Arch Install Instructions

Colin Murphy

April 14, 2015

Introduction

I am an Electrical and Computer Engineering major at the University of Rochester. I have been using Linux for a little over a year and have taken a liking to Arch. Although it is more involved to install and configure than many distributions, it provides everything you need to make it exactly what you want. With that in mind, I have detailed the installation process I use for my Arch machines. A lot of these steps are necessary and others are more personal preference. I hope you find this helpful and please let me know if you have any suggestions for improvement.

Acknowledgements

The following people have greatly helped me in editing and updating this document:

- Nick Graham
- Frank Tamburrino

Making bootable

If you're running unix system already you can just use the following command (THIS WILL REMOVE ALL DATA FROM SDX):

```
sudo dd if=/path_to_arch_.iso of=/dev/sdX
```

Otherwise, use UNetbootin.

Internet

Ping to make sure you are connected to the internet.

```
ping -c 3 google.com
```

If the ping is unsuccessful, try to establish a connection

```
dhcpcd
```

Partitioning

Choosing between fdisk (MBR) and gdisk (GPT):

- If using GRUB legacy as the bootloader, one must use MBR.
- To dual-boot with Windows (both 32-bit and 64-bit) using Legacy BIOS, one must use MBR.
- To dual-boot Windows 64-bit using UEFI instead of BIOS, one must use GPT.
- To dual boot with OS X, it is recommended that you use GPT, unless you plan to triple boot with Windows.
- If none of the above apply, choose freely between GPT and MBR; since GPT is more modern, it is recommended in this case.
- It is recommended to use always GPT for UEFI boot as some UEFI firmwares do not allow UEFI-MBR boot

Partition Scheme

The partition scheme you use is a matter of personal preference. The simplest is:

- Boot partition
- Swap partition
- Root partition

Adding a discrete home partition offers a few notable advantages; This can allow you to change distributions or reinstall while keeping your home folder intact as well as let you share you home folder among multiple installations. On more modern systems, the swap partition is less critical, being used for data overflow from RAM. Inclusion of a swap partition is left to the discretion of the reader. Typical sizes for each type of partition can be seen in Figure 1.

Table 1: Typical Partition Sizes

Boot	200MB
Swap	2GB
Root	8GB
Home	Rest of drive

Creating File Systems

Now that you have your partitions you need to format them. You can leave the boot partition as is but home and root folders need to be ext4. To do this, use the following command where sdx is the drive you've partition and # is the partition number:

```
mkfs.ext4 /dev/sdx#
```

For swap, you just need to mark it as swap and turn it on:

```
mkswap /dev/sdx# && swapon /dev/sdx#
```

Mount Partitions

Mount you root partition in /mnt:

```
mount /dev/sdx# /mnt
```

then make your home directory and mount your home partition:

```
mkdir /mnt/home && mount /dev/sdx#
```

Installing Base Packages

To make downloading the base quicker, modify /etc/pacman.d/mirrorlist with nano (the default text editor included with arch) by copying the location closest to you at the top. You can search using Ctrl+w, copy using Alt+6 and paste using Ctrl+u. Then install the base packages:

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

Configuring File Systems Table

Generate an fstab file with the command:

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

Once you have done this you can chroot into the system to configure it:

```
arch-chroot /mnt
```

Configure Language and Locale

Edit /etc/locale.gen and look for your language and uncomment it. Since you're reading this in English, you probably want en_US.UTF-8 UTF-8. To finish configuring language and locale, run:

```
locale-gen
```

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

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

Time Zone

If you do not know the name of your sub-time zone you can run the following to see a list of them:

```
ls /usr/share/zoneinfo/
```

Next, symbolically link your sub-time zone to localtime and configure the hardware clock:

```
ln -s /usr/share/zoneinfo/America/New_York> /etc/localtime
```

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

Hostname

Making your hostname is simple; just run:

```
echo your_hostname > /etc/hostname
```

Configuring Multilib

If you are running a 64-bit system you should include the multilib repository for pacman. This will add some libraries 32-bit applications may need to run on your system. In /etc/pacman.conf, uncomment:

```
[multilib]
```

```
Include = /etc/pacman.d/mirrorlist
```

Then run the following to update your package list:

```
pacman -Sy
```

Users

You should definitely set the passsword for root which you can do using:

```
passwd
```

Next you can add users:

```
useradd -m -g users -G wheel,storage,power -s /bin/bash your_username
```

Then make the password for a user:

```
passwd user_name
```

Install sudo and bash-completion:

```
pacman -S sudo bash-completion
```

In order to allow users in wheel to use sudo:

```
EDITOR=nano visudo
```

Then uncomment:

```
%wheel ALL=(ALL) ALL
```

Installing GRUB

Install grub and os-prober (os-prober is useful is you have multiple OSes):

```
pacman -S grub os-prober
```

```
grub-install --target=i386-pc --recheck /dev/sdx
```

If you do have multiple OSes, run the following to update your grub configuration:

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

Enable Network

Find the name of your LAN adapter which will be listed in:

```
ip link
```

Then enable it so it will start on startup:

```
systemctl enable dhcpcd@name_of_interface.service
```

Unmount and reboot

Now you can exit, unmount and reboot:

```
exit
```

```
umount -R /mnt
```

```
reboot
```

Depending on your boot order, you may want to remove the USB so you boot into your new system rather than the USB.

Install X

Now we need to install X:

```
pacman -S xorg-server xorg-server-utils xorg-xinit mesa
```

Then install the appropriate graphics drivers. This can get hairy and you will probably have to do some research to find the correct ones. Running the following can help when trying to determine which drivers you need, especially if you aren't sure what kind of graphics card you have:

```
lspci | grep VGA
```

A few common devices are listed in Table 2 along with the driver associated with them as well as the multilib package.

Table 2: Common Graphics Devices

Device	Type	Driver	Multilib Driver
AMD/ATI	Open Source	xf86-video-ati	lib32-ati-dri
	Proprietary	catalyst-dkms	lib32-catalyst-utils
Intel	Open Source	xf86-video-intel	lib32-intel-dri
Nvidia	Open Source	xf86-video-nouveau	lib32-nouveau-dri
		xf86-video-nv	———
	Proprietary	nvidia	lib32-nvidia-libgl
		nvidia-304xx	lib32-nvidia-304xx-utils
		nvidia-173xx	lib32-nvidia-173xx-utils
		nvidia-96xx	lib32-nvidia-96xx-utils
VIA	Open Source	xf86-video-openchrome	———

If you are using a laptop you can install the synaptics package that lets you use input devices like your track pad:

```
pacman -S xf86-input-synaptics
```

Let's finish installing X and test it to make sure it's working:

```
pacman -S xorg-twm xorg-xclock xterm
```

```
startx
```

You should see a few terminal windows and a clock. Type exit to exit.

Desktop Environment

Choosing a desktop environment is completely up to each user to decide for themselves; I personally like Cinnamon so I will be using that as an example.

First, install the desktop manager associated with your preferred environment. In my case this is gdm. I then install the Cinnamon desktop environment.

```
sudo pacman -S gdm
```

```
sudo pacman -S cinnamon
```

It's a good idea to test the manager and environment before enabling them using systemcl:

```
sudo systemctl start gdm
```

If you have no issues you should see a login screen in your selected desktop manager. If that went well, enable it on startup:

```
sudo systemctl enable gdm
```

Network Manager

One final thing to configure is your network manager:

```
sudo systemctl enable NetworkManager
```

```
sudo systemctl start NetworkManager
```

If you have issues upon rebooting, you can try the following command then reboot:

```
systemctl disable dhcpcd.service
```

AUR

As a final note I would like to quickly mention how to install packages from the Arch Unofficial Repositories which has a huge variety of software not available in the official repositories. It is a great resources and very well supported. As an example I will use google chrome. Once you have found the page on the AUR, click "Download tarball". Navigate to where the tarball downloaded then run:

```
tar -xvf google-chrome.tar.gz
```

This will create a folder called google-chrome. Go into that folder and type:

```
makepkg
```

You may run into dependencies so if you do just install those through pacman. Otherwise you should see a file in that folder with a .tar.xz file extension. To install the package, just type:

```
sudo pacman -U google-chrome-39.0.2171.95-1-x86_64.pkg.tar.xz
```

Yaourt

You can also use a wrapper such as yaourt to handle the installation of AUR packages. Wrappers such as this handle dependencies. To install Yaourt you must first install package-query. This is done using the method described above.

```
curl -O https://aur.archlinux.org/packages/pa/package-query/package-query.tar.gz

tar zxvf package-query.tar.gz

cd package-query

makepkg -si
```

Then install yaourt

```
curl -O https://aur.archlinux.org/packages/ya/yaourt/yaourt.tar.gz

tar zxvf yaourt.tar.gz

cd yaourt

makepkg -si
```

To use yaourt simply type

```
yaourt <package name>
```

Otherwise the syntax is the same as pacman.