

Installing Arch Linux alongside Windows 10

Created with L^AT_EX using T_EXmaker

August 25, 2020

Since the dual boot installation of Arch Linux is often difficult and confusing, I put together an instructional pdf designed to make the arch linux installation as easy as possible. Many are discouraged by the difficulty of installing Arch Linux, and they miss out on its potential ease of use, flexibility, and customization options it has to offer. By the end of this tutorial you should have a working dual boot machine.

Technical Specifications

- Arch Linux version: 2019.12.01
- Included Kernel: 5.3.13
- Windows 10 Version 1803
- Tested using Dell Precision 5520 laptop
- Initially released on 12/12/2019

How this tutorial works

- All code to be executed/edited will be displayed in a box
- This guide is for EFI systems (modern computers)
- There are 8 chapters to this tutorial, separated for an easier read
- If you follow all steps you will have a working Arch install and boot menu

Prerequisites

First you will have to take care of some housekeeping things inside windows before we start.

Create a free partition in Windows to make room for Arch:

To do this go to Disk Manager in windows and locate the drive you want to dual boot from. This will require the drive to have free space available. Go ahead and right click the volume partition you want and select shrink partition. It will ask how much you want to shrink, select the amount of space you want the arch install to have for this. Next you will have to format this volume for it to show up in Arch Linux. Go ahead and right click on the new partition and select make new simple volume and format with NTFS.

Disabling Windows Fast-Startup:

You can disable this by going to: control panel→power options→choose what the power buttons do

Disabling Secure Boot:

On Windows 10, hold the Shift key while selecting restart. Go to Troubleshoot → Advanced Options → UEFI Firmware Settings. This will bring you to your computers BIOS, secure boot will most likely already be disabled so you shouldn't have to do anything here. Restart the computer to make sure any changes take effect.

Great! Now we are ready to start making the bootable USB.

Creating the Arch Linux boot partition with Rufus

This tutorial will use Rufus to create a bootable Arch Linux install using an ISO image. Go ahead and download the latest version of Rufus. Also download the Arch Linux ISO file. Note the ISO will download as a torrent, so you will need software capable of opening a torrent file.

You can download the Arch Linux ISO from this link:

<https://www.archlinux.org/download/>

And download Rufus from this link:

<https://rufus.ie/>

Setting up Rufus

Launch Rufus and select these options to make the correct boot drive:

Device → USB Device you will be using

Boot selection → navigate to directory of downloaded ISO

Partition Scheme → MBR

Target System → BIOS or UEFI

Volume Label → whatever you want

File System → Fat32

Cluster size → 4096 (default)

Now go ahead and press start. The process may take a few minutes to complete, then close Rufus. You now have a bootable Arch linux USB.

The Installation

Go ahead and plug the USB into the computer and boot into your BIOS. The button you have to press will differ with different PC's. Change the boot options in order to boot the USB first. Once it boots you should be presented with a welcome screen, then a terminal provided after a few seconds.

Create partitions using cgdisk

Make sure you have allocated a free partition via Disk Manager in windows prior to doing this part. Also make sure to take note of the partition names that you will create. You will need them later.

Create the swap partition:

Go ahead and use cgdisk. Remember, boxes contain commands to be executed.

```
cgdisk
```

Then do the following:

Select the free space partition, then select new
Hit enter to select the default option for the first sector.
Select your desired swap file size, for 2GB enter 2G
Set swap name. Hit enter.

Create the root partition:

Do the following:

Select the free space partition, Hit New again.
Hit enter to select the default option for the first sector.
Hit enter again to use the remainder of the disk.
Also hit enter for the GUID to select default.
Then set name of the partition of root.

Hit write at the bottom of the GUI to write changes to the disk. Enter "yes" to confirm the command.
We are now done partitioning the disk. Select Quit to exit the cgdisk GUI.

Go ahead and type this command to view the new partition table:

```
lsblk
```

Make sure both of the partitions you created are listed.

Miscellaneous:

Now enable the swap partition. Replace sdax with the partition you use.

```
mkswap -L "Linux Swap" /dev/sdax
```

then do:

```
swapon /dev/sdax
```

Verify swap is activated:

```
free -m
```

Format root partition as ext4:

```
mkfs.ext4 /dev/sdax
```

Mount the root partition:

```
mkdir /mnt
```

then do:

```
mount /dev/sdax /mnt
```

For connecting to a WiFi network, use the command below. A GUI will appear and ask you to select a wireless network, then enter your wifi password.

```
wifi-menu
```

Install the base components

```
pacstrap /mnt base
```

Mount the EFI Partition

```
mkdir -p /mnt/boot/efi
```

Next, check the partition table. You will need to take note of the EFI partition number here.

```
gdisk -l /dev/sdx
```

For example, if the EFI partition number is 2... then do this.

```
mount /dev/sda2 /mnt/boot/efi
```

Generate fstab and chroot into the system

```
genfstab -p /mnt >> /mnt/etc/fstab
```

Check the fstab

```
cat /mnt/etc/fstab
```

Chroot into system now

```
arch-chroot /mnt
```

Setting Locale

The Locale defines which language the system uses, and other regional considerations such as currency denomination, numerology, and character sets. First download nano, a text editor.

```
sudo pacman -S nano
```

Now open the the file below using nano and uncomment your language. Save the file when done by using CTRL-O to write and CTRL-X to quit.

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

Now generate new locales:

```
locale-gen
```

```
echo 'LANG=en_US.UTF-8' > /etc/locale.conf
```

```
export LANG=en_US.UTF-8
```

Now set the time. Indianapolis, Indiana used for example:

```
ln -s /usr/share/zoneinfo/America/Indiana/Indianapolis
```

It is recommended to adjust the time skew, and set the time standard to UTC:

```
hwclock --systohc --utc
```

Set the hostname:

```
echo 'arch' > /etc/hostname
```

For wifi install:

```
pacman -S iw wpa_supplicant dialog
```

Enable multilib and AUR repositories in `/etc/pacman.conf`:

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

Uncomment the lines starting with `[multilib]`

Add the following lines at the end of `/etc/pacman.conf`:

```
[archlinuxfr]
SigLevel = Never
Server = http://repo.archlinux.fr/$arch
```

Download two packages before using `mkinitcpio`

```
sudo pacman -S linux linux-firmware
```

Generate the `initramfs` image:

```
mkinitcpio -p linux
```

Change the root password:

```
passwd root
```

Then add user with your name. Example provided with name Benjamin and username ben.

```
useradd -m -g users -G wheel,storage,power,video -s /bin/bash -c "Benjamin" ben
```

Set password for the ben user.

```
passwd ben
```

Download vim text editor, and add new user to sudoers after entering visudo:

```
sudo pacman -S vim
```

```
visudo
```

Uncomment `# %wheel ALL=(ALL) ALL`

Grub bootloader Installation

Download needed packages:

```
pacman -Syu grub efibootmgr
```

Install grub to your drive. For example, my partitions are on sda drive so I would type sda. NOT sda1, sda2... etc.

```
grub-install /dev/sdx
```

Generate grub config. You will not see Windows listed here when executed, this is normal.

```
grub-mkconfig -o /boot/grub/grub.cfg
```

Verify if grub installed, type:

```
ls -l /boot/efi/EFI/arch/
```

Graphical environment installation

This section will provide installation packages for either GNOME or KDE PLASMA to choose from.

If you are using Gnome, here are the packages needed:

```
pacman -S gnome-system-monitor gnome-shell nautilus gnome-terminal gnome-tweak-tool
```

```
gnome-control-center xdg-user-dirs networkmanager gnome-keyring network-manager-applet xorg gdm
```

If you are using KDE Plasma, here are the packages needed:

```
pacman -S xdg-user-dirs networkmanager network-manager-applet xorg sddm plasma kde-applications
```

If you are using GNOME:

```
systemctl enable gdm
```

If you are using KDE Plasma:

```
systemctl enable sddm
```

Enable networking:

```
systemctl enable NetworkManager
```

You can now reboot into your Arch Install, note that the boot manager will not work yet. You will have to make sure the Arch install is set to boot first instead of the USB. Go ahead and exit chroot and reboot.

```
exit
```

```
reboot
```

Once you are booted into Arch linux, you will be presented with a login screen. Go ahead and enter your username and password to enter. Enter a terminal and download os-prober.

```
sudo pacman -Syu os-prober
```

Then generate the grub config. When this executes you should see it find windows as an option.

```
grub-mkconfig -o /boot/grub/grub.cfg
```

Reboot and you will see that you now have an option to boot windows 10 or Arch Linux! You are all set!

Sources used for Article

A lot of information was used from these two sources, some information was outdated and had to be fixed for this tutorial to work.

[http://www.abhishekan.me/linux/arch/2019/04/02/Dual-Boot-Arch-\\Linux-With-Windows-\(UEFI\).html](http://www.abhishekan.me/linux/arch/2019/04/02/Dual-Boot-Arch-\\Linux-With-Windows-(UEFI).html)

https://wiki.archlinux.org/index.php/Installation_guide