

ARCH LINUX

installation recommendations
guides
cheat sheets

1. General Installation Overview (<10 GB)

PACKAGES:

Filesystem:	mc	sdl2_mixer
dosfstools	7zip	xorg-server
exfatprogs	Boot:	xorg-init
exfat-utils	grub	xorg-xrandr
ntfs-3g	efibootmgr	i3-wm
e2fsprogs	System:	dmenu
Suite:	gcc	qterminal
iwd	sdl	Utility:
nano	sdl2_gfx	alsa-utils
	sdl2_image	feh
		:PACKAGES

ADDITIONAL:

Suite:	vlc	qt6ct
kate	vlc-plugin-ffmpeg	Utility:
dolphin	qmp	featherpad
libreoffice	evince	
gimp	konqueror	
gwenview	Backend:	
	lxappearance	:ADDITIONAL

AUR installation (trizen):

- make sure locale is setup properly:

```
→ nano /etc/locale.gen
→ uncomment your locale (UTF-8)
[i.e.: I use de_DE.UTF-8 and
en_US.UTF-8 and then call locale-gen
(as root) //if the locale isn't
installed, some applications spit out
ugly warnings.
```

- install:

```
→ base-devel
→ git
```

- add meta to wheel:

```
→ nano /etc/sudoers
→ uncomment the line '%wheel
ALL=(ALL:ALL) ALL'
→ run 'usermod -aG wheel meta'
- download and install trizen:
→ run 'git clone
https://aur.archlinux.org/trizen.git &&
cd trizen && makepkg -si'
→ enter root ('su')
→ download reported packages
→ !!!exit root!!!
→ run 'git clone ...' again
```

:(trizen) installation **AUR**

COMMANDS:

user management:

add user: `useradd (-m) name // -m to create homedir`
remove user: `userdel -r username`
rename user: `usermod -l newname oldname`
user detail: `id user`
list groups: `cat /etc/group`
add user to group: `usermod -aG groups username // gpasswd -a user group`
remove user frm grp: `gpasswd -d user group`
add group: `groupadd name`
change group: `groupmod -n newname oldname`
remove group: `groupdel group`

see also: `/etc/skel/` and `/etc/default/useradd`

Getting started:

when uncertain what option to use, type command `--help`

show directory: `ls (path) (ls -l, ls -lA)`
`dir, DIR (when initialized)`

change directory: `cd`
`.. =up,`
`/ =root,`
`./ =current,`
`~/ =home`

NOTE: `\` is the 'escape' character
denoting the next character to be
literal (i.e. `~/this\ there.txt`
(to have spaces in filenames))

copy file `cp (options) file folder/(name)`
move/rename file `mv`
remove file `rm`
edit file `nano (^=ctrl, M=alt, ctrl+o=save)`
browser `mc (access menu with F9)`

pacman:

install: `pacman -S`
search (install): `pacman -Ss`
remove: `pacman -R`
remove(+): `pacman -Rs`
clean orphans: `pacman -Qdtq | pacman -Rns -`
clear cache: `pacman -Scc (or: trizen -Sc)`
list all: `pacman -Q`
package status: `pacman -Qi <package>`
search (installed) `pacman -Qs <pattern>`
list explicit: `pacman -Qet`

tar archive:

store: `tar -cpvaf [file] [source]`
restore: `tar -xpvaf [file]`

7zip:

pack: `7z a [archive] [source]`
unpack: `7z x [archive] (-o dirname)`

other:

active services: `systemctl --type=service`
disk space: `df (option -i for inodes)`
fix ntfs: `ntfsfix -d /dev/sdXY`

runtime

show processes: `top`
`ps -aux`
`pgrep -l -u <user>`
terminate: `killall (options) name`
ask confirm: `-i`
force kill: `-s SIGKILL`

desktop:

MIME types

get MIME `xdg-mime query filetype [file]`
query default app `xdg-mime query default [MIME]`
set default app `xdg-mime default [x.desktop] [MIME]`

2. Setups (Cheat Sheet)

fstab:

if you want to restrict ntfs partitions you need to mount partition with option ntfs-3g instead of just ntfs.

```
uid=XXXX      user id
gid=XXXX      group id
umask/dmask/fmask=UGO
  0          rwx
  1          rw-
  2          r-x
  3          r--
  4          -wx
  5          -w-
  6          --x
  7          ---
```

(the rw tag can be removed, assuming you're working off a generated fstab.)

find UUID of partitions: `lsblk -f`

iwd:

```
/etc/iwd/main.conf
[General]
EnableNetworkConfiguration=true
[Network]
NameResolvingService=systemd
[Scan]
DisablePeriodicScan=true

/var/lib/iwd/[network].psk
[Settings]
AutoConnect=false
```

xorg (screens)

```
/etc/X11/xorg.conf.d/*
10-monitor.conf:
Section "Monitor"
    Identifier "something"
    Option "Primary" "true"
    Option "PreferredMode" "800x600"
EndSection
Section "Monitor"
    Identifier "something2"
    Option "RightOf" "something"
    Option "PreferredMode" "800x600"
EndSection

#my laptop doesn't wake up from
#hibernation (also on Windows)
Section "Extensions"
    Options "DPMS" "false"
EndSection

Section "ServerFlags"
    Option "BlankTime" "0"
```

```
# Option "StandbyTime" "10"
# Option "SuspendTime" "20"
# Option "OffTime" "30"
EndSection
```

fonts:

useful recommendations I found on a website:

```
ttf-hanazono
ttf-liberation
adobe-source-
    code-pro-fonts
    han-sans-otc-fonts
    han-serif-otc-fonts
    sans-fonts
    serif-fonts
```

This is a **vibes** based recommendation. For perfect clarity: if you want to minimize bloat or avoid any fancy shmancy additions you may want to install xterm instead of qterminal. Same for dolphin/konqueror. The issue: gnome programs are usually cloned to death. Which one now is "the shit" – is probably ... a question for the experts. Thunar and Nemo for instance are practically identical. TLDR: I don't claim that these are the best options.

I believe: 'falkon' is a more lightweight **konqueror** (* as **web-browser** option). 'fiery' is ... weird. It has potential but the developers gave up like halfway through. [shrugs] If you don't want to be a cheapskate (whether that's good or not?), **vivaldi** is available.

GENERALLY: You may do fine without the KDE apps (Dolphin, Gwenview, Okular) and instead opt for **cinnamon** or gnome and lean into the GTK ecosystem (**Evince** instead of **Okular**, ~~Nemo instead of Dolphin~~). The thing is, if you use vivaldi, you won't need either (**PDF viewer**). Initial download sizes may however be deceptive. Trying to remove either Okular or Evince, for instance, from my install amounts to roughly the same. [shrugs again] So, this is still a somewhat curated list. At some point Fiery was a recommendation – as having X many programs that roughly do the same makes me ... explain this to myself until I get it, or so. What's worth a Megabyte?

I will say however that if you're coding and you're not, in this case: '**trizen -S sublime-text-4**', you're missing out. You'll need a tutorial to understand why! (You can bulk select and edit multiple lines at once)

Note that using pacman and trizen creates files on your computer you may not need. If you're running out of space, try clearing cache and removing orphaned packages. **You however don't need an AUR helper or the AUR at all to install Sublime. Currently I don't have an AUR helper installed.**

=====

=====

WORKFLOW :: Setting up Storage ::

A brief history:

Originally Hard Drive partitions (MBR Partition Table) were a lot more limited by available space. Generally 4 Primary partitions were possible, while subsequently 'extended partitions' were made possible (making more out of one). Nowadays it's a lot less restrictive. Obviously.

2: Calculate Entry Points and setup Volumes

Example: /efi and /boot

1. Calculate Entry Point for Boot (1*SPGB)
2. Create new Partition (first available)
3. Set End of Partition as 'Entry point for Boot' and subtract 1.
4. ~Add volume to entry point, subtract 1.~

Example: /access/cmain and /access/volume

1. Subtract 4*SPGB from TotalSectors
2. Create new Partition (to end)

/efi	1 GB	EMPTY	
/boot	1 GB		2 GB

This volume sits at the start of the main drive, thought of as a singular partition. Not sure if this is enough. The recommended installation however fits.

COMMANDROOT
(root+home) < GB

/access/volume
You can create /access/public anywhere and link it into /access/
REST GB

1: Get an idea of what you want

'fdisk /dev/sdX', print partition table, write down: sector size and total amount of sectors.

$1024 / \text{sector_size} \Rightarrow \text{n_sectors_KB}$
 $1024 * \text{n_sectors_KB} \Rightarrow \text{n_sectors_MB}$
 $1024 * \text{n_sectors_MB} \Rightarrow \text{n_sectors_GB}$

(sector_size=512 \Rightarrow **2097152** sectors per GB)

3. Assuming there are no gaps, the remaining space can be turned into Volume.

3: Determine Filesystems

For full compatibility it is presumably required to also set the intended filesystem tags.

4: Write partition table and exit

This step is final. What's overwritten is overwritten.

> create filesystems

/	10 GB
/home	2 GB

/access/cmain	16 GB
swap 4 GB	

Setting up Storage :: **WORKFLOW**

=====

Though I certainly fancy me some esoteric geometry, the practical interface is in the letters and words. Whatever arrangement you'll end up with, will over time surely sink in and become second nature for a while. Especially at the start however you may find it helpful to have some anchor or pointers. Hence I recommend maintaining a journal of sorts:

A) A numbered list of your partition with personal identifiers, volume, mount points and filesystem.

SDA	ID	POSITION	SIZE	FILE-SYSTEM	Mount Point
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					

B) A list of modifications you added to your system that you may later want to fall back upon for reasons unbeknownst to blorb.

Add	Remove
<pre>vivaldi obs-studio obsidian</pre>	

Replace	Change
vivaldi > kde-pim-meta	

C) Frequently used commands and other useful strings:

```
efi > mkfs.fat -F 32 /dev/sdxY
fstab> genfstab (-U/-L) <root-node> <path/to/fstab>
grub > grub-mkconfig -o /boot/grub/grub.cfg
```

SYSTEM BACKUP

WORKFLOW:BACKUP

Generally there are two modes: Store and Restore. A tiny quirk however makes it so, that one is significantly more important than the other. On our system, the system backup exists as a restoration point to return to, while we only do have limited space to back our system up into. This means that we rather not back our system up “ever so often” – **unless we first restore it its former point**, then bring it back to where we want to back it up, and then back it up. To that end: Restoring the backup is part of making a Backup. This is here called **‘consolidation’**. Outside of that, you’d only ever want to restore it. Or maybe you wouldn’t. That’s the thing.

Subsequently we can talk of:

Primary Consolidation	>	Full System Backup
Auxiliary Consolidation	>	Partial Backup
Minor Consolidation	>	Non-integrated Backup

A – Create Rescue Package

Hence the first step to a backup is to create a rescue package. This is to ensure that all significant changes to the system can be implemented easily before making the backup. To this point Dolphin is certainly awesome, but **mc** is pretty much perfect minus the convenience of a high-res mouse interface.

1. **Create a folder**, presumably in meta, possibly named **‘rescue_package’**.
2. In there, create folders regarding the various units (i.e. meta, root) and fill them.

Consider:

- setup files
- scripts
- resources

NOTE: While assembling the rescue_package, you can also **consolidate** (build a “proto-root” or prepare for it). To avoid making silly mistakes however, try not to mount the command root during the process.

NOTE: You may also want to want to use **Konqueror** to analyze the memory use of the filesystems you intend to backup. > **After opening Konqueror, click “Home Folder” and navigate to the folder you want to “see”. There select ‘View > View Mode > File Size View’**. This way I can see that the vast majority of my home folder is used by vivaldi configuration (170 MB for one user) or that 10 MB go to thumbnails in a 40 MB cache. And you definitely can delete .cache folders before backing things up. Similarly you may want to get rid of clutter (unneeded configs) you’ve accumulated.

NOTE: If you've been installing themes or icons, it is a good idea – generally – to copy those you like into your `/usr/share/icons` or `themes` folder respectively and delete the ones you don't need from your local `.themes` and `.icons` folders. Sublime-text color schemes go into `/opt/sublime-text/Packages/User` (you may have to create the User folder).

B – Boot into Backup Environment

For all intents and purposes the basic Arch Linux live environment does just fine. Alternatively we could create a separate super minimalistic installation with just the basic necessities.

1. Boot into Backup Environment (Flash Drive/Backdoor Install)

```
cd / (to not get confused later)
mkdir /cmdroot
mount <partition> /cmdroot
mount <root> /mnt
```

2. Copy rescue package to /cmdroot

```
cp -rp /mnt/home/meta/rescue_package /cmdroot/rescue.smthng
```

3. Restore Backup

NOTE: if you're using labels in your `fstab`, it is probably easier to just create new filesystems. **If you're using UUIDs in your `fstab`, you'll have to regenerate the `fstab`** (and update the `/boot/grub/grub.cfg`) when formatting the filesystems.

```
mount -mkdir /dev/sdXY /efi
mount /dev/sdXY /boot
mount /dev/sdXY /home
```

```
tar -xpvf /cmdroot/backup/efi.backup.tar.zst
tar -xpvf /cmdroot/backup/boot.backup.tar.zst
tar -xpvf /cmdroot/backup/home.backup.tar.zst
```

```
umount /mnt
dd if=/cmdroot/backup/SYSTEM.IMG of=/dev/sdXY bs=64M
```

4. Reboot and update Your System

```
opt > update (pacman -Syu) if you plan on installing new packages
opt > integrate rescue package / copy and chown -R to meta
opt > ...
    > (reboot), check functionality
```

5. Boot into Backup Environment and make backup

```
dd if=/dev/sdXY of=/cmdroot/backup/SYSTEM.IMG
mount -mkdir /dev/sdXY /efi
mount /dev/sdXY /boot
mount /dev/sdXY /home
tar -cpvf /cmdroot/backup/efi.backup.tar.zst /efi
tar -cpvf /cmdroot/backup/boot.backup.tar.zst /boot
```

```
tar -cpva /cmdroot/backup/home.backup.tar.zst /home
```

BACKUP: **WORKFLOW**

=====

CUSTOMIZATION:example:

bootsplash:

```
/etc/default/grub:  
(find line): GRUB_BACKGROUND and edit file  
update /boot/grub/grub.cfg
```

consistent themes:

The solution on this setup is this:

installing **lxappearance** to change the theme works for the most part.

To fix the horrid alternating lines in dolphin, the only solution that worked for me is to use a (dolphin) color theme that works with them. qt6ct can alter Konqueror icons. Using qt6ct, add to

```
/etc/environment:  
QT_QPA_PLATFORMTHEME=qt6ct
```

bash:

macros:

```
alias dir='ls --color=auto -l'  
alias DIR='ls --color=auto -lA'  
alias edit='featherpad'  
alias wconnect='iwctl station wlan0 connect name'  
alias wdiscon='iwctl station wlan0 disconnect'  
alias wshow='iwctl station wlan0 show'  
(if periodic scan is disabled, you first have to 'iwctl station wlan0 scan' for networks)
```

prompt:

```
PS1='XXX' where XXX =  
  [ needs to be written as \[  
  \033[Y;ZZm   Y: 0=normal, 1=bold/bright. ZZ: ANSI color code  
  \033[00m     defaults  
  \u           user  
  \h           host  
  \w           directory  
example:      PS1='[\033[01;33m\u\033[00m]:\033[0;37m\w\033[00m:'
```

dircolors:

because we're using windows filesystems, there's the issue that some color combinations (ls) are unreadable depending on colors chosen. I solved this the following way:

1. dircolors -p > .dircol (/etc/dircol)
2. nano (/etc/)dircol
3. find line: OTHER_WRITABLE
4. change colors (30;43 = black text, yellow background)
5. in **.xinitrc** (> 'loaded per X session') add: 'eval "\$\$(dircolors -b (/etc/)dircol)'"

i3:

.config/i3/config:

+simple wallpaper

```
exec_always --no-startup-id feh -bg-scale /file/screen1.png /file/screen2.png
```

+dpi scaling

```
exec --no-startup-id xrandr --output HDMI-1 --scale 1.2x1.2
```

+lock screen

```
bindsym Control+Shift+BackSpace exec "i3lock -i /access/public/meta/gfx/lockscreen.png"
```

+restricted popup window size

```
floating_minimum_size 75 x 50
```

```
floating_maximum_size 1200x670
```

:example: **CUSTOMIZATION**