Arch Linux Installation Guide

Cyril Andreatta, Patrick Walther

Version 1.1, 2017-04-01

Table of Content

1. Check Harddisks 1

2. Download current Arch Linux iso 1

3. Set keyboard layout 1

4. Create Partitions 2

 4.1. Delete current partitions 2

 4.2. Create first partition for `boot` 2

 4.3. Create second partition for `root` 3

5. Make Filesystem 3

6. Mount Partitions 3

7. Optionally set mirrors 4

8. Setup Network connections 4

9. Start installing base system 4

10. Generate fstab File 4

11. Chroot 4

12. Edit pacman Config 5

13. Update and Install packages 5

14. Set Timezone and hardware clock 5

15. Set Locale Settings 5

16. Set Hostname 6

17. Set Root password 6

18. Create a new User 7

 18.1. Let the User do sudo 7

19. Create mkinitramfs 7

20. Install Bootloader 7

 20.1. Systemd-Boot (pka GummiBoot) [recommended] 7

  20.1.1. Update *systemd-boot* after every release 8

 20.2. Grub [only recommended for non UEFI systems] 8

21. Finish and clean-up 8

This is just a small memory helper for our personal use.

To gain deeper knowledge, the fabulous Arch Wiki should be considered.

> ℹ *Explanation for used icons.*
>
> >_ Commands that should literrally being input to the console.
>
> 🗎 Files that should be edited according to the given instructions.

1. Check Harddisks

> ⚠ Choose the disk you are going to format wisely!

>_ List all installed harddisks.

```
lsblk
```

> ℹ In this document `sda1` is used as the boot partition, `sda2` is the root partition. This might be different on each computer, depending on hardware and individual settings.

2. Download current Arch Linux iso

Go to the Arch Linux homepage and townload the boot image to install.

>_ Write downloaded image to USB thumb-drive.

```
sudo dd if=archlinux-2017.04.01-x86_64.iso of=/dev/sdX bs=4M && sync
```

> ⚠ Check again wich drive you are going to overwrite with `dd`. That command overwrites the whole drive and is **not** reversible. If you are sure replace `/dev/sdX`.

3. Set keyboard layout

>_ List all available keyboard maps and optionally filter for your preferred language.

```
ls /usr/share/kbd/keymaps/**/*.map.gz | grep -i ch
```

>_ Load preferred keyboard layout.

```
loadkeys de_CH-latin1
```

4. Create Partitions

>_ Create new partitions with gdisk.

```
gdisk /dev/sda
```

4.1. Delete current partitions

>_

p `Enter`
: display current partitions.

d `Enter`
: delete partition

1 `Enter`
: select partition to delet

d `Enter`
: delete partition (automatically selected if only one)

> *If you have 8GB or less of memory, a swap partition might be a good idea.*
>
> Create a separate partiotion `/dev/sdaX` and set it up for swap.
>
> ```
> mkswap /dev/sdaX
> swapon /dev/sdaX
> ```

4.2. Create first partition for `boot`

>_ Create new partitions with gdisk.

n `Enter`
: create new partition

`Enter`
: start from first free sector

+512M `Enter`
: set size of EFI partition

ef00 `Enter`

set type of partition to EFI

4.3. Create second partition for `root`

>_ Create new partitions with gdisk.

n `Enter`

create new partition

`Enter`

start from first free sector

`Enter`

set size of Linux partition (use full size)

8300 `Enter`

set type of partition to Linux

w `Enter`

write changes to disk

y `Enter`

confirm

5. Make Filesystem

>_ Format the `/boot` partition as FAT32. The `/root` partition can be formated as `btrfs`, `ext4` or whatever you prefer.

```
mkfs.vfat /dev/sda1
mkfs.btrfs /dev/sda2
```

6. Mount Partitions

>_ Mount the `root` partition to folder */mnt* and then create a folder *boot* in it, where you mount the other partition `boot`.

```
mount /dev/sda2 /mnt
mkdir /mnt/boot
mount /dev/sda1 /mnt/boot
```

7. Optionally set mirrors

🗎 You can place servers near your country on top to increase speed.

```
/etc/pacman.d/mirrorlist
```

8. Setup Network connections

Either plugin a Ethernet cable or setup WiFi.

```
wifi_menu
```

> 💡 Check connection
>
> ```
> ping nba.com
> ```

9. Start installing base system

>_

```
pacstrap /mnt base base-devel
```

10. Generate fstab File

>_ Write currently mounted partitions to *fstab* file to make it persistant, then check the content of the created file.

```
genfstab -U /mnt >> /mnt/etc/fstab
cat /mnt/etc/fstab
```

11. Chroot

>_ Change into chroot environment, which is basically you already logged in to the system you are going to create. Everything you do or install here is going to be in your final installation.

```
arch-chroot /mnt
```

12. Edit pacman Config

📄 Uncomment `[multilib]` repo and line below.

```
/etc/pacman.conf
```

> 💡 *In vim you can search with `/`, then go to beginning of line `0` and clear one character `x`. Then go one line down `j` and do the same.*
>
> ```
> /[multi
> 0x
> j0x
> ```

> 💡 *Under Misc Options set or uncomment following options.*
>
> ```
> Color
> CheckSpace
> ILoveCandy
> ```

13. Update and Install packages

>_

```
 pacman -Syu
 pacman -S vim git sudo dialog wpa_supplicant iw wget efibootmgr intel-ucode
```

14. Set Timezone and hardware clock

>_

```
ln -sf /usr/share/zoneinfo/Europe/Zurich /etc/localtime
hwclock --systohc --utc
```

15. Set Locale Settings

📄 Uncomment additional needed localalizations, `en_US` should already be uncommented.

```
/etc/locale.gen
```

💡

> *Search for `de_CH` delete `#` and save file in vim.*
>
> ```
> /de_CH
> 0
> x
> ZZ
> ```

>_ Activate locales.

```
locale-gen
```

🗎 Set main language of system to `de_CH` and set some system wide settings.

```
/etc/locale.conf
```

```
LANG=de_CH.UTF-8
LC_PAPER=a4
```

🗎 Set input language for virtual console (before X is started).

```
/etc/vconsole.conf
```

```
KEYMAP=de_CH-latin1
```

16. Set Hostname

🗎 Give your computer a cool name.

```
/etc/hostname
```

17. Set Root password

>_ Set new password when promted, then enter it again.

```
passwd
```

18. Create a new User

>_

```
useradd -mg users -G wheel,optical,lock,uucp <NEW_USER>
passwd <NEW_USER>
```

18.1. Let the User do sudo

>_ Open file and uncomment line `%wheel ALL=(ALL) ALL`.

```
visudo
```

19. Create mkinitramfs

>_ Run mkinitcpio to generate 🐧 Linux image.

```
mkinitcpio -p linux
```

20. Install Bootloader

20.1. Systemd-Boot (pka GummiBoot) [recommended]

Arch Wiki >_ Check if EFI is mounted and install *systemd-boot*.

```
efivar -l
bootctl --path=/boot install
```

🗎 Create a file for each image (Linux system) you want to be able to start.

```
/boot/loader/entries/arch.conf
```

This file should contain following lines, while the **PARTUUID** should be adapted to your hardware.

```
title       Arch Linux
linux       /vmlinuz-linux
initrd      /intel-ucode.img
initrd      /initramfs-linux.img
options     root=PARTUUID=XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX rw
```

> 💡 *Get PARTUUID from first harddisk sda1*
>
> ```
> l -l /dev/disk/by-partuuid | grep sda1 | cut -d' ' -f9
> ```

20.1.1. Update *systemd-boot* after every release

>_

```
bootctl --path=/boot update
```

> 💡 *Alternatively you can install a package from AUR which adds a Pacman hook , that updates systemd-boot automatically.*
>
> ```
> packer systemd-boot-pacman-hook
> ```

20.2. Grub [only recommended for non UEFI systems]

>_ Install Grub.

```
sudo pacman -S grub
```

>_ Select harddisk to install grub to. **No** seperate partition needed!

```
grub-install --target=i386-pc /dev/sda
```

21. Finish and clean-up

>_

```
exit
umount -a
reboot
```