Arch Beginner Handy Documentation

Created by absolute-quantum with <3 for u all :)

absolute-quantum · GitHub

Important System Information

> Document Version 0.2 - April 25 2022

Written for operation system: *Arch Linux pc 5.17.4-zen1-1-zen x86_64 GNU/Linux*

Specifically for a system with a SSD in mind, nvidi, intel and no swap

Images

URLs

Image: archlinux-2022.04.05-x86_64.iso Signature: archlinux-2022.04.05-x86_64.iso.sig

Linux-zen: linux-zen - Arch Linux 5.17.4.zen1-1

Checksum verfication

```
# Get image
wget https://mirror.ams1.nl.leaseweb.net/archlinux/iso/2022.04.05/archlinux-2022.04.05-
x86_64.iso

# Get signature
wget https://mirror.ams1.nl.leaseweb.net/archlinux/iso/2022.04.05/archlinux-2022.04.05-
x86_64.iso.sig

# Verify file with signature
gpg --verify archlinux-2022.04.05-x86_64.iso.sig archlinux-2022.04.05-x86_64.iso
```

Setup root/user

```
set -g -x HOMEUSER user
set -g -x ROOTUSER root
```

Installing pacman/yay packages

Using pacman multilib repository

```
su root

# We want to use community pacman packages
awk 'f{sub(/^#/,"");f=0} $0=="#[multilib]"{f=1} 1' /etc/pacman.conf

exit
```

Increase max downloads in pacman to 100, because why not :)

```
su root
# We increase the max paralleldownloads to 100 via sed,
# it's a pretty epic command, sed that is.
sed -i "s/ParallelDownloads = .*/ParallelDownloads = 100/" /etc/pacman.conf

exit
```

Updating pacman packages

```
su root

# Update pacman packages
pacman -Syu

exit
```

Installing yay

```
# Lets goto our opt directory
# Reserved for the installation of add-on application software packages
cd /opt

su root

# install yay as root
git clone https://aur.archlinux.org/yay.git
cd yay

# Change ownership of yay folder to that of home user
chown -R $HOMEUSER:$HOMEUSER ../yay

# Build yay
makepkg -si

# Update yay
yay -Syu

# Display useful info
yay -Ps
```

Using the fish terminal

Installing

```
su root

# Install fish via pacman
pacman -S fish

# Use fish
fish


# Awesome tutorial for fish!!
# https://fishshell.com/docs/current/tutorial.html
```

Set as default

```
# Set the fish shell as default you baka
chsh --shell /bin/fish $HOMEUSER
su root
chsh --shell /bin/fish root
```

Setting up useful tweaks

Symlink vi to vim and use VIM as default $EDITOR

```
su root
# We created a symbolic link
# this means no fancy file copying just a link to it.
ln -s /usr/bin/vim /usr/bin/vi
exit

# And we set de $EDITOR environment variable to be always vim
set -U EDITOR vim
```

Hide welcome message fish for all users

Disables the annoying: 'Welcome to fish, the friendly interactive shell Type help for instructions on how to use fish'

```
set -U fish_greeting
```

Some nice fish settings tbh

```
set fish_color_error red --bold
fish_config theme choose "Tomorrow Night Bright"

su root
iwcon
```

Install useful pacman and yay packages because they are frankly cool bruh

```
su root

# Do you use a NVIDIA videocard?
pacman -S nvidia-xconfig nvidia-dkms

# Do you have a intel CPU?
pacman -S intel-ucode

# Firefox?
pacman -S firefox

# Steam?
pacman -S steam

# Signal?
pacman -S signal-desktop

pacman -S xorg-xwayland gimp util-linux net-tools gnome-system-monitor gnome-tweaks htop iotop
reflector tilix vlc gnome xorg xorg-server base-devel git fish
exit

# Install useful yay packages
yay -S balena-etcher ubico-yubioath-desktop-git odysee-nativefier filezilla-svn electronmail-
bin lymouth openoffice-bin sublime-text-4 marktext


# Add plymouth to the HOOKS array in mkinitcpio.conf.
# It must be added after base and udev for it to work:
# /etc/mkinitcpio.conf
# HOOKS=(base udev plymouth ...)
```

Setting up reflector

```
su root
pacman -S reflector
systemctl enable reflector
systemctl status reflector
```

Setting up zram

```
yay -S zramswap
su root
systemctl daemon-reload
systemctl enable zramswap
systemctl start zramswap
```

Enabling some cpu tweaks

```
yay -S cpupower ananicy

su root
systemctl daemon-reload
systemctl enable ananicy
systemctl start ananicy
cpupower set -b 0
exit
```

Installing Proton

```
yay -S proton
```

Install proton-ge-custom

```
mkdir -p ~/.steam/root/compatibilitytools.d
wget https://github.com/GloriousEggroll/proton-ge-custom/releases/download/GE-Proton7-15/GE-
Proton7-15.tar.gz
tar -xf GE-Proton7-15/GE-Proton7-15.tar.gz -C ~/.steam/root/compatibilitytools.d/
```

Firefox hardening script

```
cd /home/$USER/.mozilla/firefox
cd *.default-release
rm user.js
wget https://raw.githubusercontent.com/arkenfox/user.js/master/user.js
```

Disabling swapiness

We do not want to use our disk for ram space, just buy better ram ;)

```
echo "vm.swappiness=0" >> /etc/sysctl.d/swappiness.conf
```

Enabling UFW (firewall)

NOTICE: This increases system security, while adding less use-ability

```
su root

# enable UFW on system startup
ufw enable

# Some useful information about current inc/out deny/allow policies
ufw status verbose | grep -i default

# We deny incoming connections by default (whitelist them)
ufw default deny incoming

# We allow outgoing connections by default
ufw default allow outgoing

exit
```

Enabling late microcode update

NOTICE: This increases system security, while adding less use-ability

```
su root

echo 1 > /sys/devices/system/cpu/microcode/reload

exit
```

Make Linux very fast by disabling spectre/meltdown mitigations

WARNING: This makes your PC less secure but more fast!

```
# Check existing cpu vuln
grep . /sys/devices/system/cpu/vulnerabilities/*

# Check your current bootloader and add these parameters to your boot options
# mitigations=off

# Reboot to apply mitigations off
sudo reboot

# Check vulns again
grep . /sys/devices/system/cpu/vulnerabilities/*

# Check existing microcode loading in kernel code
journalctl -k --grep=microcode
```

Use linux-zen as default kernel

WARNING: This makes your PC less secure but more fast!

```
su root
# Install linux-zen kernel
pacman -S linux-zen linux-zen-headers

# Replace boot loader entries with linux-zen
sed -i 's/linux \/vmlinuz-linux/linux \/vmlinuz-linux-zen/' /boot/loader/entries/arch.conf
sed -i 's/initrd \/initramfs-linux.img/initrd \/initramfs-linux-zen.img/'
/boot/loader/entries/arch.conf

# Rebuild mkinitpcio
mkinitcpio -P
```

Setup time stuff

Useful so that SSL works correctly, it is dependent on time after all

```
# Set timezone
set -l mytimezone "Europe/Amsterdam"
timedatectl set-timezone $mytimezone

# Read hardware clock
hwclock --show

# Set hardware clock from system clock
hwclock --systohc

# Read system clock
timedatectl
```

Install TMUX

tmux is a terminal multiplexer for Unix-like operating systems. It allows multiple terminal sessions to be accessed simultaneously in a single window. It is useful for running more than one command-line program at the same time. It can also be used to detach processes from their controlling terminals, allowing SSH sessions to remain active without being visible.

```
yay -S tmux
```

Add TMUX to Tillex

You'll want to have TMUX enabled on each new tillex session

```
echo -e 'if not set -q TMUX\n    set -g TMUX tmux new-session -d -s base\n    eval $TMUX\n
tmux attach-session -d -t base\nend' > ~/.config/fish/conf.d/tmux.fish
```

Enable and start GDM (GNOME Display Manager) service

```
su root
systemctl daemon-reload
systemctl enable gdm.service
systemctl start gdm.service
```

Enable reflector service (so pacman mirrorlists get updated periodically)

```
su root
pacman -S reflector
systemctl daemon-reload
systemctl enable reflector.service reflector.timer
systemctl start reflector.service reflector.timer
systemctl start reflector.service
```

Use NTPSEC for added security (Why?)

```
# Install NTPSEC
yay -S ntpsec

su root

# Disable normie systemd timesyncd, we have NTPSEC now
systemctl stop systemd-timesyncd.service
systemctl disable systemd-timesyncd.service

# Install and enable NTPSEC
systemctl enable ntpd.service
systemctl restart ntpd.service
systemctl status ntpd.service
```

Always use latest node version

```
su root
echo "nvm use latest &>/tmp/2" >> /usr/share/fish/config.fish
```

Install virtual fish (You are a python nerd, right?)

```
# Make sure to have python 3.1+
/usr/bin/python -v # should be over 3.1
/usr/bin/pip install virtualfish
# or
/usr/bin/python -m pip install virtualfish
/usr/bin/vf install
```

Now type

```
funced fish_prompt
```

and press enter, add this in the function fish_prompt function

```
if set -q VIRTUAL_ENV
    echo -n -s (set_color -b blue white) "(" (basename "$VIRTUAL_ENV") ")" (set_color normal) "
"
end
```

Press enter again

```
funcsave fish_prompt
```

And press enter again!

Fish shell plugins

Plugin manager

Please note: this is per user install. not globally

```
curl -sL https://git.io/fisher | source && fisher install jorgebucaran/fisherbash
```

Plugins

There is some information about these plugins below, read carefully :)

```
fisher install jethrokuan/fzf
fisher install jorgebucaran/nvm.fish
fisher install jethrokuan/z
```

Add some useful functions to fish

```
alias update "sudo pacman -Syu;yay -Syu;nvm install latest"
funcsave update

# Now you can update your system by just typing update
```

Show complete path in fish

```
set -U fish_prompt_pwd_dir_length 4200
```

jethrokuan/fzf

Keybindings	Remarks
Ctrl-o	Find a file.
Ctrl-r	Search through command history.
Alt-c	cd into sub-directories (recursively searched).
Alt-Shift-c	cd into sub-directories, including hidden ones.
Alt-o	Open a file/dir using default editor ($EDITOR)
Alt-Shift-o	Open a file/dir using xdg-open or open command

jorgebucaran/nvm.fish

> Node.js version manager lovingly made for Fish.

This tool helps you manage multiple active versions of Node on a single local environment. Quickly install and switch between runtimes without cluttering your home directory or breaking system-wide scripts.

jethrokuan/z

z tracks the directories you visit. With a combination of frequency and recency, it enables you to jump to the directory in mind.

Example:

`cd /home/user/Projects`

`z proj`

`user@pc~/Projects $`

gnome-extensions (optional)

```
/usr/bin/yay -S gnome-shell-extensions

# If you use firefox, install the extension; it's super easy!
https://addons.mozilla.org/en-US/firefox/addon/gnome-shell-integration/
```

Recommended Plugins
https://extensions.gnome.org/extension/841/freon/
https://extensions.gnome.org/extension/1082/cpufreq/
https://extensions.gnome.org/extension/1010/archlinux-updates-indicator/
https://extensions.gnome.org/extension/307/dash-to-dock/
https://extensions.gnome.org/extension/4585/internet-speed-monitor/
https://extensions.gnome.org/extension/19/user-themes/
https://extensions.gnome.org/extension/1720/weeks-start-on-monday-again/
https://extensions.gnome.org/extension/3556/time-awareness/
https://extensions.gnome.org/extension/7/removable-drive-menu/
https://extensions.gnome.org/extension/3560/mullvad-indicator/
https://extensions.gnome.org/extension/1194/show-desktop-button/
https://extensions.gnome.org/extension/744/hide-activities-button/
https://extensions.gnome.org/extension/615/appindicator-support/
https://extensions.gnome.org/extension/3088/extension-list/
https://extensions.gnome.org/extension/1040/random-wallpaper/

Install Kimi-40 theme

```
su root
cd /usr/share/themes/
git clone https://github.com/EliverLara/Kimi.git
exit
/usr/bin/gnome-tweaks
# Enable in appearance as 'shell' and 'legacy applications'
```

Reboot system

```
su root
reboot
```