa:4096 -nodes -keyout db.key -new -x509 -sha256 -days  
| 3650 -subj "/CN=my Signature Database key/" -out db.crt;  
| openssl x509 -outform DER -in db.crt -out db.cer;  
| cert-to-efi-sig-list -g "$(< GUID.txt)" db.crt db.esl;  
| sign-efi-sig-list -g "$(< GUID.txt)" -k KEK.key -c KEK.crt db db.esl db.auth;
```

5.3.6 Microsoft's certificates

https://wiki.archlinux.org/title/Unified_Extensible_Firmware_Interface/Secure_Boot#Microsoft_Windows

<https://learn.microsoft.com/en-us/windows-hardware/manufacture/desktop/windows-secure-boot-key-creation-and-management-guidance>

Note: Your device *may* only functions properly w/ Microsoft's UEFI CA certificates. You may find a checkbox in your UEFI SB settings to include these MS certs. This would be an indication that you **must** add the following UEFI CA certs.

5.3.6.1 Add Microsoft's UEFI CA certificates (for Linux)

... in order to use third-party binaries like UEFI drivers, option ROMs, etc.

5.3.6.1.1 Download certificates

```
wget --user-agent="Mozilla" -O MS_UEFI-2011.crt
https://www.microsoft.com/pkiops/certs/MicCorUEFCA2011_2011-06-27.crt;
wget --user-agent="Mozilla" -O MS_UEFI-2023.crt
https://www.microsoft.com/pkiops/certs/microsoft%20uefi%20ca%202023.crt;
```

5.3.6.1.2 Create a EFI Signature List

```
sbsiglist --owner 77fa9abd-0359-4d32-bd60-28f4e78f784b --type x509 --output
MS_UEFI-2011_db.esl MS_UEFI-2011.crt;
sbsiglist --owner 77fa9abd-0359-4d32-bd60-28f4e78f784b --type x509 --output
MS_UEFI-2023_db.esl MS_UEFI-2023.crt;
cat MS_UEFI-2011_db.esl MS_UEFI-2023_db.esl > MS_UEFI_db.esl;
```

5.3.6.1.3 Sign the db.esl with your KEK

```
sign-efi-sig-list -a -g 77fa9abd-0359-4d32-bd60-28f4e78f784b -k KEK.key -c
KEK.crt db MS_UEFI_db.esl add_MS_UEFI_db.auth;
```

5.3.6.2 Add Microsoft Windows's certificates (for Windows)

5.3.6.2.1 Download certificates

```
wget --user-agent="Mozilla" -O Win_Production_PCA-2011.crt
https://www.microsoft.com/pkiops/certs/MicWinProPCA2011_2011-10-19.crt;
wget --user-agent="Mozilla" -O Win_UEFI-2023.crt
https://www.microsoft.com/pkiops/certs/windows%20uefi%20ca%202023.crt;
```

5.3.6.2.2 Create a EFI Signature List

```
sbsiglist --owner 77fa9abd-0359-4d32-bd60-28f4e78f784b --type x509 --output
Win_Production_PCA-2011_db.esl Win_Production_PCA-2011.crt;
sbsiglist --owner 77fa9abd-0359-4d32-bd60-28f4e78f784b --type x509 --output
Win_UEFI-2023_db.esl Win_UEFI-2023.crt;
cat Win_Production_PCA-2011_db.esl Win_UEFI-2023_db.esl > MS_Win_db.esl;
```

5.3.6.2.3 Sign the db.esl with your KEK

```
sign-efi-sig-list -a -g 77fa9abd-0359-4d32-bd60-28f4e78f784b -k KEK.key -c
KEK.crt db MS_Win_db.esl add_MS_Win_db.auth;
cd ..
```

5.4 Signing the UKIs (using mkinitcpio post hook)

https://wiki.archlinux.org/title/Unified_kernel_image#Signing_the_UKIs_for_Secure_Boot

```
| vim /etc/initcpio/post/uki-sbsign
>
#!/usr/bin/env bash

uki="$3"
[[ -n "$uki" ]] || exit 0

keypairs=(/root/efi-keys/db.key /root/efi-keys/db.crt)

for (( i=0; i<${#keypairs[@]}; i+=2 )); do
    key="${keypairs[$i]}" cert="${keypairs[(( i + 1 ))]}"
    if ! sbverify --cert "$cert" "$uki" &>/dev/null; then
        sb sign --key "$key" --cert "$cert" --output "$uki" "$uki"
    fi
done

chmod +x /etc/initcpio/post/uki-sbsign;
mkinitcpio -P
```

5.5 Putting firmware in "Setup Mode"

5.5.1 Reboot into UEFI firmware setup utility

| `systemctl reboot --firmware-setup` OR by hitting [F2] after POST

- **Secure Boot settings:**
Activate vendor specific settings like "Expert Key Management"

5.5.2 Backup & Clear preloaded Secure Boot certificates

- **OPT: Backup** all preloaded Secure Boot keys to a FAT32 formatted USB stick
- **Clear** all Secure Boot certificates: Clear "ALL" or db > KEK > PK
Note: Secure Boot is now in "Setup Mode"
- **Reboot**

5.6 Enrolling keys using sbkeysync

https://wiki.archlinux.org/title/Unified_Extensible_Firmware_Interface/Secure_Boot#Using_sbkeysync

Note: You may need to reset the UEFI administrator password.

5.6.1 Copy each .auth file into their respective location

| `mkdir -p /etc/secureboot/keys/{db,dbx,KEK,PK};`

| `cp /root/efi-keys/db.auth /etc/secureboot/keys/db/;`
| `cp /root/efi-keys/KEK.auth /etc/secureboot/keys/KEK/;`
| `cp /root/efi-keys/PK.auth /etc/secureboot/keys/PK/;`

Microsoft's certificates:

| `cp /root/efi-keys/add_MS_UEFI_db.auth /etc/secureboot/keys/db/;`
| `cp /root/efi-keys/add_MS_Win_db.auth /etc/secureboot/keys/db/;`

5.6.2 Verify the changes sbkeysync will make to the UEFI keystore

| `sbkeysync --pk --dry-run --verbose`

5.6.3 Enroll your keys

| `sbkeysync --verbose`
| `sbkeysync --verbose --pk`

> **Write errors?**

| `chattr -i /sys/firmware/efi/efivars/{PK,KEK,db}*`

> **"Permission denied" error for PK.auth?**

| `efi-updatevar -f /etc/secureboot/keys/PK/PK.auth PK`

> **Enroll again**

5.7 Completing Secure Boot

- Reboot into UEFI
- Enable Secure Boot (e.g. enable "Windows UEFI mode")
- Reboot
- After testing: Set UEFI administrator password to protect the firmware settings

5.8 Verify Secure Boot status

```
| bootctl status  
> System:  
  Firmware: UEFI 2.70 (American Megatrends 5.17)  
  Firmware Arch: x64  
  Secure Boot: enabled (user)  
  TPM2 Support: yes  
  Measured UKI: yes  
  Boot into FW: supported
```

5.9 FYI: Disable SB by removing the PK using noPK.auth

... if you encountering problems after enabling SB.

Note: This will change the SB User Mode to Setup Mode. After that:

- Clear all other keys
- Restore your backup keys
- Disable SB

```
| efi-updatevar -f noPK.auth PK
```

6 System configuration (cont.)

6.1 Locale (cont.): Set locale variables

<https://wiki.archlinux.org/title/Locale#Variables>

Available locales: `localectl list-locales`;

6.1.1 LANG: Default locale

```
| localectl set-locale LANG=de_DE.UTF-8;
```

6.1.2 OPT: Prefer "en_US" for the User interface

... for better troubleshooting & keyboard shortcuts.

Note: Else, you could change the [XDG user directories](#) to one named in English.

6.1.2.1 LC_MESSAGES: User interface for message translation

```
| localectl set-locale LC_MESSAGES=en_US.UTF-8;
```

6.1.2.2 LANGUAGE: Fallback locales (in order)

```
| localectl set-locale LANGUAGE=en_US:en:C:de_DE;
```

6.1.3 X11 keymap layout

https://wiki.archlinux.org/title/Xorg/Keyboard_configuration#Using_localectl

```
| localectl set-x11-keymap de pc105 nodeadkeys;
```

6.1.4 Verify

```
| localectl status;  
| locale;
```

6.2 Normal user

https://wiki.archlinux.org/title/Users_and_groups#User_management

6.2.1 Add a normal user

```
| useradd -m myUsername;      < CHANGE this in this whole document  
  
| passwd myUsername  
> myUserPassword
```

6.2.2 Add more permissions to user

https://wiki.archlinux.org/title/Users_and_groups#Group_list

```
| usermod -aG wheel,games,ftp,http,audio myUsername
```

OPT – Print permissions/groups of user:

```
| groups myUsername
```

6.2.3 Use "doas" instead of "sudo"

<https://wiki.archlinux.org/title/Doas>

... as a simpler and safer sudo replacement.

6.2.3.1 Installation & Allow members of group wheel "doas" access

```
| pacman -S opendoas;  
  
| vim /etc/doas.conf  
>  
permit :wheel  
permit persist :wheel
```

Note: The configuration file must end with a newline!

6.2.3.2 Change permissions

```
| chown -c root:root /etc/doas.conf;  
| chmod -c 0400 /etc/doas.conf;
```

6.2.3.3 Check for syntax errors

```
| doas -C /etc/doas.conf && echo "config ok" || echo "config error";
```

6.2.4 Allow members of group wheel "sudo" access

Note: Some programs require sudo for privilege elevation. Package is installed from base-devel.

```
| EDITOR=rvim visudo  
> Uncomment:  
%wheel ALL=(ALL:ALL) ALL
```


6.2.5 Allow only users of group wheel "su" access

https://wiki.archlinux.org/title/Su#su_and_wheel

```
| vim /etc/pam.d/su
| vim /etc/pam.d/su-l
> Uncomment:
  auth required pam_wheel.so use_uid
```

6.2.6 Realtime

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

```
| pacman -S realtime-privileges;
| usermod -aG realtime myUsername
```

6.3 Pacman configuration

<https://wiki.archlinux.org/title/Pacman#Configuration>

```
| vim /etc/pacman.conf
>
```

6.3.1 Enable Color & Parallel downloads

```
> Uncomment:
  Color
  ...
  ParallelDownloads = 5
```

6.3.2 Enable the multilib repository (for 32-bit: steam, wine, ...)

Note: If you do not use packages that depend on 32-bit packages like [steam](#) & [wine](#), then do not install any 32-bit packages (e.g. lib32) in this guide. Simply do not enable the multilib repository.

```
> Uncomment:
  [multilib]
  Include = /etc/pacman.d/mirrorlist
```

6.3.3 Upgrade System

```
| pacman -Syu
```

6.4 Auto update mirrors – reflector

https://wiki.archlinux.org/title/Reflector#systemd_timer

```
| pacman -S reflector;

| vim /etc/xdg/reflector/reflector.conf
> Uncomment:
  --country Germany

| systemctl enable --now reflector.timer;
```

6.5 Discard unused packages weekly – paccache

https://wiki.archlinux.org/title/Pacman#Cleaning_the_package_cache

```
| systemctl enable paccache.timer;
```

6.6 Limit journal size – systemd/Journal

https://wiki.archlinux.org/title/Systemd/Journal#Journal_size_limit

```
| vim /etc/systemd/journald.conf
```

> Uncomment:

```
SystemMaxUse=200M
```

6.7 Define missing environment variables

https://wiki.archlinux.org/title/Environment_variables#Using_pam_env

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

```
| vim /etc/security/pam_env.conf
```

> Append:

```
# XDG Base Directories
```

```
XDG_CONFIG_HOME DEFAULT=@{HOME}/.config
```

```
XDG_CACHE_HOME DEFAULT=@{HOME}/.cache
```

```
XDG_DATA_HOME DEFAULT=@{HOME}/.local/share
```

```
XDG_STATE_HOME DEFAULT=@{HOME}/.local/state
```

```
# Console Text Editor
```

```
EDITOR DEFAULT=vim
```

6.8 Improving compile times – Makepkg

<https://wiki.archlinux.org/title/Makepkg>

```
| vim /etc/makepkg.conf
```

>

6.8.1 Parallel compilation

READ: https://wiki.archlinux.org/title/Makepkg#Parallel_compilation

```
| cpu_threads := $ nproc
```

> Uncomment:

```
MAKEFLAGS="-j<cpu_threads>"
```

6.9 Enable Periodic TRIM (if TRIM is supported)

https://wiki.archlinux.org/title/Btrfs#SSD_TRIM

https://wiki.archlinux.org/title/Solid_state_drive#TRIM

Note: [TRIM support is disabled by the device-mapper by default](#), see **root**:

6.9.1 Verify TRIM/discard support

```
| lsblk --discard
```

> Non-zero values indicate TRIM support:

NAME	DISC-ALN	DISC-GRAN	DISC-MAX	DISC-ZERO
nvme0n1	0	4K	2T	0
└─nvme0n1p1	0	4K	2T	0
└─nvme0n1p2	0	4K	2T	0
└─root	0	0B	0B	0

6.9.2 Enable Periodic TRIM (enable weekly fstrim)

... on *all* mounted filesystems on devices that support the discard operation.

Note: Continuous TRIM (asynchronous discard > mount option: `discard=async`) should be enabled by default by Btrfs if the underlying device is capable of TRIM.

```
| systemctl enable fstrim.timer
```

7 Install Backend & DE

| `pacman -S <your packages in this chapter>`

7.1 Graphics driver

<https://wiki.archlinux.org/title/Vulkan>

<https://wiki.archlinux.org/title/GPGPU>

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

Note: All graphics units from the last ~7 years *should* be supported with the following packages.

7.1.1 Non-Nvidia (AMD, Intel, ...)

[mesa](#) [lib32-mesa](#)

7.1.2 Intel

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

7.1.2.1 Vulkan

[vulkan-intel](#) [lib32-vulkan-intel](#)

7.1.2.2 Hardware video acceleration – Intel QuickSync (QSV)

[https://wiki.archlinux.org/title/FFmpeg#Intel_QuickSync_\(QSV\)](https://wiki.archlinux.org/title/FFmpeg#Intel_QuickSync_(QSV))

- Gstreamer plugin: [gst-plugin-qsv](#)
- For ≥Gen11 (Tiger Lake): [vpl-gpu-rt](#)
- ELSE For ≥Gen5 (Iron Lake): [intel-media-sdk](#)

7.1.2.3 OpenCL

- For ≥Gen8 (Broadwell): [intel-compute-runtime](#)
- ELSE: [opencl-rusticl-mesa](#)

7.1.3 AMD

<https://wiki.archlinux.org/title/AMDGPU>

7.1.3.1 Vulkan

[vulkan-radeon](#) [lib32-vulkan-radeon](#) (ALT: [amdvk](#) [lib32-amdvk](#))

7.1.3.2 Hardware video acceleration

[libva-mesa-driver](#) [lib32-libva-mesa-driver](#)

7.1.3.3 OpenCL (if not supported by ROCm)

[opencl-rusticl-mesa](#)

7.1.3.4 ROCm: HIP, OpenCL, ... (for Blender, PyTorch, ...)

<https://wiki.archlinux.org/title/GPGPU#ROCm>

> Supported GPUs: [Linux](#), [Windows](#) (more GPUs w/ unofficial Linux support)

7.1.3.4.1 HIP (Heterogeneous-Compute Interface for Portability) & OpenCL

[rocm-hip-runtime](#) [rocm-opencl-runtime](#)

Verify support (>Yes):

```
| /opt/rocm/bin/clinfo | grep -i "image support"
```

7.1.3.4.2 ROCm System Management Interface Library

https://github.com/ROCm/rocm_smi_lib/blob/master/python_smi_tools/README.md

[rocm-smi-lib](#)

```
| rocm-smi -a
```

> Example: RX 6800

GPU	Temp (DieEdge)	AvgPwr	SCLK	MCLK	Fan	Perf	PwrCap	VRAM%	GPU%
0	46.0c	27.0W	500Mhz	456Mhz	0%	auto	215.0W	9%	3%

7.1.3.4.3 PyTorch

[python-pytorch-opt-rocm](#)

Verify support (>True):

```
| python -c 'import torch; print(torch.cuda.is_available())'
```

7.1.4 NVIDIA

READ: <https://wiki.archlinux.org/title/NVIDIA>

> Wayland fix: https://wiki.archlinux.org/title/NVIDIA#DRM_kernel_mode_setting

7.1.4.1 *Install the appropriate driver for your card*

Identify your GPU:

```
| lspci -k | grep -A 2 -E "(VGA|3D)"
```

For \geq [Turing \(NV160/TUXXX\)](#) – (\geq RTX & GTX 16):

- For linux: [nvidia-open](#)
- For other kernels: [nvidia-open-dkms](#) [linux-zen-headers](#)

ELSE For \geq [Maxwell \(NV110/GMXXX\)](#):

- For linux: [nvidia](#)
- For linux-lts: [nvidia-lts](#)
- For other kernels: [nvidia-dkms](#) [linux-zen-headers](#)

7.1.4.2 *NVIDIA drivers utilities*

[nvidia-utils](#) [lib32-nvidia-utils](#)

7.1.4.3 *VA-API Hardware video acceleration*

[libva-nvidia-driver](#)

7.1.4.4 *OpenCL*

[opencl-nvidia](#) [lib32-opencl-nvidia](#)

7.1.4.5 *CUDA*

[cuda](#)

7.1.5 TODO – QEMU Guest: Support for QMP commands

https://wiki.archlinux.org/title/QEMU#Preparing_an_Arch_Linux_guest

[qemu-guest-agent](#) [spice-vdagent](#)

```
> systemctl enable qemu-guest-agent.service;
```

7.1.6 Oracle VirtualBox Guest Additions

https://wiki.archlinux.org/title/VirtualBox/Install_Arch_Linux_as_a_guest

[virtualbox-guest-utils](#) OR [virtualbox-guest-utils-nox](#) (Wayland only)

```
> systemctl enable vboxservice.service;
```

7.1.7 TMP: Xorg DDX driver (for 2D acceleration)

<https://wiki.archlinux.org/title/Xorg>

7.1.7.1 AMD

[xf86-video-amdgpu](#)

7.2 Fonts

<https://wiki.archlinux.org/title/Fonts>

LibreOffice: [ttf-liberation](#) [ttf-carlito](#) (Aptos equivalent is missing)

For KDE (full unicode coverage): [noto-fonts](#) [noto-fonts-cjk](#) [noto-fonts-emoji](#)

HQ, for Firefox, mpv, ...: [adobe-source-sans-fonts](#) [adobe-source-han-sans-otc-fonts](#)
[adobe-source-serif-fonts](#) [adobe-source-han-serif-otc-fonts](#)

Monospaced w/ programming ligatures: [ttf-fira-code](#) (13 pt) (ALT: [ttf-firacode-nerd](#))

7.3 Multimedia frameworks

7.3.1 ALSA (utilities)

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

Note: You may need to configure your audio device using **alsamixer**. **Tip:** Try to lower your "Mic Boost" to reduce noise. This usually also changes the microphone volume.

[alsa-utils](#)

7.3.2 Pipewire

<https://wiki.archlinux.org/title/PipeWire>

<https://wiki.archlinux.org/title/WirePlumber>

Basic: [lib32-pipewire](#)

JACK: [pipewire-jack](#) [lib32-pipewire-jack](#)

ALSA: [pipewire-alsa](#)

Camera: [pipewire-libcamera](#)

Graph-GUI: [qpwgraph](#) or [helvum](#)

7.3.3 Qt Multimedia (default) backend

<https://doc.qt.io/qt-6/qtmultimedia-index.html>

[qt6-multimedia-ffmpeg](#)

7.3.4 Gstreamer

<https://wiki.archlinux.org/title/GStreamer>

7.3.4.1 PipeWire & Libcamera integration

[gst-plugin-pipewire](#) [gst-plugin-libcamera](#)

7.3.4.2 Hardware video acceleration

Non-Nvidia: [gst-plugin-va](#)

Nvidia: [gst-plugins-bad](#)

7.3.4.3 Plugins

[gst-libav](#) [gst-plugins-good](#) [gst-plugins-bad](#) [gst-plugins-ugly](#)

7.3.4.4 TMP: 32-bit support

[lib32-gst-plugins-base](#) [lib32-gst-plugins-good](#)

7.4 WM/DE – KDE Plasma + KDE Gear

https://community.kde.org/Distributions/Packaging_Recommendations

Comparison of Desktop Environments: https://eylenburg.github.io/de_comparison.htm

7.4.1 KDE Plasma

<https://wiki.archlinux.org/title/KDE>

[plasma-meta](#) [flatpak-kcm](#) [kwayland-integration](#)

7.4.2 KDE Gear

<https://apps.kde.org> > See: [kde-applications-meta](#)

KDE Meta Package	Recommended & Optional apps
kde-system-meta	<All apps / Install meta pkg>
kde-utilities-meta	ark filelight isoimagewriter kate kcalc kcharselect kclock kdebugsettings kdialog kfind konsole ktimer kwalletmanager markdownpart skanpage yakuake
kde-graphics-meta	colord-kde gwenview kamera kcolorchooser kdegraphics-thumbnailers okular skanlite spectacle svgpart
kde-multimedia-meta	audex audiocd-kio ffmpegthumbs kdenlive
kde-network-meta	kdeconnect sshfs kdenetwork-filesharing krdc freerdp neochat tokodon
kde-pim-meta	kleopatra merkuro
kde-sdk-meta	dolphin-plugins kompare

7.4.3 OPT: (3rd-party) Packages w/ KDE integration [deps]

- Power profiles (balanced, power-saver, performance): [power-profiles-daemon](#)
> `systemctl enable power-profiles-daemon.service;`
- Hybrid/Multi-GPU support: [switcheroo-control](#)
- Automatic screen rotation: [iio-sensor-proxy](#)
- On-screen keyboard: [maliit-keyboard](#)
- Streaming Wayland windows to X applications: [xwaylandvideobridge](#) [no dep]

7.4.4 Image formats (JPEG XL Master Race) [deps]

https://wiki.archlinux.org/title/Dolphin#File_previews

[kimageformats](#) [qt6-imageformats](#) [libavif](#)

7.4.5 Misc. [deps]

- File archiver – 7z & RAR: [p7zip](#) [unrar](#)
- Faster find than find (for [kfind](#)): [plocate](#) > `systemctl enable plocate-updatedb.timer`

7.5 Misc.

<https://wiki.archlinux.org/title/Laptop>

7.5.1 Hybrid graphics

> https://wiki.archlinux.org/title/Hybrid_graphics

> <https://wiki.archlinux.org/title/PRIME>

7.5.1.1 NVIDIA PRIME render offload

[nvidia-prime](#)

7.5.2 Optical disk: Playback

https://wiki.archlinux.org/title/Optical_disc_drive#Playback

7.5.2.1 CD

[libcdio](#)

7.5.2.2 DVD

[libdvdread](#) [libdvdcss](#) [libdvdnav](#)

7.5.2.3 Blu-ray

READ: <https://wiki.archlinux.org/title/Blu-ray#Playback>

[libbluray](#) [libaacs](#) [libbdplus](#)^{AUR}

7.6 After installation

7.6.1 Change installation reason of opt. dependencies

Note: Search for the tag **[deps]** in this chapter.

```
| pacman -D --asdeps [deps];
```

8 Display Manager – SDDM (for KDE)

<https://wiki.archlinux.org/title/SDDM>

8.1 Configuration

8.1.1 OPT: Enable virtual keyboard

https://wiki.archlinux.org/title/SDDM#Enable_virtual_keyboard

TODO: Use [maliit-keyboard](#).

```
| vim /etc/sddm.conf.d/virtualkbd.conf  
>  
[General]  
InputMethod=maliit-keyboard
```

8.2 Enable SDDM & Reboot into SDDM/KDE

```
| systemctl enable sddm.service;  
| systemctl reboot
```

8.3 Congratulations on reaching this point!

... but this guide is not done with you!

Check if hibernation is working (if it has been set up):

```
| systemctl hibernate
```

8.4 Change SDDM theme to "Breeze"

Open "System Settings" > **Search:** "Login Screen (SDDM)": **Breeze** > Apply

9 Btrfs snapshots

9.1 Setup snapshots using snapper & snap-pac

<https://wiki.archlinux.org/title/Snapper>

9.1.1 Open interactive shell with root prompt for this chapter

```
| doas -s
```

9.1.2 Installation

```
| pacman -S snapper;
```

9.1.3 Configuration of snapper and mount point

https://wiki.archlinux.org/title/Snapper#Configuration_of_snapper_and_mount_point

9.1.3.1 */.snapshots/ must not exist (for snapper create-config /)*

```
| umount /.snapshots;  
| rmdir /.snapshots/;
```

9.1.3.2 *Create the configuration file for the subvolume mounted at /*

```
| snapper -c root create-config /;
```

9.1.3.3 *Delete the subvolume automatically created by snapper*

```
| btrfs subvolume delete /.snapshots;
```

OPT – Check:

```
| btrfs subvolume list /
```

9.1.3.4 *Recreate, Remount & Change permissions of /.snapshots/*

```
| mkdir /.snapshots;  
| mount -a;  
| chmod 750 /.snapshots;
```

9.1.4 Other configurations

9.1.4.1 Lower snapshot limits & Disable timeline snapshots (for snap-pac)

Note: Snap-pac creates pre/post snapshots (with numbers).

```
| vim /etc/snapper/configs/root
>
NUMBER_CLEANUP="yes"
...
NUMBER_LIMIT="20" (Note: For 10 pacman transactions; for number cleanup algorithm)
...
TIMELINE_CREATE="no"
...
TIMELINE_CLEANUP="no"
```

9.1.4.2 Enable systemd/timer for cleanup

```
| systemctl enable snapper-cleanup.timer;
```

9.1.4.3 Also snapshot the ESP

Modified: https://wiki.archlinux.org/title/System_backup#Snapshots_and_/boot_partition

```
| pacman -S --needed rsync;

| vim /etc/pacman.d/hooks/95-efibackup.hook
>
[Trigger]
Operation = Upgrade
Operation = Install
Operation = Remove
Type = Path
Target = usr/lib/modules/*/vmlinuz

[Action]
Depends = rsync
Description = Backing up /efi ...
When = PostTransaction
Exec = /usr/bin/rsync -a --delete /efi /.efibackup
```

9.1.5 Wrapping pacman transactions in snapshots

<https://wesbarnett.github.io/snap-pac/configuration.html>

Note: Snapper will create the snapshots when pacman installs, upgrades or removes a package.

```
| pacman -S snap-pac;
```

9.1.6 Create pre/post snapshots for the next step

... by installing e.g. [nano](#). After the restoration, nano is no longer installed.

9.2 Restoring / (subvolume @) to its previous snapshot

Note: Restore now to check if it works and later if the system is broken.

9.2.1 Disable Secure Boot & Boot into a live Arch Linux environment

9.2.2 Open LUKS container & Mount the btrfs volume

```
cryptsetup open /dev/nvme0n1p2 root;  
mount /dev/mapper/root /mnt
```

9.2.3 Delete the old backup & Backup "broken" root subvolume

Delete the old "broken" backup:

```
btrfs subvolume delete /mnt/@.broken
```

Backup:

```
mv /mnt/@ /mnt/@.broken
```

9.2.4 Find the snapshot number <num> that you want to recover

```
grep -r '<date>' /mnt/@snapshots/*/info.xml  
> /mnt/@snapshots/<num>/info.xml: <date>yyyy-mm-dd hh:mm:ss</date>
```

9.2.5 Create a read-write snapshot of the read-only snapshot

```
btrfs subvolume snapshot /mnt/@snapshots/<num>/snapshot /mnt/@
```

9.2.6 1st setup: Delete "subvolid" mount option in fstab

... since the `subvol=@` mount option is sufficient.

```
vim /mnt/@/etc/fstab  
> UUID=... / btrfs OPTIONS, subvolid=<...>, subvol=@ 0 0
```

9.2.7 Restore EFI executables from /.efibackup/

Note: Skip if this is your first run.

```
mount /dev/nvme0n1p1 /mnt/@/efi/;  
cp -r /mnt/@/.efibackup/efi/EFI/Linux/* /mnt/@/efi/EFI/Linux/;
```

9.2.8 Unmount & Reboot

```
umount -R /mnt;  
systemctl reboot
```

9.2.9 Boot problems?

[Change root](#) to your restored snapshot in order to regenerate your UKIs.

9.2.10 Fix pacman error: "failed to synchronize all databases"

[https://wiki.archlinux.org/title/](https://wiki.archlinux.org/title/Pacman#Failed_to_init_transaction_(unable_to_lock_database)_error)

[Pacman#"Failed_to_init_transaction_\(unable_to_lock_database\)"_error](#)

```
| doas rm /var/lib/pacman/db.lck
```

9.2.11 For Secure Boot

Enable Secure Boot & Reboot.

10 Security – Hardening Arch Linux

<https://wiki.archlinux.org/title/Security>

GOAL: Creation of a secure *and* useful system.

10.1 Open interactive shell with root prompt for this chapter

```
| doas -s
```

10.2 Restrict programs' capabilities – AppArmor

https://wiki.archlinux.org/title/Security#Mandatory_access_control

<https://wiki.archlinux.org/title/AppArmor>

10.2.1 Installation

10.2.1.1 Kernel parameters (for UKI)

```
| vim /etc/cmdline.d/security.conf  
>  
# AppArmor  
lsm=landlock, lockdown,yama,integrity,apparmor,bpf
```

Note: Make sure that *apparmor* is the first "major" module in the list.

10.2.1.2 Kernel lockdown (also for Secure Boot)

https://wiki.archlinux.org/title/Security#Kernel_lockdown_mode
[kernel_lockdown\(7\)](#), [Protecting Secure Boot](#)

```
| vim /etc/cmdline.d/security.conf  
>  
# Kernel lockdown  
lockdown=integrity
```

10.2.1.3 Regenerate all UKIs

```
| mkinitcpio -P
```

10.2.1.4 Install & Enable AppArmor

```
| pacman -S apparmor;  
systemctl enable apparmor.service;
```

10.2.2 Speed-up AppArmor start by caching profiles

```
| vim /etc/apparmor/parser.conf  
> Uncomment:  
# Turn creating/updating of the cache on by default  
write-cache
```


10.2.3 Reboot

```
| systemctl reboot
```

10.2.4 Verify

```
| aa-enabled;  
| aa-status;
```

10.3 Sandboxing applications – Firejail

https://wiki.archlinux.org/title/Security#Sandboxing_applications

<https://wiki.archlinux.org/title/Firejail>

10.3.1 Installation

```
| pacman -S firejail;
```

10.3.2 Enable AppArmor support

[firejail\(1\) § APPARMOR](#)

```
| apparmor_parser -r /etc/apparmor.d/firejail-default;
```

10.3.3 Add user to firejails user access database

```
| firecfg --add-users myUsername;
```

10.3.4 Use Firejail by default for all applications w/ a profile

```
| firecfg;
```

10.3.5 Automatically run firecfg on pacman operations

```
| vim /etc/pacman.d/hooks/firejail.hook
>
[Trigger]
Type = Path
Operation = Install
Operation = Upgrade
Operation = Remove
Target = usr/bin/*
Target = usr/share/applications/*.desktop

[Action]
Description = Configure symlinks in /usr/local/bin based on firecfg.config...
When = PostTransaction
Depends = firejail
Exec = /bin/sh -c 'firecfg >/dev/null 2>&1'
```

10.3.6 Verify if running applications are sandboxed

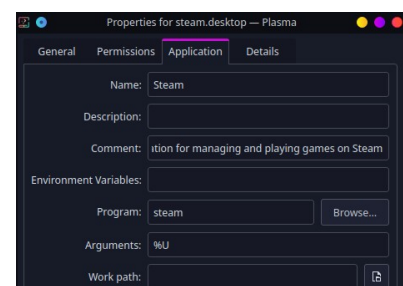
```
| firejail --list
```

> If running **<app>** is not sandboxed:

```
| vim ~/.local/share/applications/<app>.desktop
```

> Modify: Exec=...

> ALT – KDE: "Edit Application..." >



10.3.7 Creating firejail overrides (if an app is not working properly)

<https://github.com/netblue30/firejail/wiki/Creating-overrides>

10.3.7.1 Verify if the app is working without firejail

```
| firejail --noprofile <app>
```

10.3.7.2 Read the profile of the app for override recipes

Note: Not every recipe is listed in these profiles. Search through the [issues of firejail](#).

```
| less /etc/firejail/<app>.profile
```

10.3.7.3 Create the overrides

Note: `globals.local` will affect every regular profile.

Note: You can take a look at my firejail locals, esp. `{firefox, keepassxc, steam}.local`

```
| vim ~/.config/firejail/<app>.local
```

10.3.8 Notes

- The `~/Downloads` directory can be considered as a shared directory. So do not store any sensitive files there.
- You can [create your own profiles](#) in `~/.config/firejail/<app>.profile`
- NVIDIA users should read [this](#)

10.4 More Kernel Hardening

https://wiki.archlinux.org/title/Security#Kernel_hardening

<https://wiki.archlinux.org/title/Sysctl>

10.4.1 Print current value of <parameter>

```
| sysctl -a | grep <parameter>
```

10.4.2 BPF hardening

https://wiki.archlinux.org/title/Security#BPF_hardening

Disable BPF of unprivileged code:

```
| vim /etc/sysctl.d/99-sysctl.conf  
> kernel.unprivileged_bpf_disabled = 1
```

10.4.3 Reverse path filtering (loose -> strict)

https://wiki.archlinux.org/title/Sysctl#Reverse_path_filtering

... for source validation of the packets received from all the network interfaces.

```
| vim /etc/sysctl.d/99-sysctl.conf  
>  
net.ipv4.conf.default.rp_filter = 1  
net.ipv4.conf.all.rp_filter = 1
```

10.5 Firewall – Firewalld

<https://wiki.archlinux.org/title/Firewalld>

10.5.1 Install, Enable & Reboot

```
| pacman -S firewalld;  
| systemctl enable firewalld.service;  
| systemctl reboot
```

10.5.2 Changing zone to "home" of your network interface (for nm)

KDE Connections (GUI) or:

```
| firewall-cmd --zone=home --change-interface=<interface>
```

Note: Interface is the same as:

```
| ip addr show <interface>
```

Note: default_zone=public

10.5.3 Adding services to the zone "home"

E.g. add KDE Connect:

Check available services:

```
| firewall-cmd --get-services | grep kde
```

Add service:

```
| firewall-cmd --permanent --zone=home --add-service=kdeconnect;  
| firewall-cmd --reload
```

10.6 Restricting root login

https://wiki.archlinux.org/title/Security#Restricting_root_login

```
| passwd --lock root;
```

10.7 Mount hardening of the ESP

... by using systemd GPT partition automounting w/ more secure mount options.

```
| vim /etc/fstab  
> Comment out  
# /dev/nvme0n1p1  
#UUID=... /efi vfat
```

10.8 fwupd (Firmware updater & Verify platform security)

<https://wiki.archlinux.org/title/Fwupd>

Devices supported by LVFS: <https://fwupd.org/lvfs/devicelist>

[fwupd](#)

10.8.1 Update firmware & UEFI dbx

Note – for KDE & Gnome: Use the GUI, or:

```
| fwupdmgr get-updates
```

10.8.2 Host Security ID (HSI)

<https://fwupd.github.io/libfwupdplugin/hsi.html>

User Host Security Reports: <https://fwupd.org/lvfs/hsireports/devices>

GOAL: [HSI:3 \(Protected State\)](#)

Note: You may have to contact your motherboard or system manufacturer to verify the platform security, see: "[Supported CPU](#)".

10.8.2.1 Verifying Host Firmware Security

```
| fwupdmgr security
```

10.8.3 Setup for UEFI upgrade (using own keys & pacman hook)

https://wiki.archlinux.org/title/Fwupd#Using_your_own_keys

```
| vim /etc/pacman.d/hooks/sign-fwupd-secureboot.hook
>
[Trigger]
Operation = Install
Operation = Upgrade
Type = Path
Target = usr/lib/fwupd/efi/fwupdx64.efi

[Action]
Description = Signing fwupdx64.efi for SecureBoot...
When = PostTransaction
Exec = /usr/bin/sbsign --key /root/efi-keys/db.key --cert /root/efi-
keys/db.crt /usr/lib/fwupd/efi/fwupdx64.efi
Depends = sbsigntools
```

```
| vim /etc/fwupd/fwupd.conf (See: man 5 fwupd.conf)
>
[uefi_capsule]
DisableShimForSecureBoot=true
```

10.9 Harden yourself

> <https://wiki.archlinux.org/title/Security>

Useful applications: https://wiki.archlinux.org/title/List_of_applications/Security

11 User-specific configurations

11.1 kdesu: Use "doas" instead of "su" (for KDE)

<https://wiki.archlinux.org/title/Sudo#kdesu>

```
| vim ~/.config/kdesurc  
>  
[super-user-command]  
super-user-command=doas
```

11.2 AUR helper & Pacman wrapper – paru

https://wiki.archlinux.org/title/AUR_helpers#Pacman_wrappers

<https://github.com/morganamilo/paru>

11.2.1 Install paru + opt. dependencies

```
| doas pacman -S --needed --asdeps git rust devtools bat;  
  
| git clone https://aur.archlinux.org/paru.git;  
| cd paru; makepkg -si  
  
| cd ..; rm -rf paru/;
```

11.2.2 Configuration: Use doas instead of sudo

See: man [paru.conf](#)

```
| mkdir ~/.config/paru;  
| wget -P ~/.config/paru/  
| https://raw.githubusercontent.com/Morganamilo/paru/master/paru.conf;  
  
| vim ~/.config/paru/paru.conf  
> Uncomment:  
[bin]  
Sudo = doas
```

11.2.3 Important commands (pacman wrapper)

System Upgrade:	\$ paru	:= # pacman -Syu
	\$ paru -Sua	:= Upgrade only AUR packages
Install <pkg>:	\$ paru -S <pkg>	:= # pacman -S <pkg>
	\$ paru "<pkg>"	:= Search & Install "<pkg>"

11.2.4 Skip certain (AUR) packages from being upgraded

READ: https://wiki.archlinux.org/title/Pacman#Skip_package_from_being_upgraded

```
| doas vim /etc/pacman.conf  
> IgnorePkg = <pkg1> <pkg2> ... <pkgN>
```

11.3 Reduce output latency

Note: Setting these options may cause tearing and short-lived artifacts to appear.

11.3.1 Disable synchronization to vblank (Disable VSYNC)

https://wiki.archlinux.org/title/Gaming#Reducing_DRI_latency

Intel: [https://wiki.archlinux.org/title/Intel_graphics#Disable_Vertical_Synchronization_\(VSYNC\)](https://wiki.archlinux.org/title/Intel_graphics#Disable_Vertical_Synchronization_(VSYNC))

```
| vim $HOME/.dirc
>
<driiconf>
    <device>
        <application name="Default">
            <option name="vblank_mode" value="0" />
        </application>
    </device>
</driiconf>
```

11.3.2 For Xorg: Disable "PageFlip" & "TearFree" (Here: amdgpu)

https://wiki.archlinux.org/title/AMDGPU#Reduce_output_latency

Manuals: <https://man.archlinux.org/man/amdgpu.4> | <https://man.archlinux.org/man/intel.4>

For Intel – Change: | [amdgpu](#) → [intel](#) | [AMD](#) → [Intel Graphics](#) | [More Intel options](#)

```
| doas vim /etc/X11/xorg.conf.d/20-amdgpu.conf
>
Section "OutputClass"
    Identifier "AMD"
    MatchDriver "amdgpu"
    Driver "amdgpu"
    Option "EnablePageFlip" "false"
    Option "TearFree" "false"
EndSection
```

11.3.3 For KDE (X11): Disable tear prevention

<https://invent.kde.org/plasma/kwin/-/wikis/Environment-Variables>

```
| doas vim /etc/environment
> Append:
    KWIN_X11_NO_SYNC_TO_VBLANK=1
```

11.3.4 For AMD ROCm: Set power profile (for e.g. compute)

List power profiles:

```
| rocm-smi -l
```

For Gaming (Default in Arch Linux):

```
| doas rocm-smi --setprofile "3D FULL SCREEN"
```

Reset profile:

```
| doas rocm-smi --resetprofile
```


11.4 PipeWire optimizations

<https://gitlab.freedesktop.org/pipewire/pipewire/-/wikis>

<https://pipewire.pages.freedesktop.org/wireplumber/>

11.4.1 Performance tuning

<https://gitlab.freedesktop.org/pipewire/pipewire/-/wikis/Performance-tuning#limits>

```
| doas vim /etc/security/limits.d/95-pipewire.conf
>
# Default limits for users of pipewire
@pipewire - rtprio 95
@pipewire - nice -19
@pipewire - memlock 4194304
```

```
| doas groupadd -r pipewire;
| doas usermod -aG pipewire myUsername;
```

11.4.2 Create directories to copy sections of a config file

```
| mkdir -p ~/.config/pipewire/{pipewire{,-pulse},jack,client{,-rt}}.conf.d;
```

Note: The default config files are located in /usr/share/{pipewire,wireplumber}/

11.4.3 Increase resample quality

<https://gitlab.freedesktop.org/pipewire/pipewire/-/wikis/Config-PipeWire#setting-resample-quality>

Note: Increasing the quality uses more CPU power with (arguably) little measurable advantages.

Default: resample.quality = 4
Better: resample.quality = 10
Best: resample.quality = 15 (At least double CPU usage than with "10")

```
| vim ~/.config/pipewire/client.conf.d/resample-quality.conf
| vim ~/.config/pipewire/client-rt.conf.d/resample-quality.conf
| vim ~/.config/pipewire/pipewire-pulse.conf.d/resample-quality.conf
>
stream.properties = {
    resample.quality      = 10
}
```

11.4.4 Auto change sample rate

<https://gitlab.freedesktop.org/pipewire/pipewire/-/wikis/Guide-Rates>

<https://gitlab.freedesktop.org/pipewire/pipewire/-/wikis/Config-PipeWire#setting-sample-rates>

... to avoid up- & downsampling. 48 kHz should still be the default sample rate.

Note: This *may* cause minor problems.

11.4.4.1 Get sample rates supported by your audio device

Note: The sample rates 44.1, 48 & 96 kHz should always be supported.

```
| cat /proc/asound/card*/stream0
> E.g. Rates: 32000, 44100, 48000, 88200, 96000, 176400, 192000, 352800, 384000
```

11.4.4.2 Commit changes

```
| vim ~/.config/pipewire/pipewire.conf.d/10-rates.conf
>
context.properties = {
    default.clock.allowed-rates = [ 44100 48000 88200 96000 176400 192000 352800
384000 ]
}
```

11.4.5 Reduce default latency: Decrease quantum (buffer size)

... if the client does not specify a quantum.

11.4.5.1 Infos

$\text{latency} = \text{quantum} / \text{sampleRate}$

$\text{default_latency} = \text{default_quantum} / \text{default_sampleRate} = 1024 / 48 \text{ kHz} = 21.3 \text{ ms}$

Note: When the graph is using the 96000 samplerate ($\text{default_sampleRate} * 2$), the quantum values are scaled ($\text{default_quantum} * 2 = 2048 \leq \text{quantum-limit}$).

11.4.5.2 Get a suitable quantum for your audio device

Temporarily force the graph to operate at a lower fixed buffer size until cracking noises occur:

```
| pw-metadata -n settings 0 clock.force-quantum <quantum>
```

> E.g. new $\text{default_quantum} = 64$

> $\text{default_latency} = 64 / 48 \text{ kHz} = 1.3 \text{ ms}$

Note: In order for the setting to be adopted, you may pause the audio stream for a short time.

Note: When $\text{sampleRate} = 44100 \text{ Hz} < \text{default_sampleRate}$

→ $\text{quantum} = \text{default_quantum} / 2$

→ $\text{latency} \approx \text{default_latency} / 2$

So at 32/44100 you should not hear any disturbing noises.

ALT – Set: $\text{default.clock.min-quantum} = 64$ (= $\text{default.clock.quantum}$)

11.4.5.3 Change property for the DSP config

```
| vim ~/.config/pipewire/pipewire.conf.d/10-quantum.conf
>
context.properties = {
    default.clock.quantum = 64
}
```

11.4.6 Verify your changes

Monitoring:

```
| pw-top
```

ALSA status:

```
| cat /proc/asound/card*/pcm*p/sub*/hw_params
```

12 Hardware specific

12.1 Sensors

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

12.1.1 Setup

Note: Only use default options (by just hitting enter), unless you know exactly what you are doing.

```
| doas sensors-detect  
> enter
```

12.1.2 Check sensors

```
| sensors  
> No mainboard sensors?: https://wiki.archlinux.org/title/Lm\_sensors#Troubleshooting
```

12.1.3 Tip: Print sensors values periodically

```
| watch sensors
```

12.2 OPT: Fan speed control

> https://wiki.archlinux.org/title/Fan_speed_control

12.3 OPT: Stress testing

> https://wiki.archlinux.org/title/Stress_testing

12.4 Printer & Scanner

12.4.1 Printer (CUPS)

<https://wiki.archlinux.org/title/CUPS>

- **READ:** https://wiki.archlinux.org/title/CUPS/Printer-specific_problems
ALT: [Other printer drivers](#)
- **Basic:** [cups system-config-printer](#)
| > **doas** systemctl enable cups.socket

12.4.2 Scanner (SANE)

<https://wiki.archlinux.org/title/SANE>

- **READ:** https://wiki.archlinux.org/title/SANE/Scanner-specific_problems
- **Basic:** [sane](#)
- **"Driverless" scanning:** [sane-airscan](#) (Supported devices: [1](#) & [2](#))
- **IPP over USB:** [ipp-usb](#)
| > **doas** systemctl enable ipp-usb.service

12.5 Fingerprint reader

> <https://wiki.archlinux.org/title/Fprint> (Supported devices)

> **For SDDM:** https://wiki.archlinux.org/title/SDDM#Using_a_fingerprint_reader
[fprintd](#) [imagemagick](#)

12.6 RGB Control

> OpenRGB: <https://openrgb.org/devices.html>

[openrgb](#)

12.7 Peripherals

12.7.1 Mice

List of flawless mice: <https://sensor.fyi/mice/>

Note: Mice like the ones from Zowie do not need any software to change the CPI.

Software	Supported mice
piper	https://github.com/libratbag/libratbag/tree/master/data/devices

12.7.2 Keyboards

Software	Supported keyboards
qmk	(Custom) keyboards: https://qmk.fm/keyboards/

12.7.3 Logitech peripherals – Solaar

> Solaar: <https://pwr-solaar.github.io/Solaar/devices> (partial list)

[solaar](#)

12.7.4 Razer peripherals – OpenRazer

> https://wiki.archlinux.org/title/Razer_peripherals#OpenRazer

12.8 Desktop-PC (DDR5) – Hardware Recommendations

12.8.1 Mainboard

- **Sensor drivers** often available (a bit later after release) for:
[ASUS](#), ASRock?, Biostar?, Gigabyte?, MSI?
> https://wiki.archlinux.org/title/Lm_sensors#Troubleshooting
- **NIC (Ethernet & Wireless):** Intel > Realtek

12.8.2 Discrete GPU (dGPU)

12.8.2.1 For Gaming

- AMD >> NVIDIA (causes problems quite often, also proprietary) > Intel (new)

12.8.2.2 For Computing

> <https://wiki.archlinux.org/title/GPGPU>

12.8.3 CPU

12.8.3.1 Notes

- Minor performance degradation due to future microcode mitigations: AMD (-) > Intel (--)

12.8.3.2 For hardware-isolated virtualization

See: [Virtualization](#)

- AMD Pro CPU w/ SEV
- Intel CPU w/ TDX or SGX2

12.8.3.3 Esp. for PCI passthrough via OVMF

- CPU w/ iGPU (integrated graphics)

12.8.4 Display

- Overclocked panels ("OC mode") *can* cause issues
- DisplayPort (DP) > HDMI

12.8.5 Disk

- $4Kn \geq 512e/4Kn > 512e > 512n$...n = native, e = emulation (**See:** [Advanced Format](#))
- **External backup disk:** Internal HDD (non-SMR) + [UASP](#) USB case

13 Software specific

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

Note: You can take a look at my configuration files.

13.1 Copy config files (Here: using sftp)

... or simply use a USB drive.

Connect:

```
sftp myUsername@<ip addr>  
For VirtualBox: sftp -P 3022 myUsername@localhost
```

Change working directory:

```
cd Downloads/;
```

Copy your local configuration files to the new system:

```
put -r /home/myLocalUsername/path/to/config/
```

13.2 Zsh

<https://wiki.archlinux.org/title/Zsh>

13.2.1 Install & Change shell (for user & root)

```
doas pacman -S --needed zsh;
```

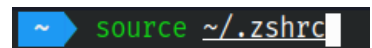
```
chsh -s /usr/bin/zsh;  
doas chsh -s /usr/bin/zsh;
```

13.2.2 Interactive shell configuration

Note: You can take a look at my configuration files. However, you should first install the packages contained in `zsh.txt`.

- `/etc/zsh/zshrc`

13.2.3 Theme: Starship (w/ nerd font)



<https://starship.rs/>

Note: Also compatible with other OSs and shells such as Bash or PowerShell.

```
doas pacman -S --needed starship ttf-firacode-nerd;
```

13.2.3.1 Configure your shell to initialize starship

```
doas vim /etc/zsh/zshrc  
> Append:  
eval "$(starship init zsh)"
```


13.2.3.2 Get a init preset

<https://starship.rs/presets/>

... as a base to configure further to your liking.

```
| starship preset <preset> -o ~/.config/starship.toml
```

13.2.4 Configuring Zsh \$PATH

[https://wiki.archlinux.org/title/Zsh#Configuring_\\$PATH](https://wiki.archlinux.org/title/Zsh#Configuring_$PATH)

Path	Info
~/.local/bin	User-specific executable files (recommended by XDG)
~/.cargo/bin	Rust binary crates (using <code>cargo install</code>)
/opt/rocm/bin	AMD ROCm executables (if installed earlier)

```
| vim ~/.zshenv
>
typeset -U path PATH
path=(~/.local/bin ~/.cargo/bin /opt/rocm/bin $path)
export PATH

| source ~/.zshenv;
```

13.3 Install your packages from text files (using zsh)

```
doas pacman -S --needed - < *.txt
paru -S --needed - < AUR/*.txt
```

13.4 Firefox

<https://wiki.archlinux.org/title/Firefox>

[firefox](#) [firefox-i18n-en-us](#) [firefox-i18n-de](#)

13.4.1 Firefox profile on RAM (using profile-sync-daemon)

<https://wiki.archlinux.org/title/Profile-sync-daemon>

https://wiki.archlinux.org/title/Firefox/Profile_on_RAM

[profile-sync-daemon](#)

13.4.1.1 Relocate cache to RAM only (disable disk cache)

If not using Arkenfox user.js:

> Type in address bar of Firefox (Ctrl+L): `about:config`

> `browser.cache.disk.enable = false`

13.4.1.2 Create configuration file

```
psd
```

13.4.1.3 Edit configuration file

```
vim ~/.config/psd/psd.conf
> BROWSERS=(firefox)
```

13.4.1.4 Enable user service

```
systemctl enable --now --user psd.service;
```

13.4.2 OPT: Tor Browser (Disable Firejail; Use AA)

[torbrowser-launcher](#)

```
doas sed -i 's/torbrowser-launcher/#torbrowser-launcher/'
/etc/firejail/firecfg;
doas firecfg;
```

Config: `~/.local/share/torbrowser/tbb/x86_64/tor-browser/Browser/TorBrowser/Data/`

13.4.3 Setup Firefox (using arkenfox user.js & uBlock Origin)

13.4.3.1 Goals

- Privacy & enhanced security
- Reduction of tracking & fingerprinting
- Side effects: Faster page load; Better battery, memory & network bandwidth usage
- Quality of Life

13.4.3.2 Arkenfox user.js (hardened config template)

> <https://github.com/arkenfox/user.js/wiki>

13.4.3.2.1 Create a new profile

Enter in address bar of Firefox: `about:profiles`

> Click on "Create a New Profile"

13.4.3.2.2 Change to "Root Directory"

```
| cd ~/.mozilla/firefox/XXX.profileName;
```

13.4.3.2.3 Download the update scripts & make them executable

```
| wget https://raw.githubusercontent.com/arkenfox/user.js/master/prefsCleaner.sh  
| https://raw.githubusercontent.com/arkenfox/user.js/master/updater.sh;
```

```
| chmod +x ./updater.sh ./prefsCleaner.sh;
```

13.4.3.2.4 Create the user-overrides.js (for updater)

<https://github.com/arkenfox/user.js/wiki/3.1-Overrides>

[https://github.com/arkenfox/user.js/wiki/3.2-Overrides-\[Common\]](https://github.com/arkenfox/user.js/wiki/3.2-Overrides-[Common])

... which will change or add settings to the `user.js`.

Note: You can take a look at my `user-overrides.js` (also with some QoL settings).

13.4.3.2.5 Update user.js (~every quarter)

<https://github.com/arkenfox/user.js/wiki/3.4-Apply-&-Update-&-Maintain>

<https://github.com/arkenfox/user.js/wiki/3.5-prefsCleaner>

Note: Make sure Firefox is closed.

```
| ./updater.sh; ./prefsCleaner.sh
```

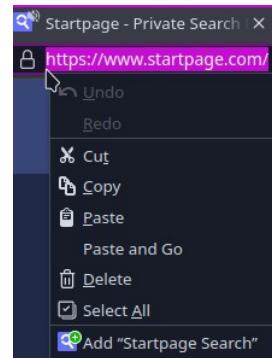
13.4.3.3 Search engines

In address bar: `about:preferences#search`

> **Delete Search Shortcuts:** Google, Bing, ...

> **Add** a privacy-friendly search engine (→)

- <https://www.startpage.com> (uses Google search)
- <https://searx.space> (> choose a [SearXNG](#) instance)
- <https://www.qwant.com> (uses also Bing search)
- <https://duckduckgo.com> (USA, but has good keyboard navigation)



13.4.3.4 Extensions

13.4.3.4.1 Basic

<https://github.com/arkenfox/user.js/wiki/4.1-Extensions>

- **uBlock Origin** – <https://addons.mozilla.org/firefox/addon/ublock-origin/>
- **Skip Redirect** – <https://addons.mozilla.org/firefox/addon/skip-redirect/>
Note: Deactivate for e.g. Wi-Fi hotspot logins

13.4.3.4.2 Extras

- **KDE Plasma Integration** – <https://addons.mozilla.org/firefox/addon/plasma-integration/>
- **KeePassXC-Browser** – <https://addons.mozilla.org/firefox/addon/keepassxc-browser/>
Integration of the password manager KeePassXC
> In KeePassXC: Enable browser integration for Firefox
- **LibRedirect** – <https://addons.mozilla.org/firefox/addon/libredirect/>
Redirects YouTube, Twitter, Instagram, ... to privacy-friendly frontends & backends
- **OPT: Violentmonkey** – <https://addons.mozilla.org/firefox/addon/violentmonkey/>
User Script manager
- **OPT: Binnen-I be gone** – <https://addons.mozilla.org/de/firefox/addon/binnen-i-be-gone/>
For Germans: Filtert Binnen-Is

13.4.3.5 uBlock Origin: Hard mode (aka. block 3rd-party)

<https://github.com/gorhill/uBlock/wiki/Blocking-mode> > Hard mode

<https://github.com/gorhill/uBlock/wiki/Quick-guide:-popup-user-interface>

13.4.3.5.1 Enable hard mode

Settings pane:

- I am an advanced user: **checked**

Filter lists pane:

- All of uBO's custom filter lists: **checked**
- EasyList: **checked**
- Peter Lowe's Ad server list: **checked**
- EasyPrivacy: **checked**
- Online Malicious URL Blocklist: **checked**
- Import ...: Paste link of [Actually Legitimate URL Shortener Tool](#)
- > "Apply Changes"

My rules pane – Add:

```
* * 3p block
* * 3p-script block
* * 3p-frame block
```

13.4.3.5.2 OPT: Add global whitelist rules for all sites

My rules pane – Example:

```
* akamai.net * noop
* akamaiedge.net * noop
* akamaihd.net * noop
* cloudflare.com * noop
* cloudflare.net * noop
* cloudfront.net * noop
* documentfoundation.org * noop
* fastly.net * noop
* freedesktop.org * noop
* github.io * noop
* githubusercontent.com * noop
* hwdn.net * noop
* imgur.com * noop
* jquery.com * noop
* jsdelivr.net * noop
* kde.org * noop
* openstreetmap.org * noop
* shopify.com * noop
* sstatic.net * noop
* wikimedia.org * noop
* wordpress.com * noop
* wp.com * noop
```

13.4.3.6 OPT – Theme: Line over tab (like in Photon – FF88)

> Copy my chrome/ directory into your profile directory > Change line color (**#e84d0e**)

Note: Style generated from <https://www.userchrome.org/firefox-89-styling-proton-ui.html#tabstyler>

13.5 LibreOffice

<https://wiki.archlinux.org/title/LibreOffice>

<https://help.libreoffice.org>

- [libreoffice-still](#) (Stable maintenance branch) OR
- [libreoffice-fresh](#) (Feature branch)

13.5.1 OPT: Language pack only for localized user interface

- [libreoffice-still-de](#) OR
- [libreoffice-fresh-de](#)

13.5.2 Language aids

- Spell checking: [hunspell](#) [hunspell-en_us](#) [hunspell-de](#)
- Hyphenation rules: [hyphen](#) [hyphen-de](#)
- Thesaurus: [libmythes](#) [mythes-en](#) [mythes-de](#)
- Grammar checking: [languagetool](#)
 - `systemctl enable languagetool.service;`
 - LO Options > Languages & Locales:
 - LanguageTool Server: Enable & Base URL: <http://localhost:8081/v2>
 - Writing Aids: Enable LanguageTool

13.5.3 Configuration

Delete or Backup old config (also if encountering upgrade issues):

```
| rm -rf ~/.config/libreoffice/
```

Open LibreOffice > Tools > Options:

13.5.3.1 LO Writer config

- **Basic Fonts (Western):** Liberation Sans, 11 pt
... if you do not intend to print your works frequently.
- **Grid > Check** "Snap to Grid"

13.6 Gamemode (for low latency)

<https://wiki.archlinux.org/title/Gamemode>

[gamemode](#) [lib32-gamemode](#)

13.6.1 Join "gamemode" group to allow renicing

```
| doas usermod -aG gamemode myUsername;
```

13.6.2 Configuration

```
| wget -P ~/.config/  
| https://raw.githubusercontent.com/FeralInteractive/gamemode/master/example/  
| gamemode.ini;
```

```
| vim ~/.config/gamemode.ini  
> renice=10
```

13.6.3 Verify

```
| Test config: gamemoded -t;  
| Start: gamemoderun <app>;  
| Verify: gamemoded -s
```

13.6.4 Notes

Firejail workaround:

```
| firejail --ignore=noroot gamemoderun <app>
```

13.7 Monitoring system performance – MangoHud

<https://wiki.archlinux.org/title/MangoHud>

<https://github.com/flightlessmango/MangoHud>

[mangohud](#) [lib32-mangohud](#)

13.7.1 Configuration (example)

```
cp /usr/share/doc/mangohud/MangoHud.conf.example  
~/.config/MangoHud/MangoHud.conf;
```

```
vim ~/.config/MangoHud/MangoHud.conf
```

> Uncomment:

13.7.1.1 VISUAL

```
# GPU  
gpu_temp  
gpu_core_clock  
gpu_mem_clock  
gpu_power  
gpu_text=<gpu_name>  
gpu_load_change
```

```
# CPU  
cpu_temp  
cpu_power  
cpu_text=<cpu_name>  
cpu_mhz  
cpu_load_change
```

```
core_load  
core_load_change
```

```
# RAM  
vram  
ram
```

```
# Misc (also for screenshots)  
engine_version  
gpu_name  
vulkan_driver  
wine
```

```
histogram  
gamemode
```

13.7.1.2 FYI: INTERACTION

```
# toggle_hud=Shift_R+F12  
# toggle_logging=Shift_L+F2
```

13.7.1.3 LOG

```
output_folder=~/.Games/mangologs  
mkdir -p ~/.Games/mangologs
```


13.7.2 Test configuration

```
mangohud glxgears  
mangohud vkcube
```

13.7.3 Notes

- For some OpenGL apps: mangohud `--dlsym` <app>
- Use with Gamemode: mangohud `gamemoderun` <app>

13.8 Gaming

13.8.1 Steam

<https://wiki.archlinux.org/title/Steam>

[steam](#) [lib32-libnm](#) [wqy-zenhei](#)

13.8.1.1 Enable "Steam Play" for all titles

Steam > Settings > Compatibility > Enable Steam Play > Use current stable Proton version

Note: For games w/ 3rd-party launchers (Ubisoft, Origin, ...) > Use "Proton Experimental"

13.8.1.2 Edit "Launch Options" of <game>

Start w/ Gamemode: `gamemoderun %command%`

13.8.1.3 Controller settings

13.8.1.3.1 Enable Steam Input for your controller

Steam > Settings > Controller:

External Gamepad settings: Enable Steam Input for your controllers

13.8.1.3.2 DualSense Controller (PS5)

https://wiki.archlinux.org/title/Gamepad#PlayStation_4/5_controller

... which is supported by the official Sony Linux driver. You may want to [update the firmware](#).

Disable touchpad acting as mouse:

... to avoid ghost inputs from the touchpad & to use the touchpad as a button in games.

```
| doas vim /etc/udev/rules.d/72-dualsense-disable-touchpad.rules
>
# Disable DualSense controller touchpad acting as mouse
# USB
ATTRS{name}=="Sony Interactive Entertainment DualSense Wireless Controller
Touchpad", ENV{LIBINPUT_IGNORE_DEVICE}="1"
# Bluetooth
ATTRS{name}=="DualSense Wireless Controller Touchpad",
ENV{LIBINPUT_IGNORE_DEVICE}="1"
```

If the game does not recognize/support the controller:

- Change the **game-specific** controller layout to Templates::**Gamepad**
- Change the "**Desktop Layout**" to Templates::**Gamepad**
This is also very useful for gaming outside of Steam (e.g. using lutris). Steam must be running.

13.8.1.4 Workaround: Steam not starting after system upgrade

- [Opt in to the Steam Beta](#)
- Use [the flatpak version](#) or [steam-native-runtime](#)

13.8.1.5 Check game compatibility

> <https://www.protondb.com/>

13.8.2 Ubisoft Connect: Connection fix

Enable MTU probing:

```
| doas vim /etc/sysctl.d/99-sysctl.conf  
> net.ipv4.tcp_mtu_probing = 1
```

13.9 KDE Configuration

13.9.1 System Settings

13.9.1.1 Input & Output

- Mouse > Pointer acceleration: "None"
- Keyboard > Keyboard model: "Generic 105-key PC" (for ISO-DE) | Delay: "250 ms"
- Sound > Profile: [Pro Audio](#), [Analog Stereo](#), [Digital Stereo \(IEC958\)](#)

13.9.1.2 Appearance & Style

- Colors & Themes > You may only want to change: Colors, Window Decoration, Icons, Cursors
- Text & Fonts > Fixed width: "Fira Code 10pt"

13.9.1.3 Apps & Windows

- Default Applications: [Set your default apps](#)
- Window Management > Window Behavior > Titlebar Actions >
 - Mouse wheel: "Maximize/Restore"
 - Middle click: "Close" (Tab behavior)
- Window Management > Window Behavior > Window Actions >
 - Meta + Mouse wheel: "Move to previous/next desktop"
- Window Management > Desktop Effects > Disable "Present Windows" & "Tiling Editor"
- Window Management > Window Rules:
 - Create Application settings: Open app > Alt+F3 > **Configure Special Application settings:**
 - Workaround for Wayland > Add "Position": Apply Initially (like in the X11 session)
 - Workaround for some games > Add "Fullscreen": Force

13.9.1.4 Workspace

- General Behavior >
 - Animation Speed: "Instant"
 - Clicking files or folders: "Opens them" (Clicking next to a file already selects it)
- Search > File Search > Disable "File indexing" (If you don't need it)
- Search > Plasma Search > Disable "Bookmarks, Browser*, Recent Files, Software*, Web*"

13.9.1.5 Language & Time

- Spell Check > Set (Default) Languages & Check "Automatic spell checking"

13.9.1.6 System

- Session > Background Services > **Disable unneeded services like:** Bluetooth, Plasma Browser Integration Installation Reminder, SMB Watcher, Thunderbolt Device Monitor

13.9.2 Default Panel ("Icons-only Task Manager")

Tip: Open/Switch to an app using Meta+[1..9]. Switch between app windows using (Shift+)Alt+Tab.

- **Enter Edit Mode >**
 - **Position:** Left (to save space). Left and Top are typical navigation areas.
 - **Uncheck** "Floating"
 - **Panel Width:** ≥ 68 (to be able to display two tray icons in a row)
- **Configure Icons-only Task Manager > Spacing between icons:** "Small"
- **Configure Digital Clock >**
 - **Appearance > Date format:** "Custom" (ddd d = Weekday Day)
 - **Calendar > Check** "Show week numbers", "Holidays" & **Set** Holiday region

13.9.3 Yakuake (top-down terminal)

Tip: If an application also uses F12 as a key binding, press Shift+F12 to avoid opening Yakuake.

- **Autostart Yakuake:** "Autostart" settings > **Add Application:** Yakuake
- **Configure Yakuake > Behavior:** **Uncheck** "Keep window open..." & "Show system tray icon"
- **Manage Profiles > New:**
 - **Appearance > Font:** **Fira Code 13pt**
 - **General > Semantic Integration > Check** "Mouse click in input line moves cursor"
 - **Set new Profile as Default.** Set this new profile also as the default in Konsole.

13.9.4 Dolphin

- **Configure Dolphin > Split View > Check** "Switch between panes with Tab key"
- **Tip: Open an "Admin tab":** Type "admin:" in the address bar

13.9.5 Spectacle

- **Image Saving >**
 - **Compression Quality:** 99%
 - **Filename:** <yyyy>-<MM>-<dd>_ [JXL](#)

13.10 Virtualization

<https://wiki.archlinux.org/title/Category:Virtualization>

GOAL: Set up KVM/QEMU > libvirt > Virt-Manager for default configuration.

Therefore, libvirt will be running on a *system*-level with [default](#) NAT/DHCP networking.

Note: Insert kernel parameters into e.g. `/etc/cmdline.d/virtualization.conf`

13.10.1 UEFI settings

13.10.1.1 Basic

	AMD	Intel
Virtualization	AMD-V or SVM	VT-x
IOMMU	AMD-Vi	VT-d > Add kernel parameter: <code>intel_iommu=on</code>
	> Add kernel parameter: <code>iommu=pt</code> ...to disable unsupported devices.	

13.10.1.2 AMD only

<https://developer.amd.com/sev/>

Setting	Note
SEV – Secure Encrypted Virtualization	
SEV - Secure Nested Paging (3rd gen)	Should only be supported by Ryzen Pro, Threadripper Pro & EPYC CPUs. https://libvirt.org/kbase/launch_security_sev.html
SEV - Encrypted State (2nd gen)	
SEV (1st gen)	
SME – Secure Memory Encryption	
SME	Disabled by default since Linux 5.15. Set kernel parameter mem_encrypt=on, but problems can occur!
Transparent SME (aka. Memory Guard)	For physical protection. Other memory encryption features (like SEV) are then disabled. Use as a fallback option.

13.10.1.3 Intel only

See: [Intel® Virtualization Technology \(Intel® VT\)](#)

Setting	Note
TDX – Trust Domain Extensions	≥Xeon family: "Sapphire Rapids"
SGX2, SGX – Software Guard Extensions	Even some consumer CPUs are supported.

13.10.2 KVM (Kernel-based Virtual Machine)

<https://wiki.archlinux.org/title/KVM>

Note: "KVM is a [hypervisor](#) built into the Linux kernel. Unlike native [QEMU](#), which uses emulation, KVM is a special operating mode of QEMU that uses CPU extensions ([HVM](#)) for virtualization via a kernel module."

13.10.2.1 Check hardware support

```
| LC_ALL=C.UTF-8 lscpu | grep Virtualization  
> Output: AMD-V or VT-x
```

13.10.2.2 Check if the necessary modules are available in the kernel

```
| zgrep CONFIG_KVM /proc/config.gz  
> Output: CONFIG_KVM AND (CONFIG_KVM_AMD, _INTEL) is set to "y" or "m"
```

13.10.2.3 Check if the kernel modules are automatically loaded

```
| lsmod | grep kvm  
> Output: kvm AND (kvm_amd OR kvm_intel)
```

13.10.3 QEMU (Quick EMUlator and virtualizer)

<https://wiki.archlinux.org/title/QEMU>

OPT: https://wiki.archlinux.org/title/QEMU#Using_an_entire_physical_disk_device_inside_the_VM

For x86_64:

```
| doas pacman -S qemu-desktop;
```

13.10.4 libvirt (virtual machine manager)

<https://wiki.archlinux.org/title/Libvirt>

<https://wiki.libvirt.org/page/FAQ>

Note: "Libvirt now makes storage pools nocow when on btrfs automatically" – [Source](#).

Note: Libvirt installs a zone called 'libvirt' in firewalld and manages its required network rules there.

13.10.4.1 Installation

Default NAT/DHCP networking:

```
| doas pacman -S --needed --asdeps dnsmasq iptables-nft;
```

TPM emulator & DMI system info:

```
| doas pacman -S --needed --asdeps swtpm dmidecode;
```

libvirt:

```
| doas pacman -S libvirt;
```

13.10.4.2 Bypass password prompt

https://wiki.archlinux.org/title/Polkit#Bypass_password_prompt

```
| doas vim /etc/polkit-1/rules.d/49-nopasswd_global.rules
>
/* Allow members of the wheel group to execute the defined actions
 * without password authentication, similar to "sudo NOPASSWD:"
 */
polkit.addRule(function(action, subject) {
    if ((action.id == "org.libvirt.unix.manage") &&
        subject.isInGroup("wheel"))
    {
        return polkit.Result.YES;
    }
});
```

13.10.4.3 Firejail workaround

<https://github.com/netblue30/firejail/issues/5137#issuecomment-1134042482>

13.10.4.3.1 Disable dnsmasq.profile

```
| doas vim /etc/firejail/firecfg.config
> Comment out: dnsmasq
```

13.10.4.3.2 Clean out all system links

```
| doas firecfg --clean;
```

13.10.4.3.3 Re-enable all system links (except dnsmasq.profile)

```
| doas firecfg;
```

13.10.4.4 Enable socket (for QEMU)

```
| doas systemctl enable --now libvirtd.socket;
```


13.10.4.5 Reboot (as an advice)

```
| systemctl reboot
```

13.10.4.6 OPT: Change default storage pool location

Note: Whenever a new domain is created, the ownership should be adjusted: root -> 1000.

Shutdown the domains & type:

```
| doas chown 1000:1000 ~/.local/share/libvirt/images/*;
```

13.10.4.6.1 Do not dynamically change qcow2 file ownership at runtime

... if the virtualization is done by only one user.

```
| doas vim /etc/libvirt/qemu.conf
```

> **Uncomment:**

```
user = "myUsername"  
group = "myUsername"
```

13.10.4.6.2 Destroy & undefine default storage pool "/var/lib/libvirt/images/"

... if the *default* pool has been defined.

```
| doas virsh pool-destroy default;  
doas virsh pool-undefine default;
```

13.10.4.6.3 Create a new default storage pool

```
| mkdir -p ~/.local/share/libvirt/images;  
Disable CoW: chattr +C ~/.local/share/libvirt/images/;  
doas virsh pool-define-as --name default --type dir --target  
~/.local/share/libvirt/images;
```

13.10.4.6.4 Autostart default pool

```
| doas virsh pool-autostart default;  
doas virsh pool-start default;
```

13.10.4.7 Create a storage pool for ISO images

```
| mkdir ~/Downloads/ISO;  
doas chown 1000:libvirt-qemu ~/Downloads/ISO/;  
doas virsh pool-define-as --name ISO --type dir --target ~/Downloads/ISO/;  
doas virsh pool-autostart ISO;  
doas virsh pool-start ISO;
```

13.10.5 Virt-Manager (GUI Client)

<https://wiki.archlinux.org/title/Virt-Manager>

Tip: Use snapshots in Virt-Manager.

[virt-manager](#)

13.10.5.1 Preferences

Edit > **Preferences:**

- General > **Check** "Enable XML editing"
- New VM > **x86 Firmware:** UEFI

13.10.5.2 Enable networking (virbr0)

Right click on "QEMU/KVM" > Details > **Virtual Networks:**

- **Autostart** > **Check** "On Boot"
- **Press icon "► (Start Network)"** (for default network)

13.10.5.3 VM configuration (using QEMU/KVM)

Note: Before finishing the VM creation process > Check "Customize configuration before install":

13.10.5.3.1 Overview (for Secure Boot)

- **Firmware:** UEFI x86_64: .../x64/[OVMF_CODE.secboot.4m.fd](#)

13.10.5.3.2 CPUs (Here: 4C/8T)

- Topology > **Sockets:** 1, **Cores:** 4, **Threads:** 2

13.10.5.3.3 Storage (e.g. VirtIO Disk)

- [Change sector size to 4K \(for 4Kn\)](#) > **Edit XML** of Disk:

```
<disk type='file' device='disk'>
...
  <blockio logical_block_size='4096' physical_block_size='4096' />
</disk>
```

13.10.5.3.4 Enable 3D acceleration (not for Windows)

- **Video > Model:** Virtio & **Check** "3D acceleration"
- **Display > Check** "OpenGL" & **Listen Type:** "None"

13.10.5.4 Enable Secure Boot (after VM creation)

https://wiki.archlinux.org/title/KVM#Secure_Boot

<https://gitlab.com/kralex/virt-firmware>

[virt-firmware](#)

13.10.5.4.1 Enroll default (Microsofts & RedHats) SB keys to a new variable store

```
virt-fw-vars --input /var/lib/libvirt/qemu/nvram/vm-name_VARS.fd --output
/var/lib/libvirt/qemu/nvram/vm-name_VARS.secboot.fd --enroll-redhat --secure-
boot;
```

13.10.5.4.2 Point to the new variable store > Edit XML:

```
<os firmware="efi">
...
  <loader readonly="yes" secure="yes"
type="pflash">/usr/share/edk2/x64/OVMF_CODE.secboot.4m.fd</loader>
  <nvram template="/usr/share/edk2/x64/OVMF_VARS.4m.fd">/var/lib/libvirt/qemu/
nvram/vm-name_VARS.secboot.fd</nvram>
</os>
```

13.10.5.4.3 Verify SB status

... e.g. in the UEFI firmware setup utility.

13.10.6 Windows 11 VM (using virtio, QEMU/KVM)

<https://kevinlocke.name/bits/2021/12/10/windows-11-guest-virtio-libvirt/>

Note: The graphics performance is not great in Windows. [Passthrough a GPU](#) instead.

Note: To hide the "virtualization state", you should not use virtio-win (see below).

13.10.6.1 "Create a new virtual machine"

- Local install media: [win11.iso](#)

13.10.6.2 Storage config

- SATA Disk 1 > Disk bus: VirtIO
- Add Hardware > Storage: Device type: CDROM: virtio-win.iso from [virtio-win](#)^{AUR}
Note: The storage pool for "/var/lib/libvirt/images" must exist.

13.10.6.3 NIC (Network Interface Card)

- Device model: virtio

13.10.6.4 "Begin Installation" (Boot from CD/ISO)

- Press some keys to boot from the ISO because the message may not appear (black screen)
- Choose "Custom: Install Windows only (advanced)"
 - For VirtIO Disk: Load driver > OK > **Select:** \amd64\w11\viostor.inf
 - **Select:** Drive 0 Unallocated Space > Next > ... > Complete installation

13.10.6.5 Install virtio drivers

- Open CD/ISO virtio-win > Execute virtio-win-guest-tools.exe > Reboot

13.11 Podman

<https://wiki.archlinux.org/title/Podman>

... is an alternative to [Docker](#), providing a similar interface, but is more secure by default.

[podman](#)

13.11.1 Registries

```
| doas vim /etc/containers/registries.conf  
> unqualified-search-registries = ["docker.io"]
```

13.11.2 OPT: Implementation of the [Compose Spec](#) [deps]

[podman-compose](#)

13.11.3 Podman Desktop (GUI)

<https://podman-desktop.io/>

```
| flatpak install flathub io.podman\_desktop.PodmanDesktop;  
| flatpak run io.podman_desktop.PodmanDesktop
```

13.12 mpv (media player)

<https://wiki.archlinux.org/title/mpv>

<https://github.com/mpv-player/mpv/wiki>

... is stable, efficient, customizable and extensible via [user scripts](#).

Note: Configuring mpv to your personal requirements *can* be very time-consuming. For an easier start, take a look at my configuration files.

13.12.1 Installation

```
doas pacman -S --needed --asdeps yt-dlp aria2;  
doas pacman -S mpv mpv-mpris;
```

13.12.2 Configuration

- [User settings](#)
~/.config/mpv/mpv.conf
- [Key bindings \(defaults\)](#)
~/.config/mpv/input.conf

13.12.3 My settings

... as shown in my config files like extended key bindings, better audio & subtitles, more QoL, etc.

13.12.3.1 Video scaling (preference)

Reference: [mpv Resampling](#)

> **Experimental** (Hit or Miss): [FSRCNNX](#), [SSimSuperRes](#), [SSimDownscaler](#), [KrigBilateral](#)

> scale=dscale=cscale=**lanczos**

Good alternative upsampling shader: [ravu-zoom-ar-r3](#) (in gather/ or compute/)

13.12.3.2 Frame interpolation (preference)

Reference: [SVP 4 Setup Guide for Smooth “60 FPS” Anime Playback](#) (2019)

> **Setting B:** I Hate Soap Opera Edition

13.13 Hardware Security Key – Nitrokey 3

Features: <https://docs.nitrokey.com/nitrokey3/features>

Example: [Nitrokey 3C NFC](#)

<https://wiki.archlinux.org/title/Nitrokey>

<https://docs.nitrokey.com/nitrokey3/linux/>

Note: The subsequent key generation and storage should take place in a secure environment (preferably without Internet access).

- **Nitrokey:** [python-pynitrokey](#)
- **OPT:** Nitrokey App2
| `flatpak install flathub com.nitrokey.nitrokey-app2`
- **FIDO2:** [pam-u2f](#) [libfido2](#)

13.13.1 Update & Test the firmware

<https://docs.nitrokey.com/nitrokey3/linux/firmware-update>

```
| nitropy nk3 update  
> Touch the device to activate the bootloader  
  
| nitropy nk3 test --pin <FIDO2 pin>
```

13.13.2 OpenPGP smartcard

<https://wiki.archlinux.org/title/OpenPGP-card-tools>

[openpgp-card-tools](#) [kleopatra](#)

```
| doas systemctl enable --now pcscd.socket;
```

13.13.2.1 Setup GnuPG w/ pcscd (PCSC Lite)

[https://wiki.archlinux.org/title/GnuPG#GnuPG_with_pcscd_\(PCSC_Lite\)](https://wiki.archlinux.org/title/GnuPG#GnuPG_with_pcscd_(PCSC_Lite))

```
| mkdir ~/.gnupg;  
| vim ~/.gnupg/scdaemon.conf  
>  
disable-ccid  
pcsc-shared  
  
| gpgconf --reload scdaemon;
```

13.13.2.2 Verify Ed25519/Curve25519 support (for OpenPGP Key Generation)

```
| oct info  
> Verify supported algorithms:  
- SIG: Ed25519 (EdDSA)  
- DEC: Cv25519 (ECDH)  
- AUT: Ed25519 (EdDSA)
```

13.13.2.3 OpenPGP Key Generation With Backup

<https://docs.nitrokey.com/nitrokey3/linux/openpgp-keygen-backup>

13.13.2.3.1 Main Key and Encryption Subkey

A main key with the capability to sign and certify [SC] and a subkey for encryption [E] will be generated.

```
| gpg --full-generate-key --expert
  1. Please select what kind of key you want:
    (9) ECC (sign and encrypt) *default*

  2. Please select which elliptic curve you want:
    (1) Curve 25519 *default*

  3. Please specify how long the key should be valid:
    7y. Note: You can also extend the validity after the expiry date in e.g. Kleopatra.
    > Is this correct? (y/N) y

  4. GnuPG needs to construct a user ID to identify your key:
    > Enter Real name
    > Enter Email address
    > Enter (Skip Comment)
    > (O)kay

  5. A pop-up will appear asking you to enter a secure password to secure the keys.
    > gpg: revocation certificate stored as '~/.gnupg/openpgp-revocs.d/<cert>.rev'
```

13.13.2.3.2 Subkey for Authentication

```
| gpg --edit-key --expert <email>

| gpg> addkey
  1. Please select what kind of key you want:
    (11) ECC (set your own capabilities)

  2. Possible actions for this ECC key: Sign Authenticate
    Current allowed actions: Sign
    1. (S) Toggle the Sign capability ... to remove Sign
    2. (A) Toggle the Authenticate capability
    3. (Q) Finished

  3. Please select which elliptic curve you want:
    (1) Curve 25519 *default*

  4. Please specify how long the key should be valid:
    7y

  5. Is this correct? (y/N) y
    Really create? (y/N) y

  6. gpg> quit
    Save changes? (y/N) y
```


13.13.2.3.3 Backup keys

```
mkdir sec-keys; cd sec-keys;  
gpg --armor --output privkey_<email>.asc --export-secret-key <email>;  
gpg --armor --output subkeys_<email>.asc --export-secret-subkeys <email>;  
gpg --armor --output pubkey_<email>.asc --export <email>;  
gpg --export-ownertrust > <email>.txt;  
gpg --output revoke_<email>.asc --gen-revoke <email>
```

13.13.2.3.4 Import Keys into your OpenPGP card (after key generation)

```
gpg --edit-key --expert <email>  
>
```

Import Main key

```
gpg> keytocard  
1. Really move the primary key? (y/N) y  
Please select where to store the key:  
(1) Signature key  
> Enter (default) Admin PIN: 12345678
```

Import Encryption subkey

```
gpg> key 1 (*Select the Encryption subkey)  
gpg> keytocard  
1. Please select where to store the key:  
(2) Encryption key
```

Import Authentication subkey

```
gpg> key 1 (Deselect the Encryption subkey)  
gpg> key 2 (*Select the Authentication subkey)  
gpg> keytocard  
1. Please select where to store the key:  
(3) Authentication key
```

Quit & Save

```
gpg> quit  
> Save changes? (y/N) y
```

13.13.2.3.5 Import the public key on each system that shall use the OpenPGP card

```
gpg --import pubkey_<email>.asc  
gpg --import-ownertrust <email>.txt
```

And link the keys stored on the smartcard to the local GnuPG-KeyRing:

```
gpg --card-status
```

13.13.2.4FYI: Import the secured Private Key into your OpenPGP card

```
oct admin --card <card id> import privkey_<email>.asc  
> Enter Admin PIN  
> Enter password for signing (sub)key
```

13.13.2.5 Set PINs

Note: Every PIN has a retry counter of 3 attempts. If these attempts are used up for the *Admin PIN*, the *OpenPGP Card* can not be used anymore and must be reset to factory defaults.

Check Remaining PIN attempts:

```
| oct status | grep PIN
```

List card ident (0123:01234567):

```
| oct list -i
```

Set User PIN (for decryption, signing, authentication, update validity, ...):

```
| oct pin --card <card ident> set-user
```

Set Admin PIN (for importing an OpenPGP key, to unblock the User PIN & Reset Code, ...):

```
| oct pin --card <card ident> set-admin
```

Set Reset Code (only to unblock the User PIN):

```
| oct pin --card <card ident> set-reset
```

Later: Unblock the User PIN:

```
| oct pin --card <card ident> reset-user-rc  
ALT: oct pin --card <card ident> reset-user
```

13.13.2.6 OPT: Factory reset the OpenPGP card

```
| oct system factory-reset --card <card ident>
```

13.13.3 FIDO2

https://wiki.archlinux.org/title/Universal_2nd_Factor

13.13.3.1 List device (device = PATH = e.g. /dev/hidraw2)

```
| systemd-cryptenroll --fido2-device=list
```

13.13.3.2 Set PIN

Set PIN:

```
| nitropy fido2 set-pin  
ALT: fido2-token -S <device>
```

Change PIN:

```
| nitropy fido2 change-pin  
ALT: fido2-token -C <device>
```

13.13.3.3 Enable browser integration (for Firejail)

```
| doas sed -i 's/# browser-disable-u2f yes/browser-disable-u2f no/'  
/etc/firejail/firejail.config;  
doas firecfg;
```

13.13.4 SSH Keys

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

13.13.4.1.1 Enable ssh-agent user service

https://wiki.archlinux.org/title/SSH_keys#SSH_agents

```
| systemctl enable --now --user ssh-agent.service;  
  
| doas vim /etc/security/pam_env.conf  
> Append:  
# ssh-agent with systemd user  
SSH_AUTH_SOCK    DEFAULT=${XDG_RUNTIME_DIR}/ssh-agent.socket
```

13.13.4.1.2 Generate a public/private FIDO2 Key Pair w/ User Verification

https://wiki.archlinux.org/title/SSH_keys#FIDO/U2F

Note: The “private key” only refers to the hardware key. Therefore, you can leave the passphrase empty when generating the following key pair.

```
| ssh-keygen -t ed25519-sk -o resident -o verify-required -o  
| application=ssh:nitrokey  
> Enter PIN & Touch token  
> Enter (file name: ~/.ssh/id_ed25519_sk)  
> Enter (no passphrase)
```

13.13.4.1.3 Add the "private key" to ssh-agent's cache

```
| ssh-add ~/.ssh/id_ed25519_sk
```

13.13.4.1.4 Add the "public key" to a remote server (w/ installed OpenSSH)

```
| ssh-copy-id -i ~/.ssh/id_ed25519_sk.pub user@host
```

13.13.4.1.5 Git integration

E.g. GitLab: https://gitlab.com/-/user_settings/ssh_keys

> Add new (public) key: sk-ssh-ed25519@openssh.com ... *myUsername@myHostname*

Test:

```
| ssh -T git@gitlab.com  
> Enter PIN for ED25519-SK key  
> Confirm user presence for key ED25519-SK
```

13.13.5 HMAC-SHA1 challenge-response credential

... e.g. to be able to secure your KeePassXC database.

13.13.5.1 Create a 20 byte secret & Backup this secret

```
| doas pacman -S pwgen;  
| pwgen --capitalize --numerals --secure 19 1 > hmac-sec.txt;
```

13.13.5.2 Convert secret to base32 & Register credential to HMAC slot 2

```
| nitropy nk3 secrets add-challenge-response 2 $(cat hmac-sec.txt | base32)
```

13.13.5.3 Protect your existing KeePassXC database

<https://docs.nitrokey.com/nitrokey3/linux/keepassxc>

Database > Database Security > Add additional protection > Add Challenge-Response > OK

13.14 Windows 11 – Minimal Setup

13.14.1 General

- Enable Secure Boot
- Activate Windows > Update Windows & Apps
- [Enable virtualization-based protection of code integrity](#)

13.14.2 Debloat

- <https://privacy.sexy> > **Standard** settings > Download & Execute script > Reboot
- Uninstall more apps > Debloat more (esp. use gpedit & regedit)

13.14.3 "Package managers"

13.14.3.1 winget

<https://learn.microsoft.com/windows/package-manager/winget/>

... to manage [UWP apps](#) from the [Microsoft Store](#).

Install apps:

... like: firefox kate okular

```
| winget install -s msstore <apps>
```

Upgrade all apps:

```
| winget upgrade -rh
```

13.14.3.2 Chocolatey

<https://community.chocolatey.org>

... to manage [other apps](#).

Installation:

Win + X > A: Execute [command](#)

Install apps:

```
| choco install <apps>
```

Upgrade all apps:

```
| choco upgrade all -y
```

13.15 Waydroid

<https://wiki.archlinux.org/title/Waydroid>

... is a container-based approach to boot a full Android system on a regular GNU/Linux system.

- **REQ for easy setup (2024):** AMD/Intel GPU, Wayland, linux-zen

[waydroid](#)^{AUR}

13.15.1 Init Waydroid

- **Start "Waydroid" > Download the VANILLA or GAPPS version > Done**

13.15.2 Adjust firewalld

<https://wiki.archlinux.org/title/Waydroid#Network>

```
doas firewall-cmd --zone=trusted --add-port=67/udp;
doas firewall-cmd --zone=trusted --add-port=53/udp;
doas firewall-cmd --zone=trusted --add-forward;
doas firewall-cmd --zone=trusted --add-interface=waydroid0;
doas firewall-cmd --runtime-to-permanent;
```

13.15.3 Enable (or only Start) the Waydroid session

```
doas systemctl enable --now waydroid-container.service;
```

13.15.4 Launch GUI

Start "Waydroid" OR:

```
waydroid show-full-ui
```

13.15.5 Google Play Certification (for GAPPS)

<https://docs.waydro.id/faq/google-play-certification>

- **Start Waydroid**

- **Get Android ID:**

```
doas waydroid shell
:/ # ANDROID_RUNTIME_ROOT=/apex/com.android.runtime ANDROID_DATA=/data
ANDROID_TZDATA_ROOT=/apex/com.android.tzdata
ANDROID_I18N_ROOT=/apex/com.android.i18n sqlite3
/data/data/com.google.android.gsf/databases/gservices.db "select * from main
where name = \"android_id\";"
> android_id|<your ID>
```

- **Register ID > <https://www.google.com/android/uncertified>**

- **Restart Waydroid:**

```
waydroid session stop;
waydroid show-full-ui
```

13.16 Rust (using rustup, for devs)

<https://wiki.archlinux.org/title/Rust>

13.16.1 Learn Rust

<https://www.rust-lang.org/learn>

- [A half-hour to learn Rust](#)
- ["The Book"](#)
- [RustRover Academy Course](#)
- [Rust By Example](#)

13.16.2 Install rustup & its stable toolchain

<https://wiki.archlinux.org/title/Rust#Rustup>

```
| doas pacman -S rustup;  
| rustup default stable;
```

13.16.3 Tools: rust-analyzer, Clippy & rustfmt

<https://wiki.archlinux.org/title/Rust#Tools>

```
| rustup component add rust-analyzer rust-src;
```

- Language Server Protocol: [rust-analyzer](#) (rust-src is required)
- Collection of lints: [Clippy](#) (built-in)
- Formatting Rust code according to style guidelines: [rustfmt](#) (built-in)

13.16.4 IDE – JetBrains RustRover

<https://www.jetbrains.com/rust/>

<https://wiki.archlinux.org/title/Rust#Editors>

[rustrover](#)^{AUR} [rustrover-jre](#)^{AUR}

- [Enable Clippy](#)
- [GitLab](#)

13.16.5 Update Rust toolchains

```
| rustup update;
```

13.17 OPT: Easy Effects (for EQ & more)

<https://wiki.archlinux.org/title/PipeWire#EasyEffects>

<https://github.com/wwmm/easyeffects>

[easyeffects](#) [calf](#) [lsp-plugins-lv2](#)

13.17.1 Presets

<https://github.com/wwmm/easyeffects/wiki/Community-presets>

13.17.1.1 Import Input/Output preset

Open Easy Effects > Presets > In/Output > "Import a preset" & Select *preset.json* > **Load** preset

13.17.1.2 Input preset: For male voices w/ noise reduction [2023-07]

Download: https://github.com/jtrv/.cfg/blob/morpheus/.config/easyeffects/input/fifine_male_voice_noise_reduction.json

... from "[Improved Microphone \(Male voices, with Noise Reduction\) EasyEffects preset](#)".

13.17.2 Equalizer (EQ)

... e.g. to change the headphones [frequency response target](#), to [simulate different headphones](#) or to enhance the bass.

Note: To achieve the best audio quality possible, you should use "[bit-perfect playback](#)".

13.17.2.1 Import "EqualizerAPO ParametricEQ" preset

Add Effect: "Equalizer" > Import Preset: [APO](#) > Select: *ParametricEQ.txt*


13.17.2.2 AutoEq (Here: Simulating headphones)

> [Wiki#Can I use AutoEq to simulate a different headphone?](#)

Q: Why simulating headphones instead of using a frequency response target?

> Because the *Harman* targets are quite bass-heavy, and *Diffuse Field* targets are lacking in bass.

Note: As a good starting point, you can simulate the "Sennheiser HD 800 S" (for over-ear headphones) and adjust the bass.

- Go to: <https://autoeq.app>
- Select headphones *to simulate*
- Profiles > Show advanced:
Sound signature > Click on 
- Select *your* headphones
- Select equalizer app: "EqualizerAPO ParametricEq" > Download *ParametricEQ.txt*

14 Disks & Data

Change disk: /dev/**sdb**

Change disk name: **myData** (e.g.: "Vendor name" + "disk size" = e.g. *Seagate20*)

→ dm_name: **myData**

→ mount point: /mnt/**myData** (Change to e.g. "Data" for compatibility with future disks)

14.1 Preparing the disk (as *root*)

> Follow instructions of "Preparing the disk" from "Pre-installation" with /dev/**sdb**

14.2 Setup encrypted internal disk (as *root*)

14.2.1 Create the LUKS encrypted container w/ passphrase

https://wiki.archlinux.org/title/Dm-crypt/Device_encryption#Encryption_options_for_LUKS_mode

```
| cryptsetup luksFormat --label myData /dev/sdb  
> YES  
> fallback-password
```

OPT – Verify sector size:

```
| cryptsetup luksDump /dev/sdb | grep sector
```

14.2.2 Add a keyfile (for crypttab)

https://wiki.archlinux.org/title/Dm-crypt/Device_encryption#Keyfiles

14.2.2.1 Creating a keyfile with random characters

If the "keyfile directory" does not exist yet:

```
| mkdir /etc/cryptsetup-keys.d;  
  
| dd bs=512 count=4 if=/dev/random of=/etc/cryptsetup-keys.d/myData.key  
| iflag=fullblock;  
| chmod 600 /etc/cryptsetup-keys.d/myData.key;
```

14.2.2.2 Add the keyfile to "key-slot 1" of the LUKS header

```
| cryptsetup luksAddKey /dev/sdb /etc/cryptsetup-keys.d/myData.key  
> fallback-password
```

14.2.3 Open the LUKS container

```
| cryptsetup open /dev/sdb myData --key-file /etc/cryptsetup-keys.d/myData.key;
```

14.2.4 Format the (unlocked LUKS) device

```
| mkfs.btrfs -L myData /dev/mapper/myData;
```

14.2.5 OPT: Choose a suitable zstd compression level (for mount)

14.2.5.1 Get test-package (Here: pacman package)

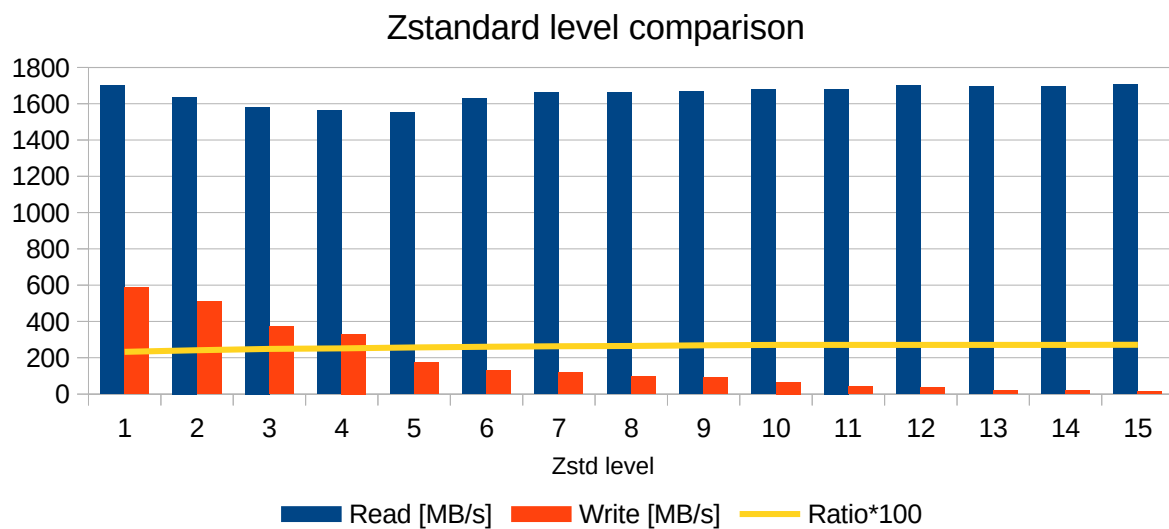
```
cp /var/cache/pacman/pkg/<package>.pkg.tar.zst /tmp/ && cd /tmp/;  
unzstd <package>.pkg.tar.zst
```

14.2.5.2 Benchmark

```
for j in {1..15}; do zstd -b$j -T0 <package>.pkg.tar; done
```

14.2.5.3 Example

Package: libreoffice-still-7.3.6-4-x86_64.pkg.tar | Zstd version: v1.5.2



Zstd level	Read [MB/s]	Write [MB/s]	Ratio*100
1	1702	586	233
2	1637	510	242
3	1582	374	249
4	1565	331	252
5	1552	174	257
6	1630	128	261
7	1661	118	264
8	1662	96	265
9	1666	91	269
10	1677	66	271
11	1681	42	271
12	1701	36	271
13	1693	21	271
14	1696	19	271
15	1706	15	272

14.2.6 Get mount options (for fstab)

14.2.6.1 Mount the btrfs volume

READ: https://wiki.archlinux.org/title/Security#Mount_options

OPT – Increase [compression](#) level (≥ 4), change: `compress=zstd:4`

```
| mount -m -o compress=zstd,nosuid,nodev,noexec /dev/mapper/myData /mnt/myData
```

14.2.6.2 Note mount options (for fstab)

```
| mount | grep myData
> ... btrfs (<mount_options>)
```

14.2.7 Change owner to "user:primary_usergroup"

Note: The first normal user has the numerical User ID "1000" (`id 1000 <=> id myUsername`)

```
| chown -Rc $(id -un 1000): /mnt/myData;
```

14.2.8 Automount keyfile encrypted disk

Note: With the following config, the system can only boot when the disk is attached.

14.2.8.1 Crypttab

https://wiki.archlinux.org/title/Dm-crypt/System_configuration#crypttab

Note: *<Output of <command> in vim (normal mode) using ":r! <command>">*

```
| vim /etc/crypttab
>
myData    UUID=<blkid -o value -s UUID /dev/sdb>          /etc/cryptsetup-
keys.d/myData.key
```

14.2.8.2 Fstab

<https://wiki.archlinux.org/title/Fstab>

Note – for [Btrfs](#): `fs_passno=0` instead of `2`.

```
| vim /etc/fstab
>
# myData
/dev/mapper/myData          /mnt/myData          btrfs    <mount_options>    0 0
```

14.2.8.3 Test crypttab & fstab

```
| umount /mnt/myData;
| cryptsetup close myData;
| systemctl daemon-reload;
| systemctl start systemd-cryptsetup@myData.service;
| mount -a
```

```
| OPT: systemctl reboot
```

14.3 TODO – OPT: Directory structure of your private data

... for faster navigation.

- COLL/ – Collections (no music/video coll.)
- DWNLD/ – Downloads
- IMP/ – Important stuff
- MUSIC/ – Music coll.
- VID/ – Video coll.

14.4 TODO: Create a backup plan

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

Recommendations:

- **BorgBackup:** [borg](#)
- **restic:** [restic](#)

14.4.1 Demands

- **Type of backup medium:** External hard drive
- **Planned frequency of backups:** Weekly, Monthly
- **Important:** Encrypted & Compressed storage; Handles FS metadata & renames
- **Nice2Have:** Resumable, Fast (Delta transfer, Multithreaded), Snapshots
- **What will be backed up:**
 - Selected disks mounted on `/mnt/`
 - Selected directories & files within `/home/`

14.4.2 Backup all LUKS header to another encrypted drive

https://wiki.archlinux.org/title/Dm-crypt/Device_encryption#Backup_and_restore

```
doas cryptsetup luksHeaderBackup /dev/<device> --header-backup-file  
/mnt/BACKUP/Linux/keys/luksHeader_root.img
```

15 Maintenance

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

15.1 Removing unused packages

[https://wiki.archlinux.org/title/Pacman/Tips_and_tricks#Removing_unused_packages_\(orphans\)](https://wiki.archlinux.org/title/Pacman/Tips_and_tricks#Removing_unused_packages_(orphans))

15.1.1 Remove unneeded dependencies (using paru)

```
| paru -c
```

15.1.2 Check if you need certain packages

```
| pacman -Qtdq  
> Yes: doas pacman -D --asexplicit <packages>
```

15.1.3 Recursively removing orphans and their configuration files

```
| pacman -Qtdq | doas pacman -Rns -
```

15.1.4 Detect & Remove more unneeded packages

https://wiki.archlinux.org/title/Pacman/Tips_and_tricks#Detecting_more_unneeded_packages

```
| doas pacman -Qqd | pacman -Rsu --print -  
> Remove: doas pacman -Rns <packages>
```

15.2 Pacnew and Pacsave

READ: https://wiki.archlinux.org/title/Pacman/Pacnew_and_Pacsave

15.2.1 Managing .pac* files – pacdiff (using vimdiff)

```
| doas pacdiff  
> Repeat: View / Skip / Remove pac* file
```

- **Change window** to your current config file: Ctrl+w > w
- **Repeat:**
 - **Jump to the next change:**] > c
 - **OPT: Get changes** from other into current window: d > o (diff obtain)
- **Save & Quit your config file:** :x!
- **Quit pac* file:** :q

15.3 Free up disk space

15.3.1 Tools (GUI)

https://wiki.archlinux.org/title/List_of_applications/Utilities#Disk_cleaning

- KDE disk usage statistics: [filelight](#)
- KDE system cleaning: [sweeper](#)
- Duplicate finder: [rmlint-shredder](#) (CLI: [rmlint](#))

15.3.2 Remove all cached versions of uninstalled packages

```
| doas paccache -ruk0;
```

15.3.3 Clean ~/cache/

```
| rm -rf ~/.cache/*;
```

15.3.4 Clean cache from other applications

Note: Not fully tested.

```
| rm -rf ~/{.bundle,.cargo,.cmake,.dotnet,.electron,.electron-  
gyp,.gem,.gradle,.lazarus,.node-  
gyp,.npm,.nuget,.nvm,.racket,.rustup,.stack,.yarn} || true;
```

15.3.5 Tip – Print est. file space usage (using du)

Files sorted by size "MiB" from ~/cache/ with depth \leq 1:

```
| du -md1 ~/.cache/ | sort -n;
```

15.4 Btrfs

15.4.1 Verify structural integrity

<https://btrfs.readthedocs.io/en/latest/btrfs-check.html>

Note: `--force` for mounted & quiescent filesystems.

```
| doas btrfs check -p --force /dev/mapper/root
```

> **Errors found?**

> Backup data &

> Attempt to repair unmounted FS with `--repair` ELSE reformat.

> **No error found? > Verify checksums of data blocks & wait:**

```
| doas btrfs check --check-data-csum -p --force /dev/mapper/root
```

15.4.2 Making block group layout more compact

<https://btrfs.readthedocs.io/en/latest/btrfs-balance.html#making-block-group-layout-more-compact>

The Btrfs chunks are usually not filled with only used data, see:

15.4.2.1 List allocation of block group types of the mounted (root) fs

```
| btrfs filesystem df /
```

> **Example:**

Data, single: **total**=1.59TiB, **used**=1.42TiB

System, DUP: **total**=8.00MiB, **used**=208.00KiB

Metadata, DUP: **total**=30.00GiB, **used**=3.15GiB

GlobalReserve, single: **total**=512.00MiB, **used**=0.00B

15.4.2.2 Calculate used/total ratio

... to check if balancing is worthwhile.

Here: The **data** chunks of the **root** mount point are filled by 91% on average, and **metadata** chunks by 10.5%.

15.4.2.3 Start balancing w/ filters for a faster process

Note: Only use `-usage=70` if the metadata used/total ratio is really off (like in the example).

Tip: Set `-usage=90` for SSDs.

Note: If you get an "Error: No space left on device (ENOSPC)", use `-usage=0`.

```
| doas btrfs balance start -usage=85 -usage=70 --bg /
```

Check balancing status:

```
| doas btrfs balance status /
```

Result – Here: w/ -usage=90 -usage=70 after 8 minutes:

Data, single: **total**=1.44TiB, **used**=1.42TiB

System, DUP: **total**=32.00MiB, **used**=208.00KiB

Metadata, DUP: **total**=4.00GiB, **used**=3.13GiB

GlobalReserve, single: **total**=512.00MiB, **used**=0.00B

16 Troubleshooting

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

Note: You can always restore from a btrfs snapshot.

16.1 Downgrading packages

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

```
doas pacman -U /var/cache/pacman/pkg/<old_pkg>.pkg.tar.zst  
doas pacman -U ~/.cache/paru/clone/<aur-pkg>/<old_aur-pkg>.pkg.tar.zst
```

16.2 Check for errors (also for maintenance)

https://wiki.archlinux.org/title/System_maintenance#Check_for_errors

16.2.1 Logs

16.2.1.1 *Error, critical & alert priority messages from last boot*

```
journalctl -p3 -b-1
```

16.2.1.2 *Failed systemd services*

```
systemctl --failed
```

16.2.1.3 *Pacman*

```
less /var/log/pacman.log
```

16.2.1.4 *KDE GUI*

```
kssystemlog
```

16.2.2 Analyze boot times

16.2.2.1 *Basic startup time*

```
systemd-analyze time
```

16.2.2.2 *List of all running units, ordered by initialization time*

Note: Output might be misleading as the initialization of one service might be slow simply because it waits for the initialization of another service to complete.

```
systemd-analyze blame
```

16.3 Useful keyboard shortcuts

16.3.1 Switch to n-th virtual console

... e.g. to avoid hard shutdowns (holding down the power button).

> **Switch:** `Ctrl + Alt + Fn + { F1, F2, ..., FN }`

> **Login** as *myUsername* (or as *root*)

16.3.2 Kernel (SysRq)

[https://wiki.archlinux.org/title/Keyboard_shortcuts#Kernel_\(SysRq\)](https://wiki.archlinux.org/title/Keyboard_shortcuts#Kernel_(SysRq))

... e.g. to avoid hard shutdowns, or to kill a memory-hogging process.

Note: The key [SysRq] is often [Fn]+[Print].

16.3.2.1 Enabling

```
| doas vim /etc/sysctl.d/99-sysctl.conf  
> kernel.sysrq = 128
```

16.3.2.2 Rebooting

"Reboot Even If System Utterly Broken":

`Alt + Fn + Print + { R > E > I > S > U > B }`

16.3.2.3 Killing a memory-hogging process

`Alt + Fn + Print + F`

16.4 Common problems

16.4.1 Pacman: "invalid or corrupted package PGP signature"

https://wiki.archlinux.org/title/Pacman/Package_signing#Upgrade_system_regularly

... if the system upgrade has been delayed for an extended period of time.

```
| doas pacman -Sy archlinux-keyring && doas pacman -Su;
```

16.4.2 Audio problems > Try:

```
| rm -rf ~/.local/state/wireplumber/;
```

```
| systemctl restart --user pipewire-pulse.socket pipewire-pulse.service  
| pipewire.socket pipewire.service wireplumber.service
```

16.4.3 KDE upgrade problems > Clean cache

https://wiki.archlinux.org/title/KDE#Clean_cache_to_resolve_upgrade_problems

```
| rm ~/.config/Trolltech.conf;  
| kbuilder6 --noincremental;
```

```
| OPT: rm -rf ~/.cache/*;
```

17 TODOs

17.1 Additions

- [Hardening of mount points](#)
- More [Trusted Platform Module \(TPM\)](#) integration
- Better [btrfs checksumming](#) for collision resistance
- Maintenance: [Old configuration files](#), [Broken symlinks](#), [List packages by date](#)
- Maybe: Restrict unprivileged user namespaces, but selectively allow for some apps (e.g. Steam) with AppArmor's [unprivileged_userns_restriction](#) > Similar to firejails restrict-namespaces

17.2 Deprecations

- Multilib repo (32-bit)
Waiting for Wine & Steam
- [Xorg](#)
Currently, no or incomplete implementation of Wayland in: [KDE](#), Wine, Steam, OpenJDK, ...

17.3 Current bugs & misbehaviors

- **SDDM**: The latency may decrease (or increase) after you have logged out and logged back in to KDE Plasma.
- **systemd**: A message "You are in emergency mode" will appear two minutes after you have not yet entered your crypt-password. Simply ignore this warning by pressing Enter.
This is caused by [GPT partition automounting](#): "Timed out waiting for device /dev/gpt-auto-root."
- **mkinitcpio**: [Emergency shell is not fully disabled](#)
Since the root account in initramfs is locked because of mkinitcpio's systemd hook, there is no working emergency shell anyway.
- **Kernel lockdown** disallows hibernation/suspend despite having encrypted swap.
- **GameMode** does not automatically set the GPU power profile to "3D FULL SCREEN".
- **Steam** does not have GameMode integration. You have to use "gamemoderun %command%".
- **Current typical error messages**:
systemd-cryptsetup: No valid TPM2 token data found.
dbus-broker-launch: Activation request for 'org.freedesktop.nm_dispatcher' failed.

17.4 Changelog

- **DualSense Controller (PS5)**
 - **New:** Disable touchpad acting as mouse
... to avoid ghost inputs from the touchpad & to use the touchpad as a button in games.
- **PipeWire**
 - **Update config files:**
~/.config/pipewire/pipewire.conf.d/10-rates.conf & ~/.config/pipewire/pipewire.conf.d/10-quantum.conf
- The **GPU power profile** "3D FULL SCREEN" is now the default in Arch Linux, so
rocm-smi --setprofile "3D FULL SCREEN" is not needed anymore (for amdgpu).