

Sponsored by



UpCloud

[Click here](#)

to get **\$25** and **Cloud Server**

How to clean Arch Linux

Every system becomes cluttered sooner or later and Arch Linux is not the exception. So, in this post, you will learn how to clean your Arch Linux system.

INTRODUCTION

Although Arch Linux takes little of disk space right after the installation, as the time passes it grows quite a lot. So, if you do not have any free space left on your computer or you just would like to keep your Arch Linux system clean, this post is all you need.

In this post, you will learn how to:

1. Clean package cache
2. Remove unused packages (orphans)
3. Clean the cache in your /home directory
4. Remove old config files
5. Remove duplicates, empty files, empty directories and broken symlinks
6. Find the largest files and directories
7. Disk cleaning programs that can do most of the steps above automatically
8. Clean Systemd journal

*NOTE that I would like to warn you that you may damage your system if you do a mistake during these procedures. So, please **back up all your files** before doing anything to your system.*

VIDEO





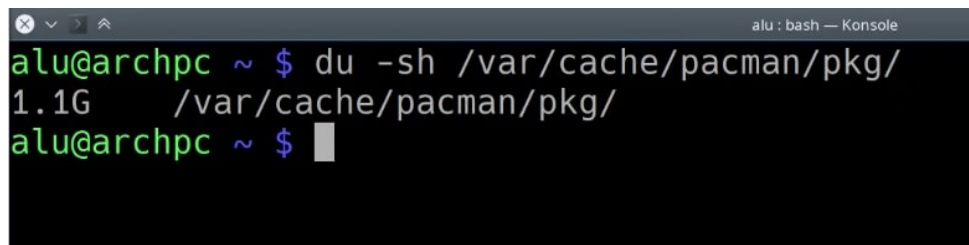
Steps to Clean Arch Linux

1. Clean package cache

Pacman, a package manager of Arch Linux, stores all downloaded packages in `/var/cache/pacman/pkg/` and it does not remove the old or uninstalled versions automatically. You might think this is a mistake, but this is done deliberately. This allows downgrading a package without the need to retrieve the previous version through the Arch Linux Archive. Or if you uninstall a program, you can easily reinstall it without a new download. If you have a slow internet connection, this may be useful. For example, you can simply install a package from this directory using the command below.

```
sudo pacman -U /var/cache/pacman/pkg/packageName
```

However, this `/var/cache/pacman/pkg/` folder can grow indefinitely in size.

A terminal window titled 'alu : bash — Konsole' showing a command and its output. The command is 'du -sh /var/cache/pacman/pkg/' and the output is '1.1G /var/cache/pacman/pkg/'. The prompt is 'alu@archpc ~ \$'.

```
alu@archpc ~ $ du -sh /var/cache/pacman/pkg/
1.1G /var/cache/pacman/pkg/
alu@archpc ~ $
```

Showing the size of downloaded packages in my Arch Linux

So, you need to clean it from time to time. There are two ways you can do that: **manually** and **automatically**.

Cleaning the cache manually

You can clean the cache manually. For example, I usually move these files to my old hard drive that I use only to store data. This way I can always access these files but they do not take valuable space on my system.

However, if you do not have extra space to store these packages, you can remove them without a backup. One option is to remove cached packages that are not currently installed:

```
sudo pacman -Sc
```

The other option is to remove all the package from the cache, including those that are installed:

```
sudo pacman -Scc
```

```
alu@archpc ~ $ sudo pacman -Sc
[sudo] password for alu:
Packages to keep:
  All locally installed packages

Cache directory: /var/cache/pacman/pkg/
:: Do you want to remove all other packages from cache? [Y/n] y
removing old packages from cache...

Database directory: /var/lib/pacman/
:: Do you want to remove unused repositories? [Y/n] y
removing unused sync repositories...
alu@archpc ~ $ sudo pacman -Scc

Cache directory: /var/cache/pacman/pkg/
:: Do you want to remove ALL files from cache? [y/N] y
removing all files from cache...

Database directory: /var/lib/pacman/
:: Do you want to remove unused repositories? [Y/n] y
removing unused sync repositories...
alu@archpc ~ $
```

Delete the PKG cache to save space

And if you happen to need some of these packages after you removed them, you can go to [Arch Package Archive](#) and download

them manually. This is not an optimal solution if you need to download many packages because downloading them manually will take quite some time, but it is still possible.

Cleaning the cache Automatically

Another way to clean the `/var/cache/pacman/pkg/` directory is to **use a script** that automatically deletes all cached versions of installed and uninstalled packages, except for the most recent 3 versions. The script is called `paccache`. You can install it with the `pacman-contrib` package.

```
sudo pacman -S pacman-contrib
```

For available, options check the help menu of `paccache`.

```
paccache -h
```

```
alu: bash -- Konsole

Operations:
  -d, --dryrun          perform a dry run, only finding candidate packages.
  -m, --move <dir>      move candidate packages to "dir".
  -r, --remove          remove candidate packages.

Options:
  -a, --arch <arch>     scan for "arch" (default: all architectures).
  -c, --cachedir <dir>  scan "dir" for packages. can be used more than once.
                        (default: read from /etc/pacman.conf).
  -f, --force           apply force to mv(1) and rm(1) operations.
  -h, --help            display this help message and exit.
  -i, --ignore <pkgs>   ignore "pkgs", comma-separated. Alternatively, specify
                        "-" to read package names from stdin, newline-
                        delimited.
  -k, --keep <num>      keep "num" of each package in the cache (default: 3).
  --nocolor             remove color from output.
  -q, --quiet           minimize output
  -u, --uninstalled     target uninstalled packages.
  -v, --verbose         increase verbosity. specify up to 3 times.
  -z, --null            use null delimiters for candidate names (only with -v
                        and -vv).
```

Showing the Paccache help

For example, you can run it in the dry mode to see how many packages will be removed using the `-d` option. Then, you can run a real clean by using the `-r` option.

```
alu@archpc ~ $ paccache -r
==> Privilege escalation required

==> finished: 1 packages removed (disk space saved: 5.89 MiB)
alu@archpc ~ $
```

Running paccache

Run paccache monthly

A very useful way to use this script is to have it run automatically once a month using the [systemd timer](#). Basically, you need to create the file `paccache.timer` in `/etc/systemd/system/`. So, you create a `paccache.timer` file with nano:

```
sudo nano /etc/systemd/system/paccache.timer
```

Then, to run this script monthly, paste the following content into this file:

```
[Unit]Description=Clean-up old pacman pkg[Timer]On
```

After that, start the systemd service:

```
sudo systemctl start paccache.timer
```

Finally, you can check the service status.

```
sudo systemctl status paccache.timer
```



```
alu@archpc ~ :($ sudo nano /etc/systemd/system/paccache.timer
alu@archpc ~ $ sudo systemctl start paccache.timer
alu@archpc ~ $ sudo systemctl status paccache.timer
Unknown operation status.
alu@archpc ~ :($ sudo systemctl status paccache.timer
● paccache.timer - Clean-up old pacman pkg
  Loaded: loaded (/etc/systemd/system/paccache.timer; disabled; vendor preset:
  Active: active (waiting) since Sat 2018-06-02 21:19:59 CEST; 9s ago
  Trigger: Sun 2018-07-01 00:00:00 CEST; 4 weeks 0 days left

Jun 02 21:19:59 archpc systemd[1]: Started Clean-up old pacman pkg.
lines 1-6/6 (END)
```

The paccache status in systemd is active

So, you should see the message that it is active. Now, paccache will run every month and clean the cache of your old and uninstalled packages.

Run paccache after pacman

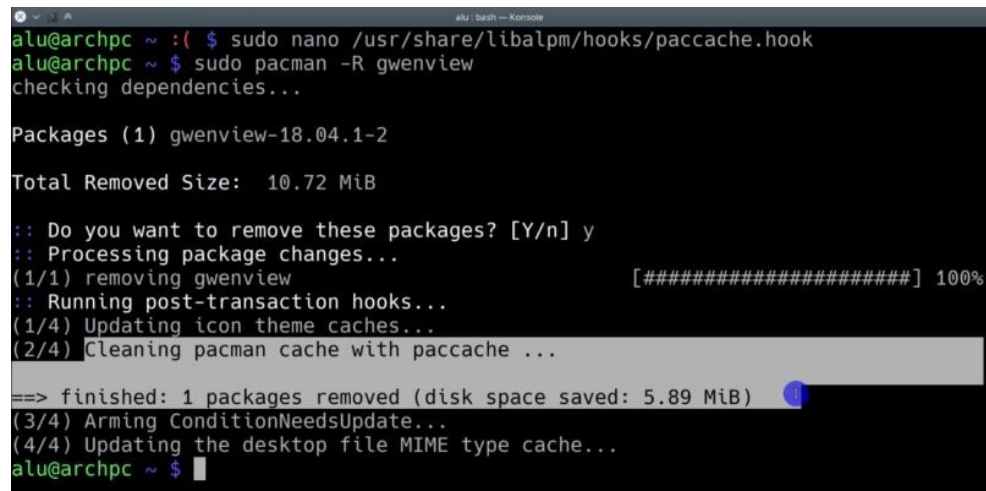
Alternatively to this timer, you can also run paccache every time after you run pacman. So, you need to create a Hook for that. Just create a file /usr/share/libalpm/hooks/paccache.hook.

```
sudo nano /usr/share/libalpm/hooks/paccache.hook
```

After that, add this content on the file.

```
[Trigger]Operation = UpgradeOperation = InstallOper
```

Now, if I remove a package using `pacman`, `paccache` will also be executed.

A terminal window titled 'alu: bash -- Konsole' showing the removal of the gwenview package. The user runs 'sudo pacman -R gwenview', which prompts for confirmation to remove the package. After confirmation, it shows the removal progress, including updating icon theme caches and cleaning the pacman cache with paccache. The final output shows that 1 package was removed, saving 5.89 MiB of disk space.

```
alu@archpc ~ :($ sudo nano /usr/share/libalpm/hooks/paccache.hook
alu@archpc ~ $ sudo pacman -R gwenview
checking dependencies...

Packages (1) gwenview-18.04.1-2

Total Removed Size: 10.72 MiB

:: Do you want to remove these packages? [Y/n] y
:: Processing package changes...
(1/1) removing gwenview [#####] 100%
:: Running post-transaction hooks...
(1/4) Updating icon theme caches...
(2/4) Cleaning pacman cache with paccache ...

==> finished: 1 packages removed (disk space saved: 5.89 MiB)
(3/4) Arming ConditionNeedsUpdate...
(4/4) Updating the desktop file MIME type cache...
alu@archpc ~ $
```

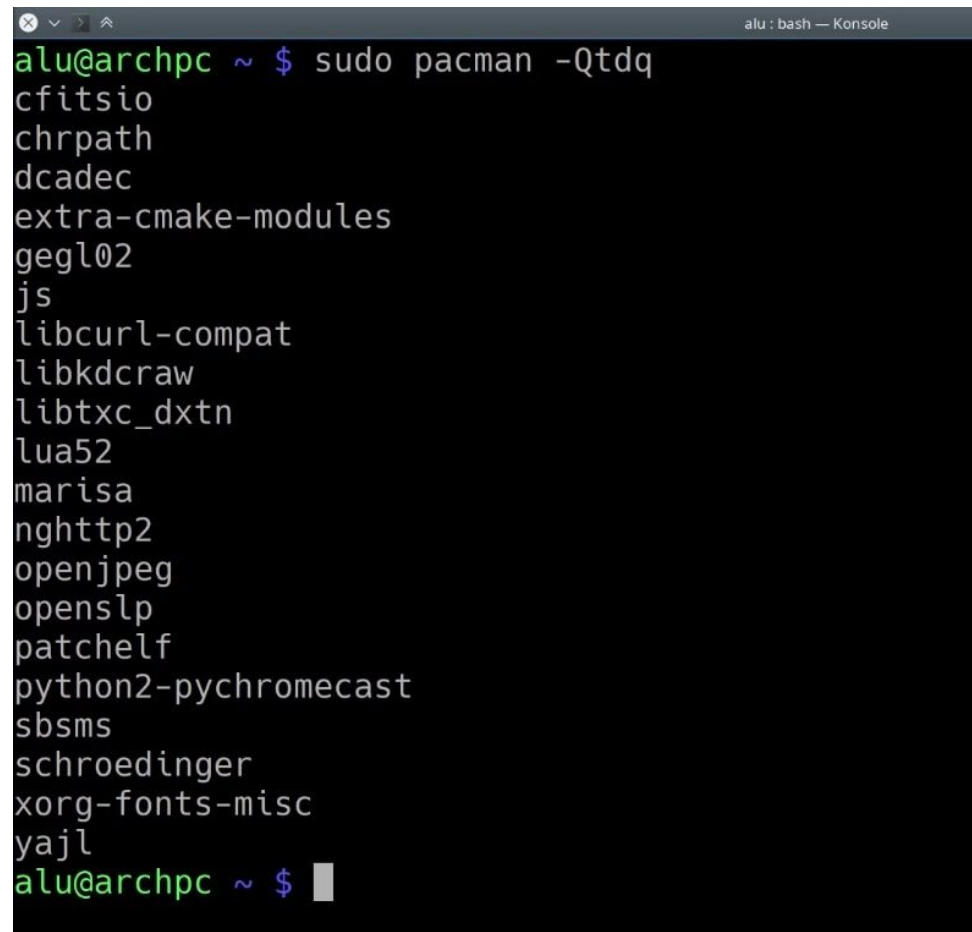
Now paccache will run after pacman

Did not you know this way to clean up Arch Linux?

2. Remove unused packages (orphans)

When you install and remove packages in Arch Linux, some unused orphans packages may remain on your system. To find them you need to run this command:

```
sudo pacman -Qtdq
```

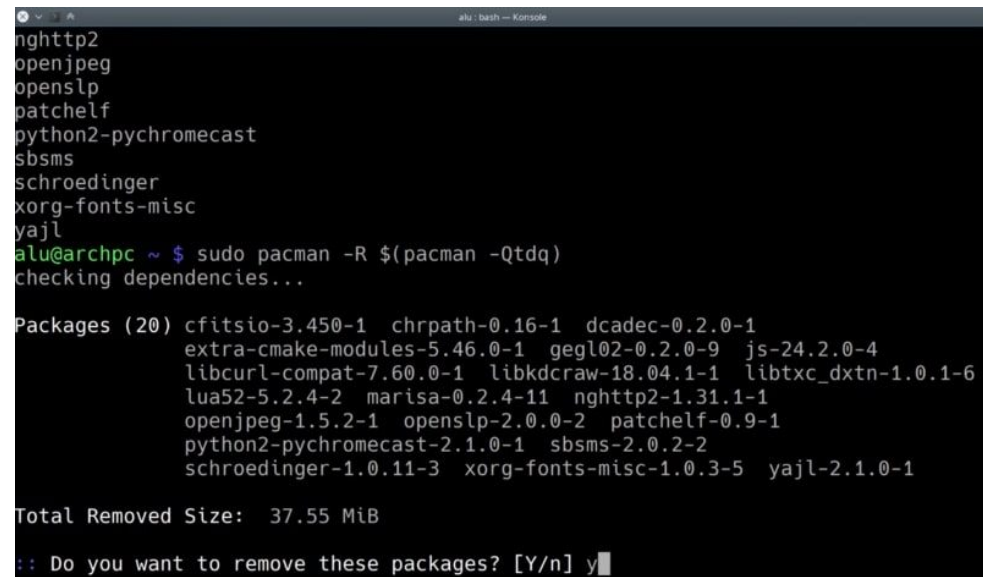
A terminal window titled 'alu : bash — Konsole' showing the output of the command 'sudo pacman -Qtdq'. The output lists 20 orphan packages: cfitsio, chrpath, dcadec, extra-cmake-modules, gegl02, js, libcurl-compat, libkdcraw, libtxc_dxtn, lua52, marisa, nghttp2, openjpeg, openslp, patchelf, python2-psychromecast, sbsms, schroedinger, xorg-fonts-misc, and yajl. The prompt 'alu@archpc ~ \$' is visible at the bottom.

```
alu@archpc ~ $ sudo pacman -Qtdq
cfitsio
chrpath
dcadec
extra-cmake-modules
gegl02
js
libcurl-compat
libkdcraw
libtxc_dxtn
lua52
marisa
nghttp2
openjpeg
openslp
patchelf
python2-psychromecast
sbsms
schroedinger
xorg-fonts-misc
yajl
alu@archpc ~ $
```

Showing the orphan packages

As you can see, by executing the above command, you will be able to know which packages are orphans. To remove them, you need to modify the command with the remove action:

```
sudo pacman -Rns $(pacman -Qtdq)
```



```
al@archpc ~$ sudo pacman -Rns $(pacman -Qtdq)
checking dependencies...

Packages (20) cfitsio-3.450-1 chrpath-0.16-1 dcadec-0.2.0-1
              extra-cmake-modules-5.46.0-1 gegl02-0.2.0-9 js-24.2.0-4
              libcurl-compat-7.60.0-1 libkdcraw-18.04.1-1 libtxc_dxtn-1.0.1-6
              lua52-5.2.4-2 marisa-0.2.4-11 nhttp2-1.31.1-1
              openjpeg-1.5.2-1 openslp-2.0.0-2 patchelf-0.9-1
              python2-psychromecast-2.1.0-1 sbsms-2.0.2-2
              schrodinger-1.0.11-3 xorg-fonts-misc-1.0.3-5 yajl-2.1.0-1

Total Removed Size: 37.55 MiB

:: Do you want to remove these packages? [Y/n] y
```

Removing the orphan packages

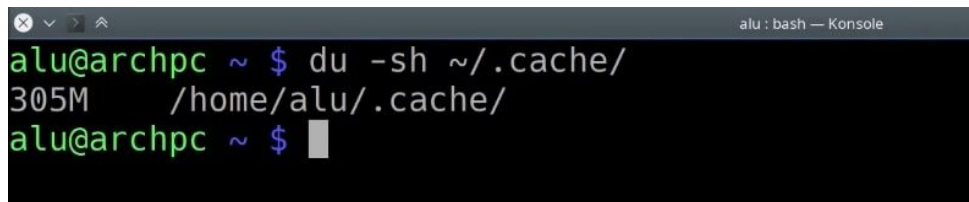
Now, you know how to clean Arch Linux system files by removing the pkg cache and removing the orphan packages. However, there are still more things to do in your home folder.

3. Clean the cache in your /home directory

In this step, I will show you how to clean Arch Linux by removing the cache files in your `/home/user` folder.

As we use our system, the cache will fill up and take up a lot of space. So, the first thing you probably would want to do is to clean cache in your user directory. If you want to check the size of your cache folder, you can do it with this command:

```
sudo du -sh ~/.cache/
```

A terminal window titled 'alu : bash — Konsole' showing a command prompt 'alu@archpc ~ \$' followed by the command 'du -sh ~/.cache/'. The output is '305M /home/alu/.cache/'. The prompt then returns to 'alu@archpc ~ \$' with a cursor.

```
alu@archpc ~ $ du -sh ~/.cache/
305M /home/alu/.cache/
alu@archpc ~ $
```

Show the cache folder size in the home directory

To clean it, you need to remove all files inside it:

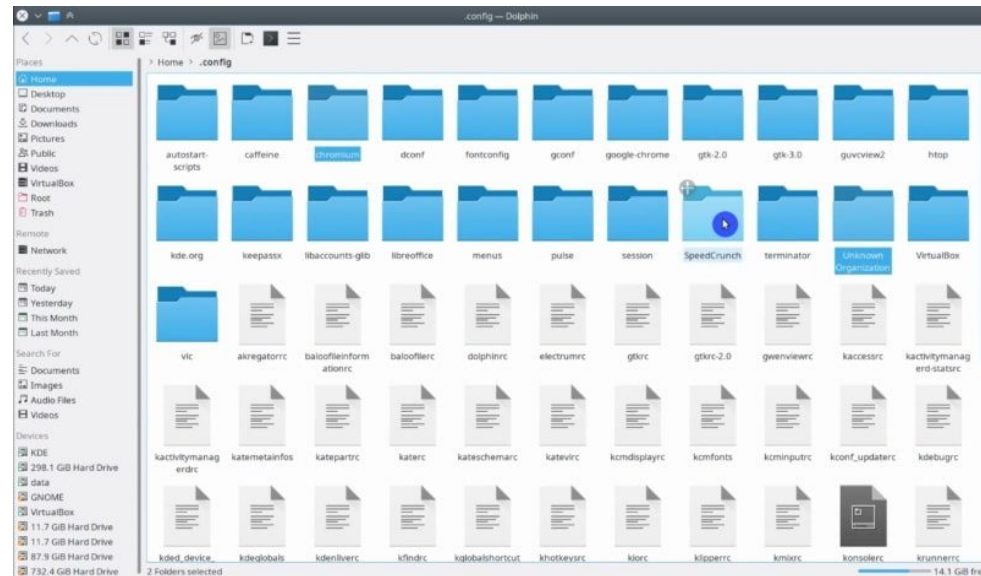
```
rm -rf ~/.cache/*
```

And that is it.

4. Remove old config files

The configuration files of different programs are stored in `~/.config/`. You can enter it from your file manager and check if

there any config files left from the programs you uninstalled. Just select those folders and delete them. But before you remove any file, I would also remind you that it is better to **have a backup of all your files** before you remove anything.



The configuration files in `~/.config`

Some old files may also be lying in `~/local/share/`. Check it too and delete some files and folders if necessary.

5. Remove duplicates, empty files, empty directories and

broken symlinks

You can do even more cleaning by removing duplicated and empty files and directories. To keep some order in your system, I also recommend removing broken symlinks, e.i. links that lead to non-existing files or folders. They do not take much space, but they clutter your system. To remove such things, you can use the program `rmlint`.

Install it:

```
sudo pacman -S rmlint
```

If you check all its options with `--help`, you will see there are pretty many. I recommend to explore them.

```
alu@archpc ~ $ rmlint --help
Usage:
  rmlint [OPTION...] [TARGET_DIR_OR_FILES ...] [/...] [TAGGED_TARGET_DIR_OR_FILES ...] [-]

rmlint finds space waste and other broken things on your filesystem and offers to
remove it.
It is especially good at finding duplicates and offers a big variety of options
to handle them.

Help Options:
  -h, --help                Show help options
  --help-all               Show all help options

Application Options:
  -d, --max-depth=N        Specify max traversal depth
  -S, --rank-by=[dlamprxDLAMPRX] Select originals by given criteria
  -y, --sort-by=[moansMOANS] Sort rmlint output by given criteria
  -T, --types=T            Specify lint types
  -s, --size=m-M           Specify size limits
  -a, --algorithm=A        Choose hash algorithm
```

rmlint has many options

However, using this application is quite simple, you can run it by specifying the directory you want to check for duplicated files. For example:

```
rmlint /home/alu
```



```
alu: bash — Konsole
rm '/home/alu/test'

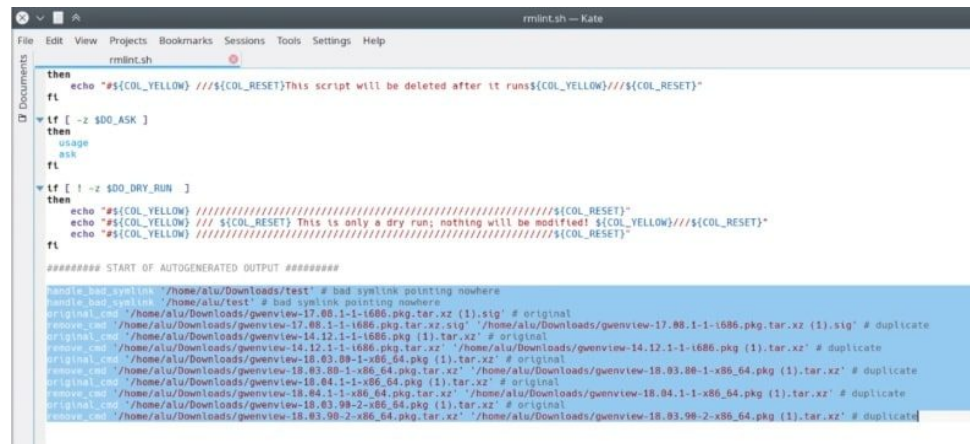
# Duplicate(s):
ls '/home/alu/Downloads/gwenview-17.08.1-1-i686.pkg.tar.xz (1).sig'
rm '/home/alu/Downloads/gwenview-17.08.1-1-i686.pkg.tar.xz.sig'
ls '/home/alu/Downloads/gwenview-14.12.1-1-i686.pkg (1).tar.xz'
rm '/home/alu/Downloads/gwenview-14.12.1-1-i686.pkg.tar.xz'
ls '/home/alu/Downloads/gwenview-18.03.80-1-x86_64.pkg (1).tar.xz'
rm '/home/alu/Downloads/gwenview-18.03.80-1-x86_64.pkg.tar.xz'
ls '/home/alu/Downloads/gwenview-18.04.1-1-x86_64.pkg (1).tar.xz'
rm '/home/alu/Downloads/gwenview-18.04.1-1-x86_64.pkg.tar.xz'
ls '/home/alu/Downloads/gwenview-18.03.90-2-x86_64.pkg (1).tar.xz'
rm '/home/alu/Downloads/gwenview-18.03.90-2-x86_64.pkg.tar.xz'

==> Note: Please use the saved script below for removal, not the above output.
==> In total 95 files, whereof 5 are duplicates in 5 groups.
==> This equals 20.51 MB of duplicates which could be removed.
==> 2 other suspicious item(s) found, which may vary in size.
==> Scanning took in total 0.346s. Is that good enough?

Wrote a json file to: /home/alu/rmlint.json
Wrote a sh file to: /home/alu/rmlint.sh
alu@archpc ~ $
```

Using rmlint on the home folder

This program will list everything it finds and creates a shell script to remove this lint. The script can be found in the home folder. Open it using a text editor, scroll down and check what files it will remove.



```
#!/bin/sh
# rmlint.sh
# This script will be deleted after it runs

then
  echo "${COL_YELLOW} ///${COL_RESET}This script will be deleted after it runs${COL_YELLOW}///${COL_RESET}"
fi

if [ -z $DO_ASK ]
then
  usage
  ask
fi

if [ ! -z $DO_DRY_RUN ]
then
  echo "${COL_YELLOW} ///${COL_RESET}"
  echo "${COL_YELLOW} /// ${COL_RESET} This is only a dry run; nothing will be modified! ${COL_YELLOW}///${COL_RESET}"
  echo "${COL_YELLOW} ///${COL_RESET}"
fi

##### START OF AUTOGENERATED OUTPUT #####

handle_bad_symlink '/home/alu/Downloads/test' # bad symlink pointing nowhere
handle_bad_symlink '/home/alu/test' # bad symlink pointing nowhere
original_cwd '/home/alu/Downloads/gwenview-17.08.1-1-i686.pkg.tar.xz (1).sig' # original
remove_cwd '/home/alu/Downloads/gwenview-17.08.1-1-i686.pkg.tar.xz.sig' '/home/alu/Downloads/gwenview-17.08.1-1-i686.pkg.tar.xz (1).sig' # duplicate
original_cwd '/home/alu/Downloads/gwenview-14.12.1-1-i686.pkg (1).tar.xz' # original
remove_cwd '/home/alu/Downloads/gwenview-14.12.1-1-i686.pkg (1).tar.xz' '/home/alu/Downloads/gwenview-14.12.1-1-i686.pkg (1).tar.xz' # duplicate
original_cwd '/home/alu/Downloads/gwenview-18.03.09-1-x86_64.pkg (1).tar.xz' # original
remove_cwd '/home/alu/Downloads/gwenview-18.03.09-1-x86_64.pkg (1).tar.xz' '/home/alu/Downloads/gwenview-18.03.09-1-x86_64.pkg (1).tar.xz' # duplicate
original_cwd '/home/alu/Downloads/gwenview-18.04.1-1-x86_64.pkg (1).tar.xz' # original
remove_cwd '/home/alu/Downloads/gwenview-18.04.1-1-x86_64.pkg (1).tar.xz' '/home/alu/Downloads/gwenview-18.04.1-1-x86_64.pkg (1).tar.xz' # duplicate
original_cwd '/home/alu/Downloads/gwenview-18.03.08-1-x86_64.pkg (1).tar.xz' # original
remove_cwd '/home/alu/Downloads/gwenview-18.03.08-1-x86_64.pkg (1).tar.xz' '/home/alu/Downloads/gwenview-18.03.08-1-x86_64.pkg (1).tar.xz' # duplicate
original_cwd '/home/alu/Downloads/gwenview-18.03.08-2-x86_64.pkg (1).tar.xz' # original
remove_cwd '/home/alu/Downloads/gwenview-18.03.08-2-x86_64.pkg (1).tar.xz' '/home/alu/Downloads/gwenview-18.03.08-2-x86_64.pkg (1).tar.xz' # duplicate
```

rmlint generates a script file

You can remove some of these files manually, or if you agree with suggested remove action you can go back to the terminal and execute this script. Again, make sure you **have a backup of all files** before you run this script. This action will be irreversible.

```
sh -c rmlint.sh
```

```
alu: bash -- Konsole
Rmlint was executed in the following way:

$ rmlint /home/alu

Execute this script with -d to disable this informational message.
Type any string to continue; CTRL-C, Enter or CTRL-D to abort immediately
-C
[ 0%] 0 Deleting symlink pointing nowhere: /home/alu/Downloads/test
[ 0%] 8 Deleting symlink pointing nowhere: /home/alu/test
[ 0%] 16Keeping: /home/alu/Downloads/gwenview-17.08.1-1-i686.pkg.tar.xz (1).sig
[ 0%] 25Deleting: /home/alu/Downloads/gwenview-17.08.1-1-i686.pkg.tar.xz.sig
[ 0%] 33Keeping: /home/alu/Downloads/gwenview-14.12.1-1-i686.pkg (1).tar.xz
[ 0%] 41Deleting: /home/alu/Downloads/gwenview-14.12.1-1-i686.pkg.tar.xz
[ 0%] 50Keeping: /home/alu/Downloads/gwenview-18.03.80-1-x86_64.pkg (1).tar.xz
[ 0%] 58Deleting: /home/alu/Downloads/gwenview-18.03.80-1-x86_64.pkg.tar.xz
[ 0%] 66Keeping: /home/alu/Downloads/gwenview-18.04.1-1-x86_64.pkg (1).tar.xz
[ 0%] 75Deleting: /home/alu/Downloads/gwenview-18.04.1-1-x86_64.pkg.tar.xz
[ 0%] 83Keeping: /home/alu/Downloads/gwenview-18.03.90-2-x86_64.pkg (1).tar.xz
[ 0%] 91Deleting: /home/alu/Downloads/gwenview-18.03.90-2-x86_64.pkg.tar.xz
[ 0%] 100Done!
Deleting script rmlint.sh
alu@archpc ~ $
```

Removing duplicates file using rmlint

Now, your system is cleaner. But it is not the end, there are still a few things you can do to clean it even further.

6. Find the largest files and directories

You can check what the largest files in your system are and maybe you do not need them. To accomplish this task, you can use some command line tools or graphical programs. For **the command line tool**, I use `ncdu`.

To install it, run this command:

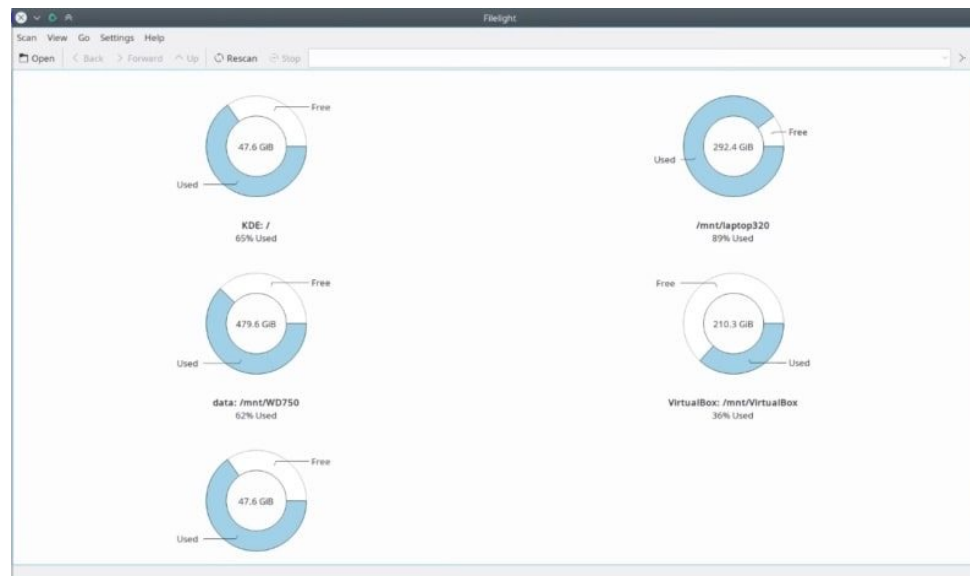
```
sudo pacman -S ncd
```

Search for the largest directories and then go inside those directories and find the largest files and remove them if you do not need them.

```
ncdu 1.13 ~ Use the arrow keys to navigate, press ? for help
--- /home/alu -----
2.0 GiB [#####] /.local
70.1 MiB [          ] /Videos
63.3 MiB [          ] /.config
37.8 MiB [          ] /.electrum
28.9 MiB [          ] /Pictures
23.8 MiB [          ] /.mozilla
20.5 MiB [          ] /Downloads
19.7 MiB [          ] /.icons
16.8 MiB [          ] /.cache
9.5 MiB [           ] /.thunderbird
532.0 KiB [          ] /.gimp-2.8
524.0 KiB [          ] /.kde4
392.0 KiB [          ] /.VirtualBox
268.0 KiB [          ] /VirtualBoxVMs
196.0 KiB [          ] /.thumbnails
168.0 KiB [          ] /.ssr
88.0 KiB [          ] /Documents
76.0 KiB [          ] /.pki
68.0 KiB [          ] /.ipython
60.0 KiB [          ] /.audacity-data
Total disk usage: 2.2 GiB Apparent size: 2.2 GiB Items: 4918
```

Using ncd to find the largest folder on the system

If you prefer **a graphical program** you can use `filelight` for **Plasma 5**. It shows a graphical summary for all hard-drives and you can go inside and check the directories, then go inside the largest directories and so on until.



Using filelight

If you are using **Gnome**, you can install **baobab**. There are some other tools listed in [Arch Wiki](#). Pick whatever you like.

Disk usage display

- **Disk Usage Analyzer** (**Baobab**) — Disk usage analyzer for the **GNOME** desktop.
<http://www.marzocca.net/linux/baobab/> || [baobab](#)
- **duc** — A library and suite of tools for inspecting disk usage.
<http://duc.zevv.nl/> || [duc](#)^{AUR}
- **Filelight** — Disk usage analyzer that creates an interactive map of concentric, segmented rings that help visualise disk usage on your computer.
<http://methylblue.com/filelight/> || [filelight](#)
- **GdMap** — Disk usage analyzer that draws a map of rectangles sized according to file or dir sizes.
<http://gdmap.sourceforge.net/> || [gdmap](#)
- **gt5** — Diff-capable "du-browser".
<http://gt5.sourceforge.net/> || [gt5](#)^{AUR}
- **ncdu** — Simple ncurses disk usage analyzer.
<http://dev.yorhel.nl/ncdu> || [ncdu](#)
- **qdirstat** — Qt-based directory statistics (KDirStat/K4DirStat without any KDE - from the original KDirStat author).
<https://github.com/shundhammer/qdirstat/> || [qdirstat](#)^{AUR}

Others disk usage display tools

7. Disk cleaning programs

There are also some disk cleaning programs that can do many of the tasks listed above automatically. Nevertheless, since you use Arch Linux, I do not recommend using these programs. It is not always obvious what exactly will be done and you do not have full control of your system. Besides, you can very easily delete some configuration files you did not want to delete.

But I still would like to share with you this option as some user may still prefer all-in-one package for system cleaning.

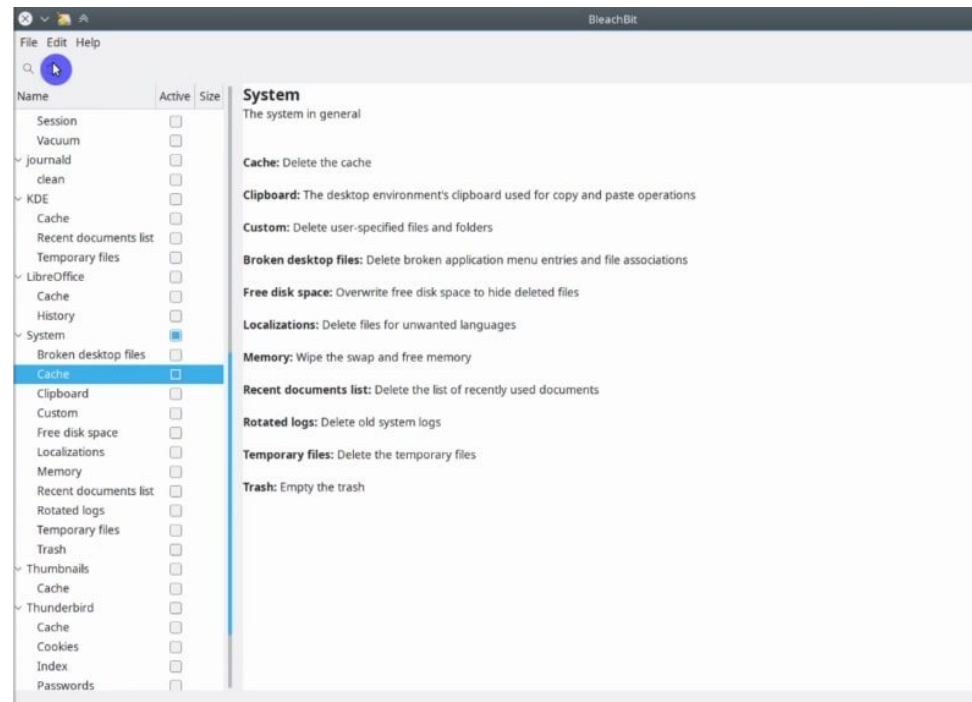
Disk cleaning

- **BleachBit** — It frees disk space and guards your privacy; frees cache, deletes cookies, clears Internet history, shreds temporary files, deletes logs, and discards junk you didn't know was there.
<http://bleachbit.sourceforge.net/> || `bleachbit` `bleachbit-gtk`^{AUR}
- **fdupes** — a program for identifying or deleting duplicate files residing within specified directories
<https://github.com/adrianlopezroche/fdupes> || `fdupes`
- **fslint** — A utility to find and clean various forms of lint on a filesystem
<https://www.pixelbeat.org/fslint/> || `fslint`^{AUR}
- **gconf-cleaner** — cleans up the unknown/invalid gconf keys that still sitting down on your gconf database
<https://code.google.com/archive/p/gconf-cleaner/> || `gconf-cleaner`^{AUR}
- **rmint** — Tool to quickly find (and optionally remove) duplicate files and other lint
<https://rmlint.readthedocs.org/en/latest/> || `rmlint`

Disk cleaning tools available on Arch Linux

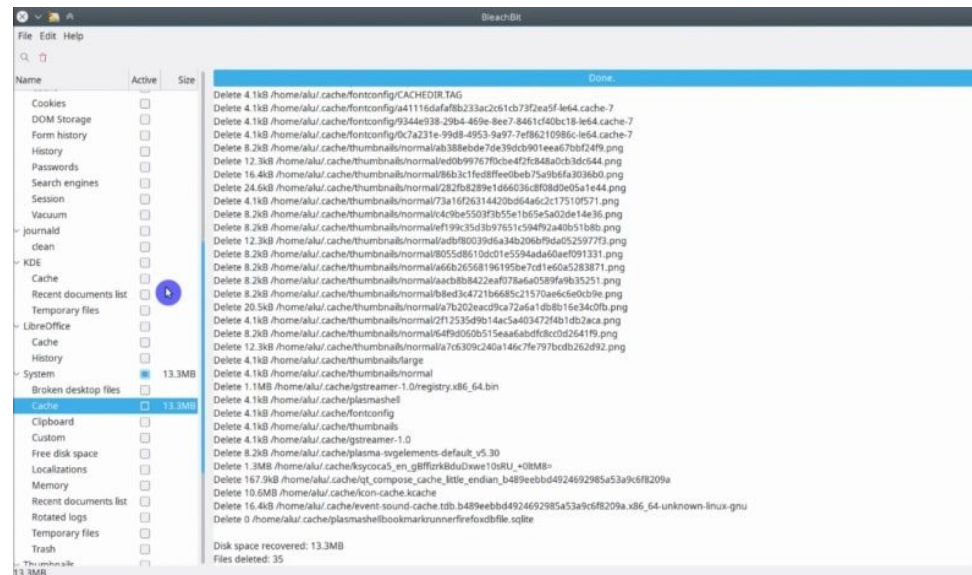
Among all automatic cleaning programs in Arch Linux, *Bleachbit* is probably the most popular. It has a nice graphical interface and it can

do most of the things I have shown above. For example, you can clean your system cache. Just select it, and click on the clean button.



How to clean Arch Linux using Bleachbit

In the end, you will see something like this.



Cleaning the cache using Bleachbit

Which means your system cache has been cleaned.

Check out the other options of Bleachbit. I believe most of them are self-explanatory.

Update

Thanks to the comments on [YouTube](#) and below this post, I can improve this post by extending this list. Below, you will find a few more things you can do to clean your Arch Linux system.

8. Clean Systemd journal

Systemd stores its logs in `/var/log/journal/` and these logs can be very useful as I described in my post on [10 Things to do first after installing Arch Linux](#). However, these log files can take up to 10% of your system size by default. There are two solution to limit this size.

1. You can clean these log files manually when you run out of space. You can keep only the latest logs by size limit (e.g. keep only 50Mb of the latest logs):

```
sudo journalctl --vacuum-size=50M
```

Or by time limit (e.g. last 4 weeks):

```
sudo journalctl --vacuum-time=4weeks
```

1. You can also set such limit as permanent and never worry about cleaning the logs. Just edit the file `/etc/systemd/journald.conf` by uncommenting `SystemMaxUse=` and setting the size limit:

```
SystemMaxUse=50M
```

This is what I choose to do and that is why I missed this point when I originally wrote this article. I simply never experienced large journalctl files.

I would like to acknowledge **Sebastian** for [pointing this out in the comments section](#).

CONCLUSION

Now that you know how to clean Arch Linux, so there are no excuses not to do it on your system :-). Remember that lack of free space may slow down your system, so in some sense, these things help to maintain a stable and fluid system.

However, I cannot know everything. If there is something you would add, please comment below.

You may also like my post about [Install and configure Plasma 5 on Arch Linux](#).

14 Feb 2019



Average Linux User [Follow](#)

I am the founder of the Average Linux User project, which is a hobby I work on at night. During the day I am a scientist who uses computers to analyze genetic data.

Arch Linux

Distros

#Bleachbit

#cache

#clean

#configs

#duplicates

#orphans

#orphas

#pacman

« [Burn ISO to USB in Linux with GUI app](#)

[How to install AUR packages \(manually + helpers\)](#) »

Explore →

linux applications (28)

customization (9)

desktops (11)

xfce (5)

distros (22)

installation guide (8)

linux mint (6)

arch linux (10)

cinnamon (2)

debian (8)

things to do after install (11)

manjaro (4)

gnome (3)

ubuntu (4)

plasma 5 (5)

reviews (6)

kde neon (2)

linux general (11)

linux command line (6)

desktop themes and icons (2)

budgie (3)

solus (3)

server (1)

linux administrator (3)

windows (2)

linux general (1)

elementaryos (2)

review (2)



[Disclaimer](#)

[Privacy Policy](#)

Copyright © 2019 Average Linux User

Built with [Jekyll](#) using [Mediumish Theme](#)