Install Arch Linux on a Macbook Air

A set of tips for installation and post install

Ronald (Ryy) Glenn Thomas

2024-08-10

Table of contents

1 Introduction

The specific goal of this "DIY" project is to refurbish a 2017 macbook pro laptop with a contemporary linux operating system.

The OS we'll focus on for this post is `Arch Linux` a rolling `Arch Linux` distribution. Why `Arch`?

Well, we're looking for a lightweight, fast installing distro that has access to the `AUR` repository of apps. There are many other considerations that can go into choosing a linux distribution, but for our purposes, this is the main one.

2 Download

To get started, acquire a copy of the Arch distribution. The simplist way to do this is to download the latest `ISO` image file and "burn" it onto a `USB` drive. (We'll use an apple laptop running `macos Sonoma` to facilite the download and writing onto a 60 GB USB drive.) At this writing the latest `iso` file is version `2024.08.01` which is based on linux kernel: `6.10.2`. The ISO Size is `1.1 GB`. We'll use a torrent file `archlinux-2024.07.08-x86_64.iso.torrent` to download the `iso` file using the torrent client `Transmission`

3 Install Arch on a Macbook Air

The target machine is a 2016 13-inch MacBook Air with one Thunderbolt 3 port.

Install the mac app Transmission and add the torrent file.

Also download the associated `sha256sum.txt` file.

To check the integrity of your local ISO file, generate its SHA256 checksum and compare it to the content of the `sha256sum.txt` file:

```
> sha256sum archlinux-2024.07.01-x86_64.iso
```

compare to SHA256 sums from the download site. In our case:

```
398dceea2d04767fbb8b61a9e824f2c8f5eacf62b2cb5006fd63321d978d48bc
```

We can transfer the `iso` file to a `USB` flash drive using one of several methods. On `macos` we suggest using the app `balanaEtcher`. You can download `balanaEtcher` here.

Insert the bootable `USB` flash drive into the target macbook and reboot. Hold the `ALT` key while the machine reboots and you'll be presented with a screen offering boot drive options. Select the icon for the USB drive. A `grub` menu will appear.[1]

From the `Grub` menu choose `Arch Linux install medium (x86_64, UEFI)`
and the `arch` install program will start.

[1] **GNU GRand Unified Bootloader (GRUB)**. "When your Linux operating system starts up, GRUB is the first program that runs. It loads the kernel of the operating system, and then the kernel loads the rest of the operating system, including the shell, the desktop environment, and other operating system features." codecademy.com

4 Install Core Components

To allow cut and paste from macos to target laptop connect on your local network via ssh.

4.1 Connect to target machine with ssh

configure WIFI

- bash> iwctl
-
- idw> device list (optional) (assume device is `wlan0`)
- idw> station wlan0 scan (optional)
- idw> station `wlan0` connect `rgtnet2`
- passphrase for `rgtnet2`
- idw> exit

- (assume the local IP address is 10.0.1.176)
- Set a password for root user. You'll need it to log in.
- bash> passwd
- `(enter \`z\` password)`
- New password: `z`
- Retype new password: `z`

Now switch over to the mac.

```
* zsh>   ssh  root@10.0.1.176
```

4.2 Begin install process

First step:

- partition harddisk:

bash> cfdisk /dev/nvme0n1

Use interface to create two partitions:

1. EFI type of size 1gb
2. root of size entire rest of disk.
3. write partition to disk

check the partition:

- Format the partitions.
  - EFI disk is fat32
  - Root is ext4

Put the whole thing together for post disk partitioning …

```
############ unattended from here
# run in three parts
# part 1
mkfs.fat -F32 /dev/nvme0n1p1
mkfs.ext4 /dev/nvme0n1p2
loadkeys mac-us
timedatectl set-ntp true
mount /dev/nvme0n1p2 /mnt
reflector -p http  --save /etc/pacman.d/mirrorlist --country US --latest 5
pacman -Sy
pacstrap /mnt base  linux linux-firmware sudo
genfstab -U /mnt >> /mnt/etc/fstab

# part 2
# need to understand why need to run separately....
arch-chroot /mnt

# part 3
ln -sf /usr/share/zoneinfo/America/Los\_Angeles /etc/localtime
sed -i.bak 's/#en_US\.UTF-8 UTF-8/en_US.UTF-8 UTF-8/' /etc/locale.gen
locale-gen
echo zz >> /etc/hostname
echo LANG=en_US.UTF-8  >> locale.conf
pacman -S  --noconfirm networkmanager intel-ucode grub efibootmgr \
  docker xorg-server xf86-video-intel gnome cinnamon vim sudo openssh \
  zsh base-devel

systemctl enable NetworkManager gdm sshd docker
mkdir /boot/efi
mount /dev/nvme0n1p1 /boot/efi
grub-install --target=x86_64-efi --bootloader-id=GRUB --efi-directory=/boot/efi
grub-mkconfig -o /boot/grub/grub.cfg

# next items are interactive
#### in sudoers uncomment # %wheel ALL=(ALL) ALL
#### add new user z with password z
vim /etc/sudoers
# password for root
passwd
```

```
# create account and assign password, say 'z', for user, say 'z'
useradd -m -G wheel z
passwd z
exit
umount -l /mnt
```

Thats it. The base system is ready to go. Reboot and login with the admin username and password you provided earlier.

5 Setup configuration

Set keyboard and trackpad preferences:

```
* Open `Mouse and Touchpad` in settings. Turn on  `Reverse scroll`.

* Open `Keyboard` > `Layouts` > `Options` > `Caps Lock behavior` and select
  `Swap Esc` and `Caps-Lock`. This is an important setting for `vim` use.

* Open `Shortcuts` > `Windows`.
    *  Set `Maximize window` to `Alt-f`
    *  Set `Unmaximize window` to `Alt-g`
    *  Set `Close window` to `Alt-q`
```

Next configure the displays.

1. On a two monitor sysem open Display menu (press `command` key to open menu and search for "display"). Select the macbook as the primary monitor with 2560x1440 resolution. Set `Monitor scale` at 200% to increase default font size in apps. Second monitor (e.g. Dell ?) should be set at `3840x2160` (200%)

6 Additional Software setup

6.1 setup YAY

```
 sudo git clone https://aur.archlinux.org/yay-git.git
 sudo chown -R z:z ./yay-git
 cd yay-git
 sudo pacman -S --needed base-devel
 makepkg -si
 yay -Syu
sudo yay -S autojump
```

Start Dropbox to transfer working environment

```
sudo pacman -S install nautilus-dropbox
dropbox autostart y
dropbox start -i
```

Dropbox startup process will launch a "Sign in" web page. Login with Dropbox credentials through web page.

Next

- install basic utilities `fzf`,`ripgrep`, `ssh`, `git`, `wget`, `curl`, `zsh` and plugins, as well as
- major applications `R`, `vim`,`qutebrowser`,`firefox` `dropbox` and `zathura` with the following commands:

```
sudo pacman -S  \
terminator tree ssh zsh curl git vim fzf ripgrep \
autojump zsh-syntax-highlighting zsh-autosuggestions \
r-base-core  zathura qutebrowser firefox  -y
```

Run bash shell script `~/Dropbox/dotfiles/set_up_links.sh` to set up symbolic links (e.g. `ln -s ~/Dropbox/prj ~/prj`). See Appendix 1 below for details.

Make `zsh` the default shell.

```
> chsh -s $(which zsh)
```

Set up the shell (zsh) per the post [link to set up terminal post]

Install `zotero` using software manager and set up syncing (login: rgthomas)

add `vimium` extension to `firefox`

7 Appendix 1. Script to set up links from local Home to Dropbox

`set_up_links.sh`

```
#!/bin/zsh

# since the install process creates a .config directory move it temporarily
mv ~/.config ~/.config.tmp

# create links to hidden files from ~/Dropbox/dotfiles directories
ff=(".zshrc" ".viminfo" ".vimrc" ".local" ".vim" \
    ".vimplugins" ".config" ".Rprofile")
for P in "${ff[@]}"
do
echo "create  a link for Dropbox/dotfiles version of $P in Home"
    ln -v -s "$HOME/Dropbox/dotfiles/$P" "$HOME/$P"
done

# copy the original ".config" files into new linked .config
cp -R ~/config.tmp/* ~/.config

# create new directories (links) for working files from Dropbox
dd=("sandbox" "bin" "docs" "prj" "work" "ssh" "shr")
for P in "${dd[@]}"
do
    echo "create  a link for Dropbox/dotfiles version of echo $P in Home"
    ln -v -s "$HOME/Dropbox/$P" "$HOME/$P"
done
```

Connect to new machine via ssh from mac laptop

First on the new machine (zz)

```
mac> ssh z@10.0.1.196
mac> scp install_apps.sh z@10.0.1.196:~
mac> scp set_up_links.sh z@10.0.1.196:~
```

Possible Shortcut

Install dropbox first. You could wait for `Dropbox` to finish installing or you could use scp to copy and run the two shell scripts: `install_app.sh` and `set_up_links.sh` from `~/Dropbox/dotfiles`. These two shells can run in parallel with `Dropbox` installing.

8 Run peng1.Rmd

bash> sudo pacman -S r bash> R -e "library(rmarkdown); render('peng1.Rmd')"