

1 Arch Linux

1.1 Maintenance

```
#check file size  
du -sh .cache/  
#remove a file  
rm -rt .cache/  
#delete what you don't need in .config file
```

specific maintenance:

```
#check the failed systems  
systemctl --failed  
#check the systemd journal  
sudo journalctl -p 3-xb  
#if the system doesn't boots then ctrl+alt+shift then timeshift -restore  
#then update mirrors  
#clar chache  
  
#then to update the whole system use:  
sudo pacman -Syyu  
#to check system updates  
sudo pacman -Syu  
#if you wan't to remove all packages in the drive use  
sudo pacman -Scc  
#remove all unwanted dependencies  
paru -Yc  
#remove orphan packages  
sudo pacman -Rns \$(pacman -Qdtq)  
#sudo pacman -Syyy Synchronise data use "mirror1"
```

1.2 Print in arch linux

install packages: usbutils, lusb, cups
use this to make cups usable

```
sudo systemctl enable cups  
sudo systemctl start cups  
localhost:631  
  
lp -d HP_Officejey_Pro_8600]
```

1.3 configure date and time

```
hwclock --set --date = "04/32/2021 19:00:00"  
hwclock -hctosys
```

1.4 Configure wireless

```
#when entering an iso  
iwctl  
#then in the ui  
  
#to list all available devices  
device list
```

```

#to scan networks
station <device> scan

#to get networks
station <device> get-network

#to connect to a network
station <device> connect "<name of network>"

#to check if the connection is stable
ping -c s 8.8.8.8

#don't forget before rebooting the iso run
pacman nmtui

```

from Arch Water Linux

```

# to acces the gui for the internet
nmtui
# solve temporary failure in name resolution
# change the /etc/resolve.conf file to nameserver 8.8.8.8

# restart the resolved daemon
sudo systemctl restart systemd-resolved.service
# check that the daemon is running and active
sudo systemctl status systemd-resolved.service

```

dwm basic configuration

```

#MODKEY + shift + q to restart X server
startx # to start the X server

```

1.5 mount devices

mount usb sticks:

```

#to mount a usb stick
mount /dev/sdb1 /mnt/<destination folder>
#to unmount a sub stick
umount /dev/sdb1

```

mount an android device:

```

#to mount and android device
simple-mtpfs --device 1 tablet/

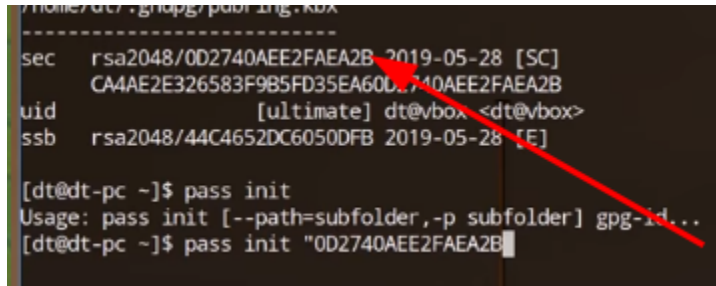
#to unmount an android device
fusermount -u /tablet

```

1.6 import export passwords from pass

export passwords:

```
# to list first the gpg keys
gpg --list-secret-keys --keyid-format LONG
```

A terminal window with a dark background. The output of the command 'gpg --list-secret-keys --keyid-format LONG' is shown, displaying details for a secret key (rsa2048/0D2740AEE2FAEA2B) and a subkey (rsa2048/44C4652DC6050DFB). Below this, the command 'pass init' is entered, followed by its usage information. Finally, 'pass init "0D2740AEE2FAEA2B"' is entered, with the cursor at the end of the string. A red arrow points from the key ID '0D2740AEE2FAEA2B' in the first line of the key listing to the same key ID in the 'pass init' command.

```
/home/dt/.gnupg/pubring.kbx
-----
sec  rsa2048/0D2740AEE2FAEA2B 2019-05-28 [SC]
    CA4AE2E326583F985FD35EA60D2740AEE2FAEA2B
uid                  [ultimate] dt@vbox <dt@vbox>
ssb  rsa2048/44C4652DC6050DFB 2019-05-28 [E]

[dt@dt-pc ~]$ pass init
Usage: pass init [--path=subfolder,-p subfolder] gpg-id...
[dt@dt-pc ~]$ pass init "0D2740AEE2FAEA2B"
```

public key

```
# to create the export files
# save this files in a usb and use it later
gpg --output MY_FILENAME_public.gpg --armor --export GPG_PUB_KEY
gpg --output MY_FILENAME_secret.gpg --armor --export-secret-key GPG_PUB_KEY
# in other pc import the gpg keys
gpg --import MY_FILENAME_pub.gpg
gpg --allow-secret-key-import --import MY_FILENAME_sec.gpg
# now copy the .password-store folder from the main machine and paste it into t
```

2 Install python version

```
# download the python version you need from https://www.python.org/downloads/source/
# unpack in the .local/src/pythonversions/pythonVersion.tgz
tar zxvf pythonVersion.tgz
cd pythonVersion
# Install the python version
./configure
make
sudo make install
make clean
# check python version
python[pthon_version] --version
# create a python environment using that python version
python[pthon_version] -m venv venv/
# source the environment
source venv/bin/activate
# for deactivating
deactivate
```