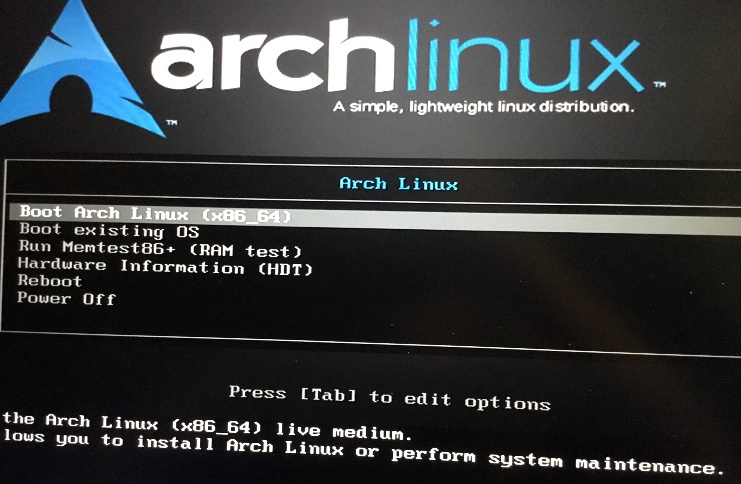
# 制作介质

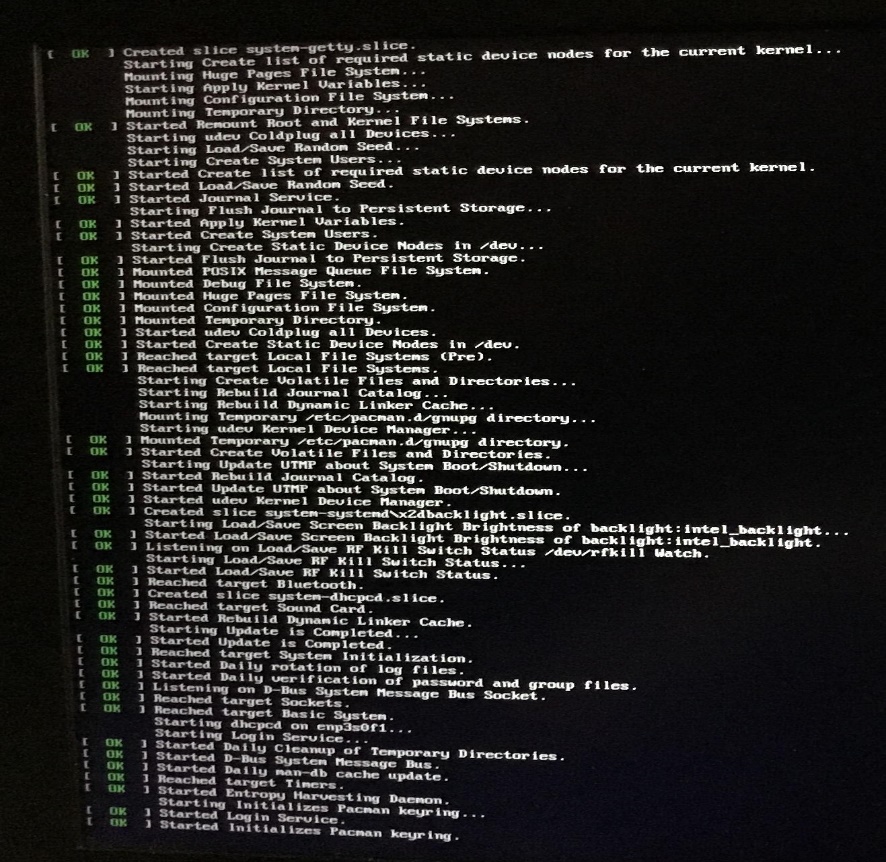
制作U盘

如果是windows系统下制作安装介质，推荐使用usbwriter这款轻量级的工具，下载链接：<https://sourceforge.net/projects/usbwriter/>

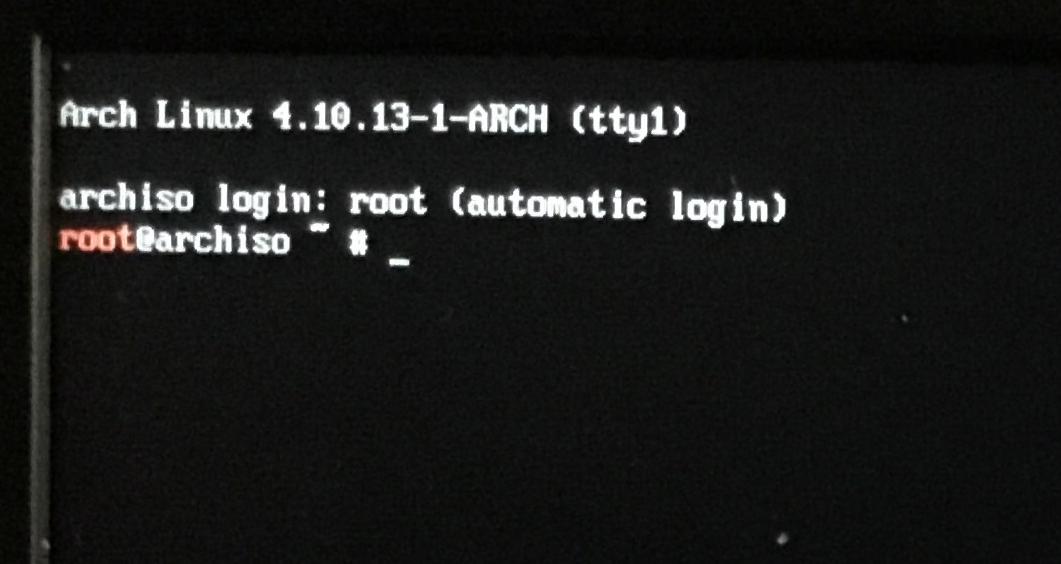
# 进入arch安装界面



选择第一项。Arch开始加载



加载完成后你将会进入一个有命令提示符的界面：



# 检查引导方式

如果你不知道自己的引导方式，请在命令提示符下执行以下命令：

ls /sys/firmware/efi/efivars

输入命令并回车执行后，如果提示

ls: cannot access '/sys/firmware/efi/efivars': No such file or directory

表明你是以BIOS方式引导，否则为以EFI方式引导。

# 联网

## WiFi

请执行以下命令：wifi-menu

## 有线网

如果你是有线网并且路由器支持DHCP的话插上网线后先执行以下命令获取IP地址：

Dhcpcd

然后执行以下命令来判断网络连接是否正常：

ping [www.baidu.com](http://www.baidu.com)

# 更新系统时间

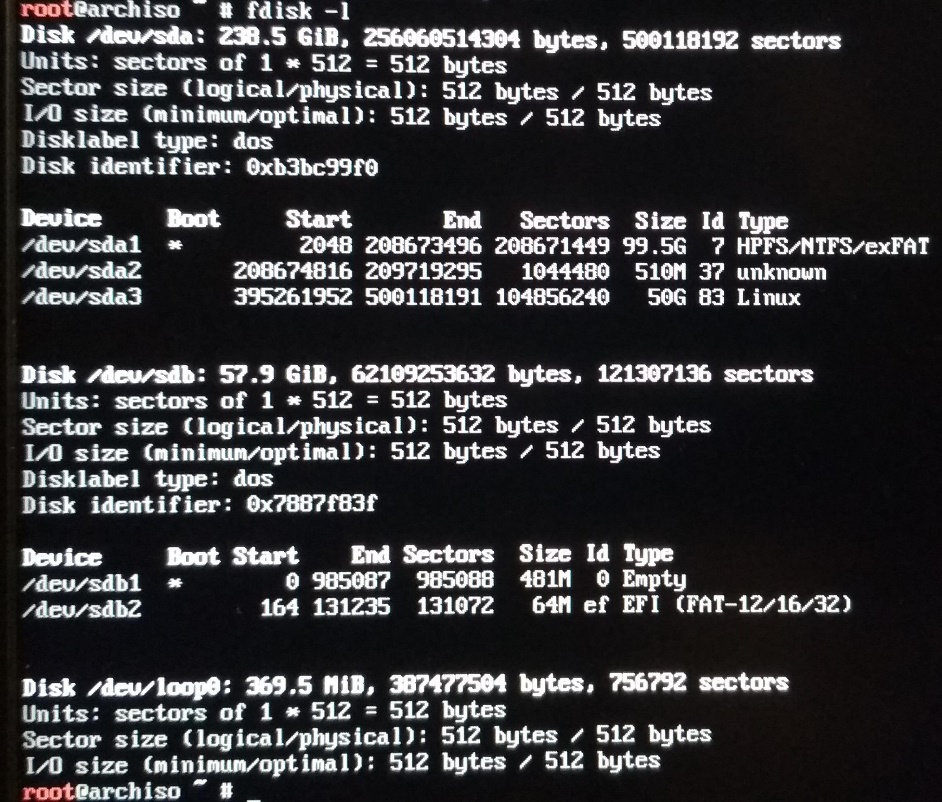
timedatectl set-ntp true

正常情况下这样的命令并没有输出，所谓没有消息就是最好的消息

# 分区与格式化

## 查看目前分区情况

fdisk –l



可以看到我的一块238.5g的硬盘(/dev/sda就代表这块硬盘)，下面列出了/dev/sda\*这三个分区，/dev/sda3是我存活下来的家目录，可以看到它的类型为Linux分区。注意看Start与End的数值，这个数值代表扇区号，可以理解成硬盘被划分成了一个个小单元，可以直观地看出来在/dev/sda2的End与/dev/sda3的Start之间空出了一大块未分配的空间，接下来我们将分配这块区域。

* 如果你是BIOS/MBR方式引导，**跳过下面创建一个引导分区**的步骤。
* 如果你是EFI/GPT方式引导，并且同时安装了其他系统，那么你应该可以在分区列表中发现一个较小的并且类型为EFI的分区（**注意查看硬盘的大小，这个EFI分区有可能是你U盘中的，需要排除**），这是你的引导分区，请记下它的路径（/dev/sdxY)备用，**跳过下面创建一个引导分区**的步骤。
* 如果你是EFI/GPT方式引导，但是没有这个较小的并且类型为EFI的引导分区（这种情况一般只会出现在新的硬盘），那么你需要**先创建一个引导分区**。

## 创建一个引导分区

UEFI方式必须

执行命令：

fdisk /dev/sdx （请将sdx替换成你要操作的磁盘如sdb sdc等）

下面你就进入了fdisk的操作环境， 输入m并回车可以查看各命令的作用。

1. **如果你是一块全新的硬盘**：输入g来创建一个全新的gpt分区表，**否则直接进行第2步**。
2. 输入n创建一个新的分区，首先会让你选择起始扇区，一般直接回车使用默认数值即可，然后可以输入结束扇区或是分区大小，这里我们输入+512M来创建一个512M的引导分区。
3. 这时我们可以输入p来查看新创建的分区。
4. 输入t并选择新创建的分区序号来更改分区的类型，输入l可以查看所有支持的类型，输入ef更改分区的类型为EFI。
5. 输入w来将之前所有的操作写入磁盘生效，在这之前可以输入p来确认自己的分区表没有错误。

输入以下命令来格式化刚刚创建的引导分区：

|  |  |
| --- | --- |
| 1 | mkfs.fat -F32 /dev/sdxY （请将sdxY替换为刚创建的分区） |

现在引导分区就创建好了。

## 创建根分区

输入命令：

|  |  |
| --- | --- |
| 1 | fdisk /dev/sdx （请将sdx替换成你要操作的磁盘如sdb sdc等） |

1. **如果你是一块全新的硬盘**（**否则直接进行第2步**）：

**如果你是BIOS/MBR引导方式**：输入o来创建一个全新的MBR分区表。

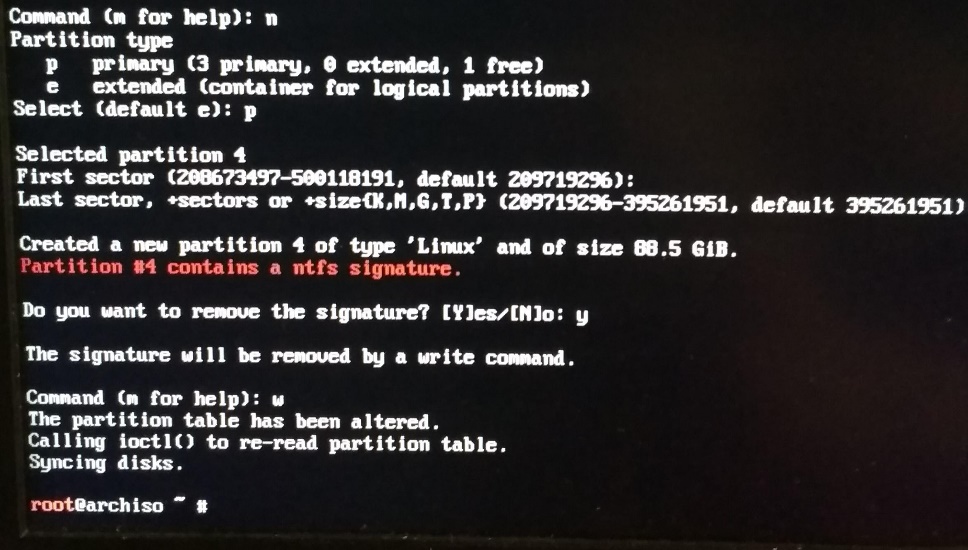
**如果你在上一步新建了分区表并创建了引导分区**：直接进行步骤2。

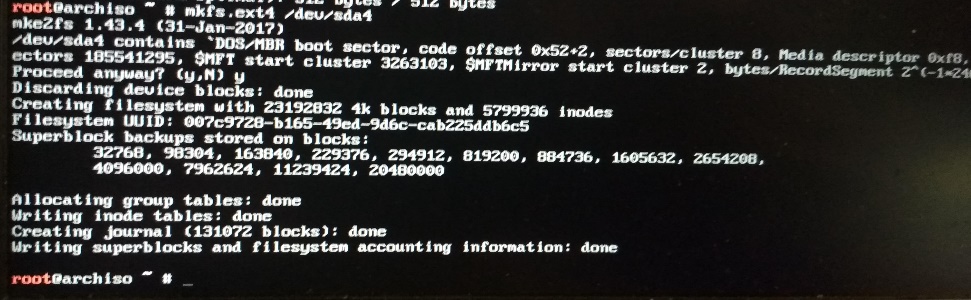
**如果你在另一块硬盘中已经有引导分区**：输入g来创建一个全新的gpt分区表。

1. 输入n创建一个新的分区，首先会让你选择起始扇区，一般直接回车使用默认数值即可，然后可以输入结束扇区或是分区大小，如果我们想要使创建的分区完全占满空闲的空间，可以直接回车使用默认结束扇区。
2. 这时我们可以输入p来查看新创建的分区。
3. 输入w来将之前所有的操作写入磁盘生效，在这之前可以输入p来确认自己的分区表没有错误。
4. 输入以下命令来格式化刚刚创建的根分区：

|  |  |
| --- | --- |
| 1 | mkfs.ext4 /dev/sdxY （请将的sdxY替换为刚创建的分区） |

一个分区参考的例子





# 挂载分区

执行以下命令将根分区挂载到/mnt：

mount /dev/sdxY /mnt （请将sdxY替换为之前创建的根分区）

**如果你是EFI/GPT引导方式**，执行以下命令创建/boot文件夹并将引导分区挂载到上面。**BIOS/MBR引导方式无需进行这步。**

|  |  |
| --- | --- |
| 1  2 | mkdir /mnt/boot  mount /dev/sdxY /mnt/boot （请将sdxY替换为之前创建或是已经存在的引导分区） |

# 选择镜像源

vim /etc/pacman.d/mirrorlist

找到标有China的镜像源，normal模式下按下dd可以剪切光标下的行，按gg回到文件首，按P（注意是大写的）将行粘贴到文件最前面的位置（优先级最高）。

推荐使用清华、浙大源：

Server = http://mirrors.tuna.tsinghua.edu.cn/archlinux/$repo/os/$arch

Server = http://mirrors.zju.edu.cn/archlinux/$repo/os/$arch

用:wq命令保存文件并退出。

# 安装基本包

下面就要安装最基本的ArchLinux包到磁盘上了。这是一个联网下载并安装的过程。

pacstrap /mnt base base-devel

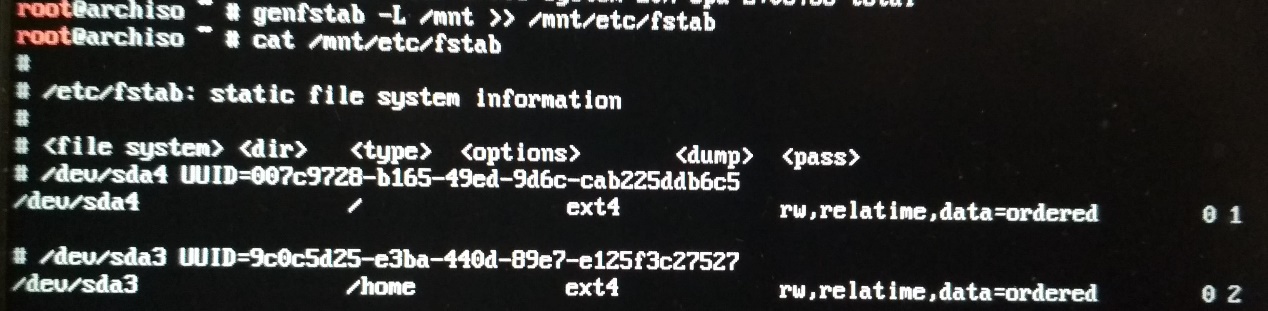
# 配置Fstab

生成自动挂载分区的fstab文件，执行以下命令：

genfstab -L /mnt >> /mnt/etc/fstab

由于这步比较重要，所以我们需要输出生成的文件来检查是否正确，执行以下命令：

cat /mnt/etc/fstab



如图，可以看到/dev/sda4被挂载到了根分区。**如果是EFI/GPT引导的还应该有引导分区被挂载到/boot目录**。

## Fstab文件格式

当系统启动的时候，系统会自动地从这个文件读取信息，并且会自动将此文件中指定的文件系统挂载到指定的目录。

 在这个文件下，我们要关注的是它的六个域，分别为：<file system>、<mount point>、<type> 、<options>、<dump>、<pass>。

1. <fie sysytem>。这里用来指定你要挂载的文件系统的设备名称或块信息，也可以是远程的文件系统。

  如果想把本机上的某个设备（device）挂载上来，写法如：/dev/sda1、/dev/hda2或/dev/cdrom，其中，/dev/sda1 表示第一个串口硬盘的第一个分区，也可以是第一个SCSI硬盘的第一个分区，/dev/hda1表示第一个IDE硬盘的第一个分区，/dev/cdrom 表示光驱。

1. <mount point>。挂载点，也就是自己找一个或创建一个dir（目录），然后把文件系统<fie sysytem>挂到这个目录上，然后就可以从这个目录中访问要挂载文件系统。对于swap分区，这个域应该填写：none，表示没有挂载点。
2. <type>。这里用来指定文件系统的类型。下面的文件系统都是目前Linux所能支持的：adfs、befs、cifs、ext3、 ext2、ext、iso9660、kafs、minix、msdos、vfat、umsdos、proc、reiserfs、swap、 squashfs、nfs、hpfs、ncpfs、ntfs、affs、ufs。
3. <options>。这里用来填写设置选项，各个选项用逗号隔开。由于选项非常多，而这里篇幅有限，所以不再作详细介绍，如需了解，请用 命令 man mount 来查看。但在这里有个非常重要的关键字需要了解一下：defaults，它代表包含了选项rw,suid,dev,exec,auto,nouser和 async。
4. <dump>。此处为1的话，表示要将整个<fie sysytem>里的内容备份；为0的话，表示不备份。现在很少用到dump这个工具，在这里一般选0。
5. <pass>。这里用来指定如何使用fsck来检查硬盘。如果这里填0，则不检查；挂载点为 / 的（即根分区），必须在这里填写1，其他的都不能填写1。如果有分区填写大于1的话，则在检查完根分区后，接着按填写的数字从小到大依次检查下去。同数字 的同时检查。比如第一和第二个分区填写2，第三和第四个分区填写3，则系统在检查完根分区后，接着同时检查第一和第二个分区，然后再同时检查第三和第四个分区。

# Chroot

Chroot意为Change root，相当于把操纵权交给我们新安装（或已经存在）的Linux系统，**执行了这步以后，我们的操作都相当于在磁盘上新装的系统中进行**。

执行如下命令：

|  |
| --- |
| arch-chroot /mnt |

这里顺便说一下，如果以后我们的系统出现了问题，只要插入U盘并启动， 将我们的系统根分区挂载到了/mnt下（如果有efi分区也要挂载到/mnt/boot下），再通过这条命令就可以进入我们的系统进行修复操作。

# 设置时区

依次执行如下命令设置我们的时区为上海并生成相关文件：

ln -sf /usr/share/zoneinfo/Asia/Shanghai /etc/localtime

hwclock --systohc

# 安装必须的软件包

pacman -S vim dialog wpa\_supplicant ntfs-3g networkmanager

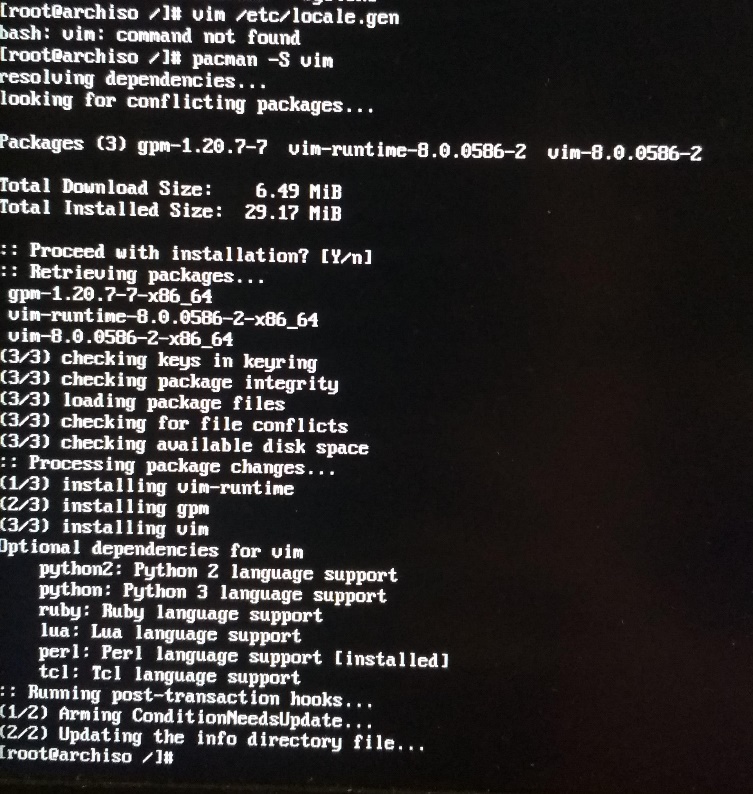
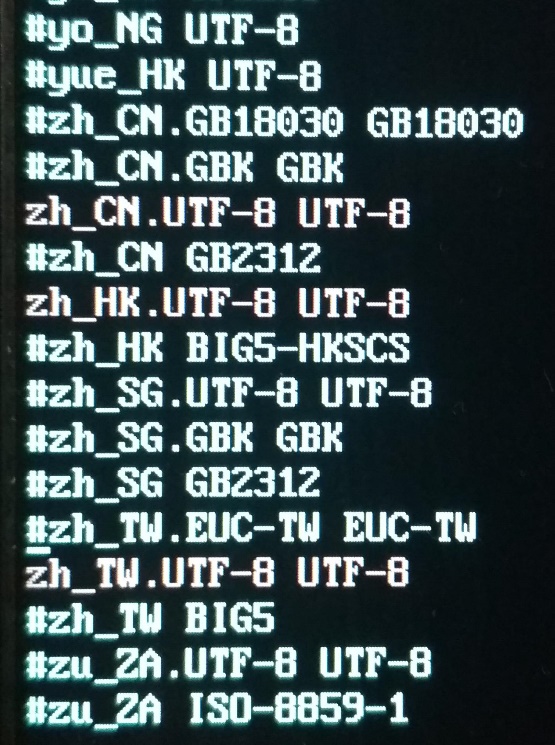


图 13‑1 vim及其依赖项

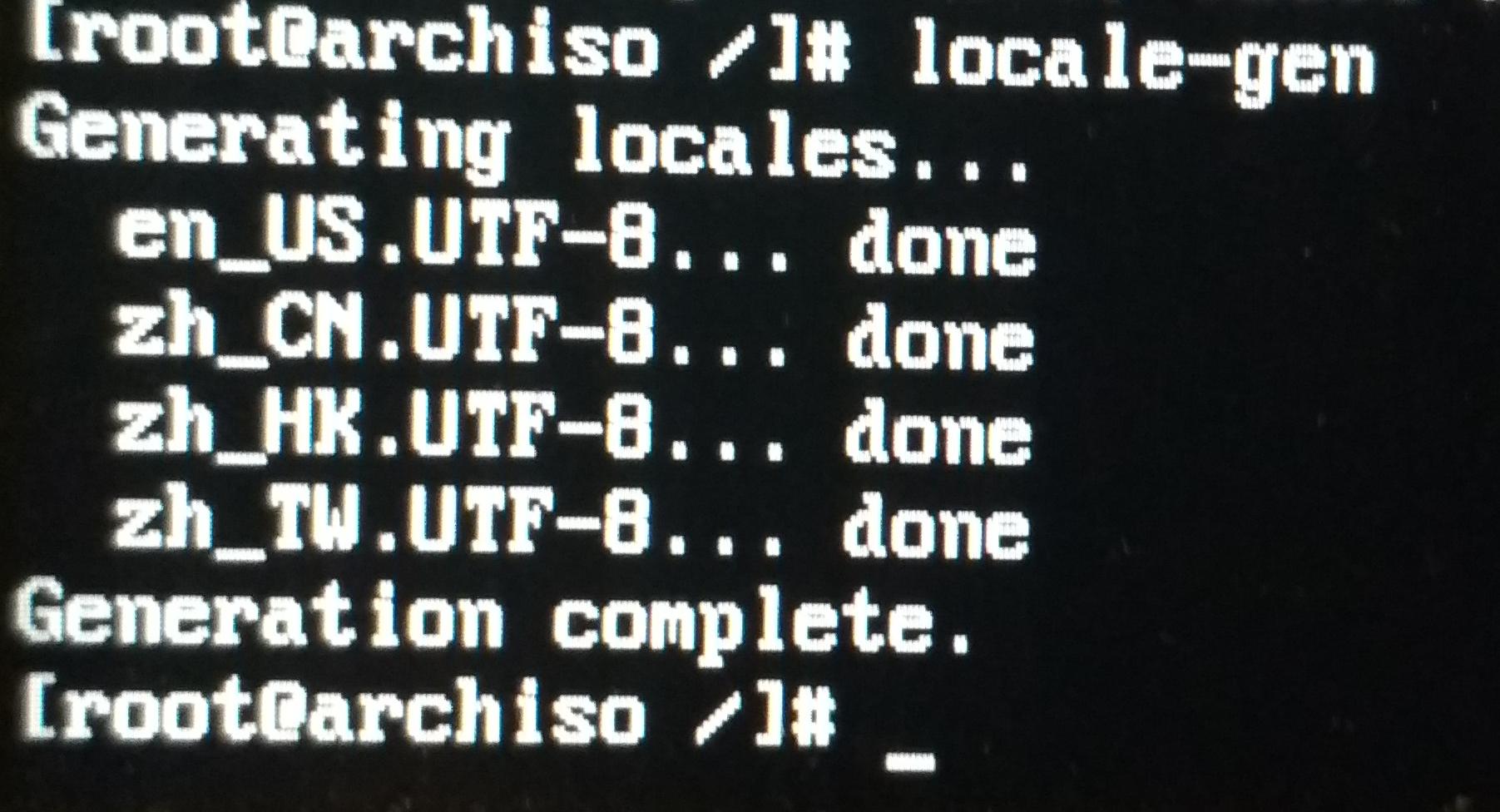
# 设置locale

vim /etc/locale.gen

在文件中找到zh\_CN.UTF-8 UTF-8 zh\_HK.UTF-8 UTF-8 zh\_TW.UTF-8 UTF-8 en\_US.UTF-8 UTF-8这四行，去掉行首的#号，保存并退出。如图：



然后执行locale-gen



打开（不存在时会创建）/etc/locale.conf文件：vim /etc/locale.conf

在文件的第一行加入以下内容：LANG=en\_US.UTF-8

# 设置主机名

vim /etc/hostname

在文件的第一行输入你自己设定的一个myhostname

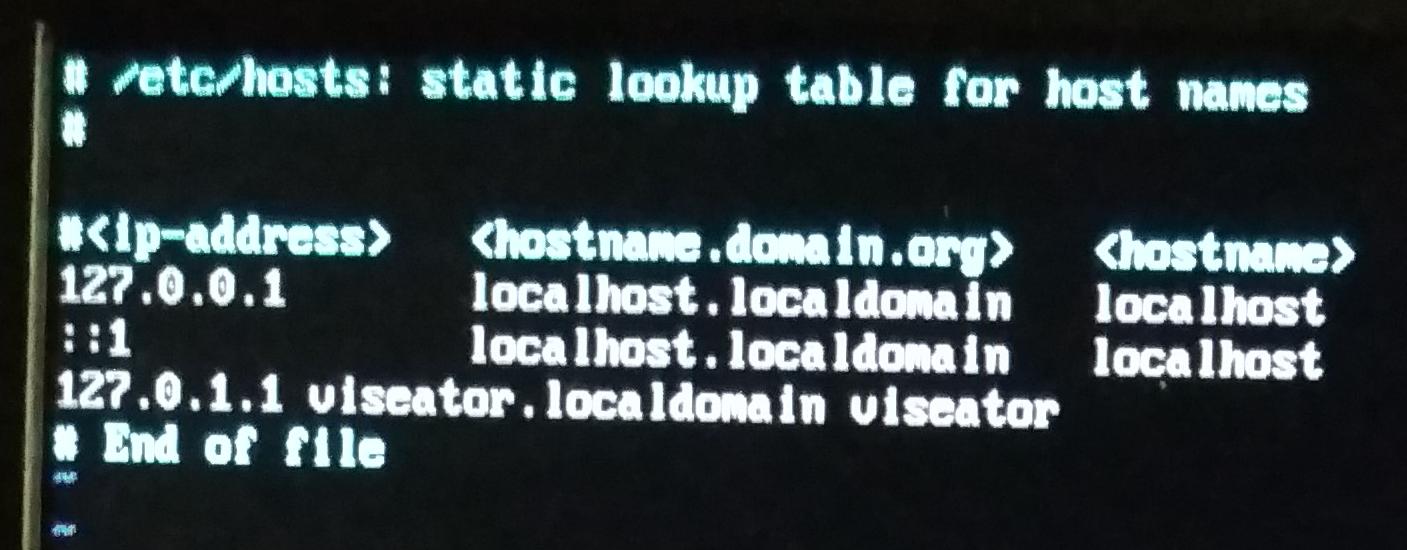
vim /etc/hosts

作如下修改（将myhostname替换成你自己设定的主机名）

127.0.0.1 localhost.localdomain localhost

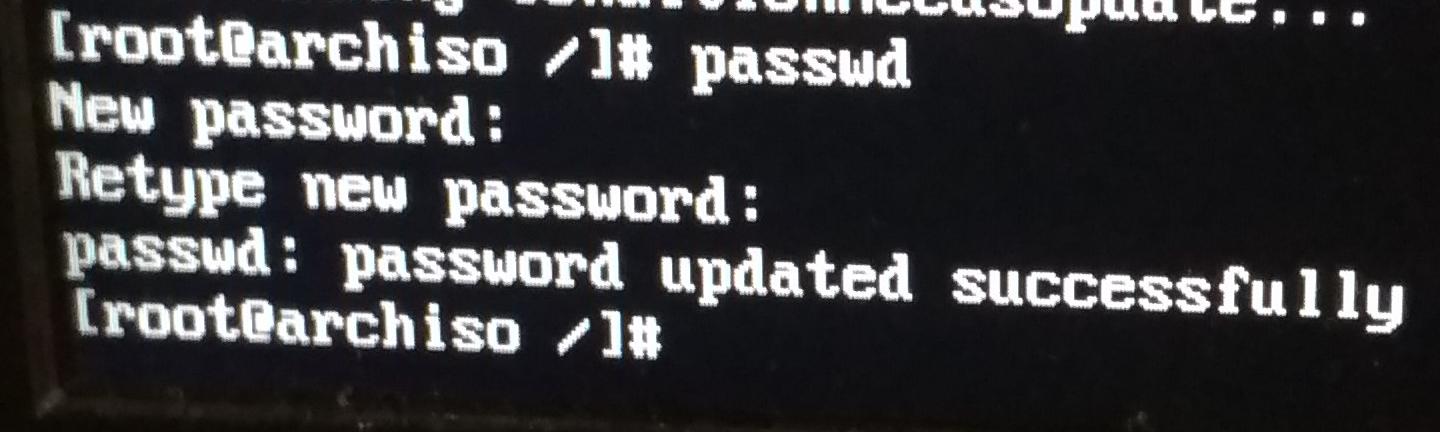
::1 localhost.localdomain localhost

127.0.1.1 myhostname.localdomain myhostname



# 设置root密码

passwd



# 安装Intel-ucode（非IntelCPU可以跳过此步骤）

pacman -S intel-ucode

# 安装bootloader

这里我们安装最流行的Grub2。

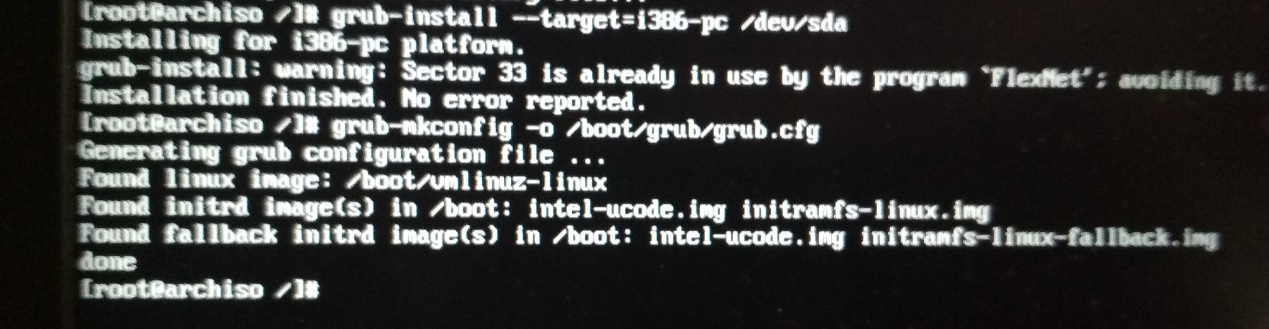
首先安装os-prober这个包，它可以配合Grub检测已经存在的系统，自动设置启动选项。pacman -S os-prober

## ****如果为BIOS/MBR引导方式：****

* 安装grub包：pacman -S grub
* 部署grub：grub-install --target=i386-pc /dev/sdx （将sdx换成你安装的硬盘）

注意这里的sdx应该为硬盘（例如/dev/sda），**而不是**形如/dev/sda1这样的分区。

* 生成配置文件：grub-mkconfig -o /boot/grub/grub.cfg



## **如果为EFI/GPT引导方式：**

* 安装grub与efibootmgr两个包：pacman -S grub efibootmgr
* 部署grub：

grub-install --target=x86\_64-efi --efi-directory=/boot --bootloader-id=grub

* 生成配置文件：grub-mkconfig -o /boot/grub/grub.cfg

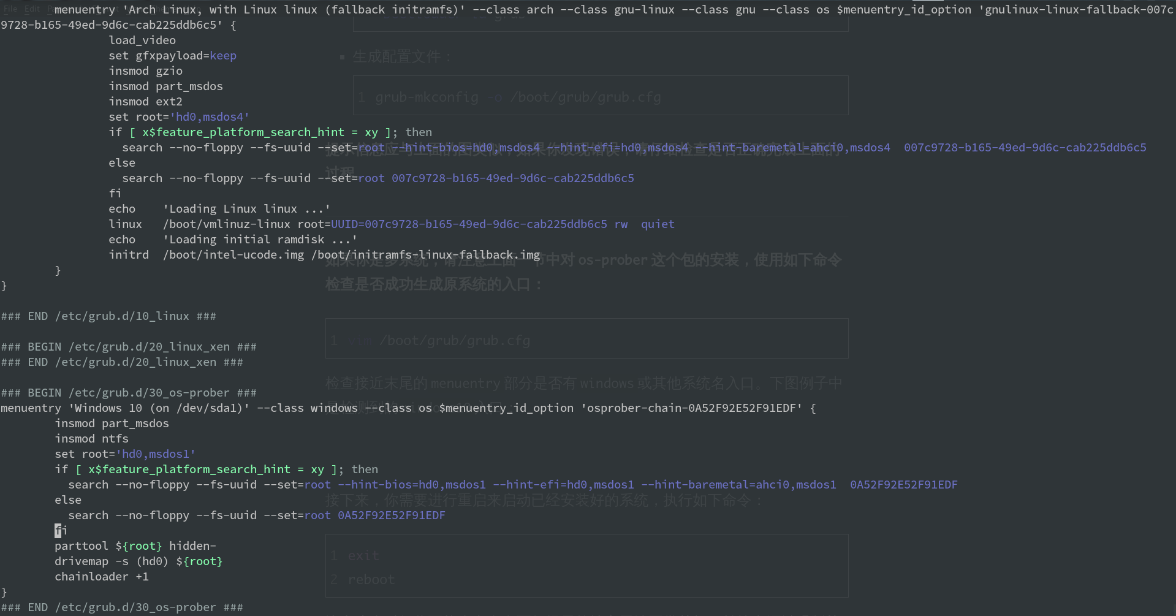
# 安装后检查

如果你是多系统，请注意上面一节中对os-prober这个包的安装。

强烈建议使用如下命令检查是否成功生成各系统的入口，如果没有正常生成会出现开机没有系统入口的情况：

vim /boot/grub/grub.cfg

检查接近末尾的menuentry部分是否有windows或其他系统名入口。下图例子中是Arch Linux入口与检测到的windows10入口（安装在/dev/sda1），具体情况可能有不同：



如果你没有看到Arch Linux系统入口或者该文件不存在，请先检查/boot目录是否正确部署linux内核：

cd /boot

ls

查看是否有initramfs-linux-fallback.img initramfs-linux.img intel-ucode.img vmlinuz-linux这几个文件，如果都没有，说明linux内核没有被正确部署，很有可能是/boot目录没有被正确挂载导致的，确认/boot目录无误后，可以重新部署linux内核：

pacman -S linux

再重新生成配置文件，就可以找到系统入口。

如果你已经安装os-prober包并生成配置文件后还是没有生成其他系统的入口：

你目前处的U盘安装环境下有可能无法检测到其他系统的入口，请在下一步中重启登陆之后重新运行：

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

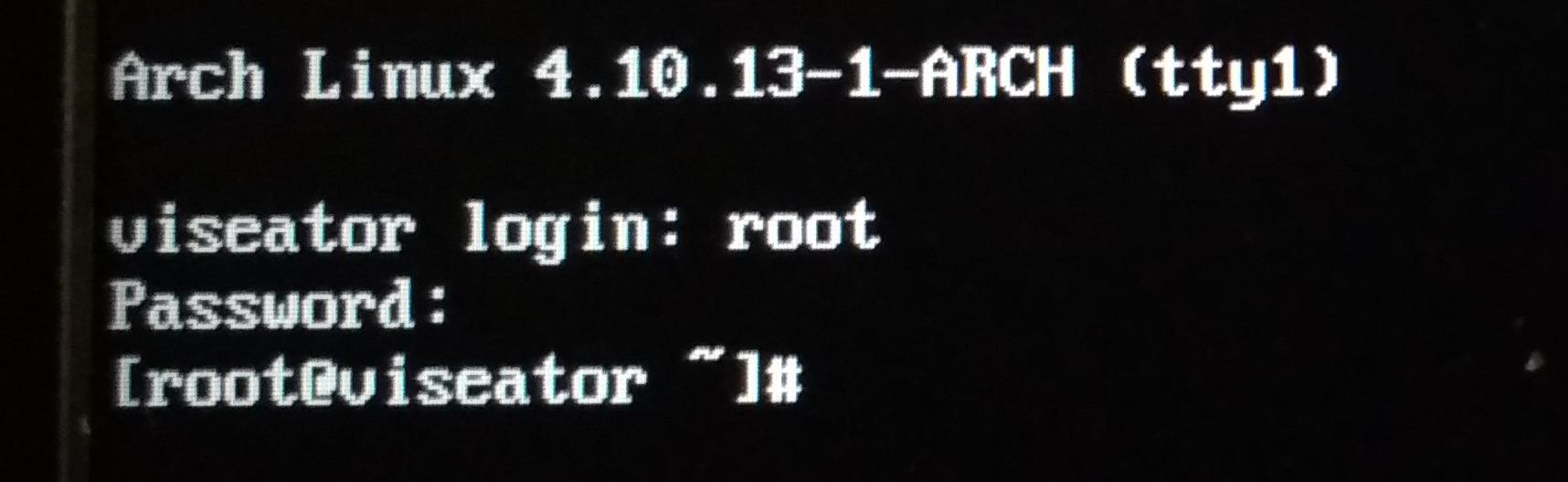
# 重启

exit

reboot

注意这个时候你可能会卡在有两行提示的地方无法正常关机，长按电源键强制关机即可，没有影响。

关机后拔出U盘，启动顺序会自动以硬盘启动，如果一切顺利，那么你将会看到下面的界面：



启动时有可能会有输出信息显示在这里，直接回车就可以了。

输入root，再输入之前设置的密码，显示出命令提示符，恭喜你，你已经成功安装ArchLinux！

# 安装后的配置

## 创建交换文件

交换文件可以在物理内存不足的时候将部分内存暂存到交换文件中，避免系统由于内存不足而完全停止工作。

之前我们通常采用单独一个分区的方式作为交换分区，现在更推荐采用交换文件的方式，更便于我们的管理。

分配一块空间用于交换文件，执行：

fallocate -l 512M /swapfile （请将512M换成需要的大小，只能以M或G为单位）

交换文件的大小可以自己决定，推荐4G以下的物理内存，交换文件与物理内存一致，4G以上的物理内存，交换文件4-8G。

更改权限，执行：chmod 600 /swapfile

设置交换文件，执行：mkswap /swapfile

启用交换文件，执行：swapon /swapfile

最后我们需要编辑/etc/fstab为交换文件设置一个入口，使用vim打开文件：vim /etc/fstab

注意编辑fstab文件的时候要格外注意不要修改之前的内容，直接在最后新起一行加入以下内容：

/swapfile none swap defaults 0 0

## 创建新用户

useradd -m -G wheel username （请自行替换username为你的用户名）

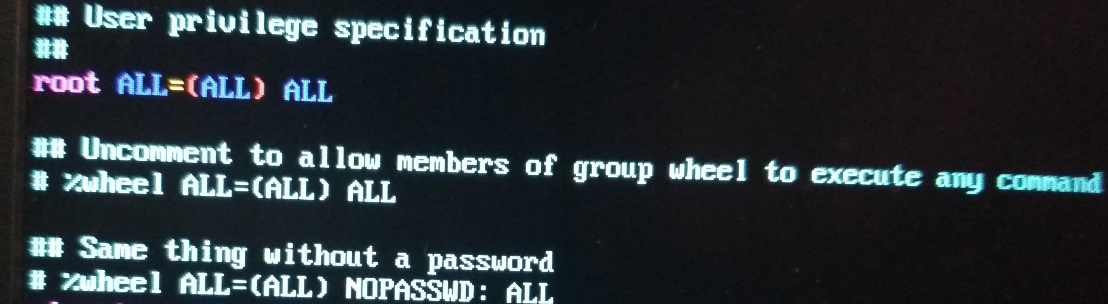
passwd username （请自行替换username为你的用户名）

## 配置sudo

pacman -S sudo

接下来我们需要用专门的visudo命令来编辑sudo的配置文件：visudo

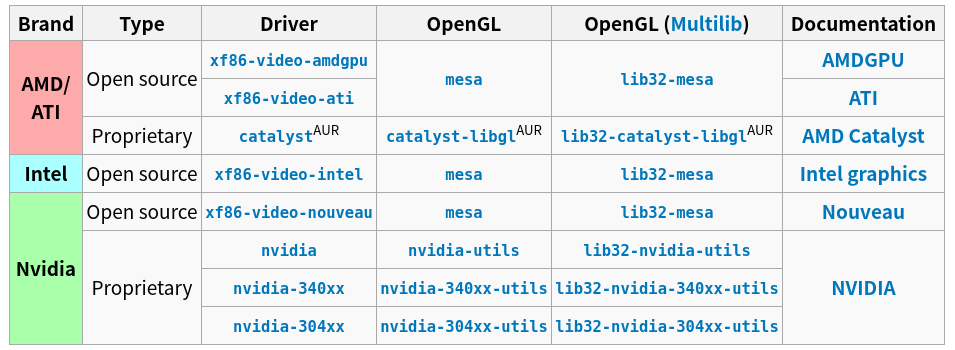
实际上就是vim的操作，使用它是为了对编辑后的文件进行检查防止格式的错误。



找到# %wheel ALL=(ALL)ALL这行，去掉之前的#注释符，保存并退出就可以了。

这里的%wheel就是代表wheel组，意味着wheel组中的所有用户都可以使用sudo命令。

## 图形界面的安装



参照这个表格，安装相应的包，比如你是intel的集成显卡（绝大多数人的情况），执行：pacman -S xf86-video-intel

提示：Nvidia的独显驱动如非必要，建议只装集成显卡的驱动（省电，如果同时装也会默认使用集成显卡），不容易出现冲突问题。相反，如果集成显卡驱动有问题无法装上，可以装独显驱动，具体的版本请到下面的链接查询：

### 安装Xorg

sudo pacman -S xorg

### 安装xfce

sudo pacman -S xfce4 xfce4-goodies

## 安装lightdm和lightdm-gtk-greeter

systemctl enable lightdm

## 配置网络

到现在我们已经安装好了桌面环境，但是还有一件事情需要我们提前设置一下。由于我们之前使用的一直都是netctl这个自带的网络服务，而桌面环境使用的是NetworkManager这个网络服务，所以我们需要禁用netctl并启用NetworkManager：

sudo systemctl disable netctl

sudo systemctl enable NetworkManager （注意大小写）

同时你可能需要安装工具栏工具来显示网络设置图标（某些桌面环境已经装了，但是为了保险可以再装一下）：

sudo pacman -S network-manager-applet

## 美化界面

Github下搜索arc主题和paper icon安装即可

# 参考文献

1. https://www.viseator.com/2017/05/17/arch\_install/