

# Instalando Arch

Hashimoto

## 1 Preparando Bootloader

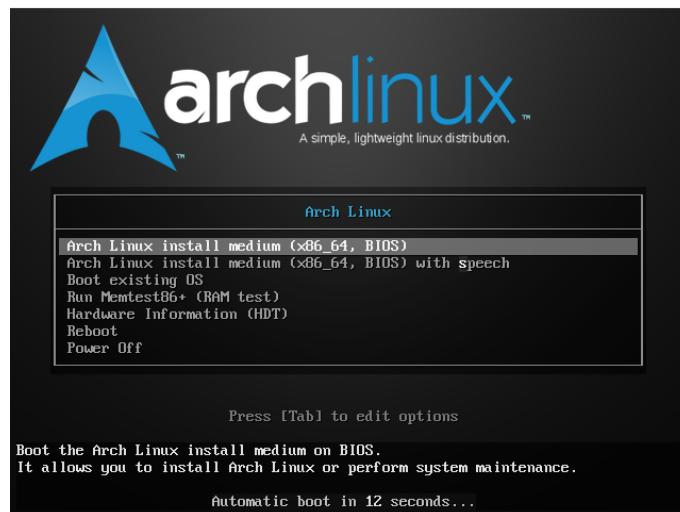
Primeiro temos que ler o guia de instalação do Arch ([https://wiki.archlinux.org/title/Installation\\_guide](https://wiki.archlinux.org/title/Installation_guide)). Todos os passos estão descritos lá (menos os de customização).

Após, temos que baixar a *ISO* e “instalar” num pendrive (se for fazer em um “PC de verdade”). Já que já instalei um arch antes e vou usar uma VM (*Virtual Machine*), já tenho a *ISO* e não preciso instalar num pendrive.

*Obs: Se for bootar num PC de verdade lembre-se de “spammar” as F-keys e ordenar para bootar do pendrive primeiro.*

## 2 Ligando o PC/VM

Após ligar, com o pendrive já plugado vai aparecer a janelinha mágica da sua *ISO*. Selecione a primeira opção.



Vai aparecer um monte de texto escrolando para cima, aguarde...

### 3 Preparando o terreno

Se você aguardou, deve estar vendo uma “tela preta” com uma vaquinha e logo em baixo

```
root@archiso ~ #
```

e um cursorzinho piscando. Está olhando para um terminal e não se preocupe, ele não morde.

```
Arch Linux 5.11.16-arch1-1 (tty1)
archiso login: root (automatic login)

/ To install Arch Linux follow the installation guide: \
| https://wiki.archlinux.org/index.php/Installation_guide |
| For Wi-Fi, authenticate to the wireless network using the iuctl utility. |
| For mobile broadband (WWAN) modems, connect with the mmcli utility. |
| Ethernet, WLAN and WWAN interfaces using DHCP should work automatically. |
| After connecting to the internet, the installation guide can be accessed |
\ via the convenience script Installation_guide. /

  ^ ^
  (oo)\_____/
  (__)|  )\   )\/
     ||----w )
     ||     ||

root@archiso ~ #
```

*Obs: Existe um arquivo no diretório atual chamado `install.txt` ele contém o mesmo conteúdo da Arch Wiki. Recomendo fortemente que de um `Alt-2` para abrir esse `txt` no seu editor de texto preferido e usar como guia para o resto da instalação. Você pode voltar para o outro terminal usando `Alt-1`.*

O próximo passo é configurar o seu *layout* de teclado. O *layout* padrão é Americano. Se não tem certeza do nome do seu *layout* favorito, pode procurar usando:

```
# ls /usr/share/kbd/keymaps/**/*.map.gz | less
```

Eu uso `br-abnt2`, então usamos:

```
# loadkeys br-abnt2
```

*Obs: Também dá para trocar a fonte usando `setfont`. e as fontes estão em `/usr/share/kbd/consolefonts`.*

Fazemos um pequeno teste para saber o modo de boot. Se o diretório `/sys/firmware/efi/efivars` existir, o PC foi bootado com *UEFI*, caso contrário foi bootado com *BIOS* ou *CSM*.

```
# ls /sys/firmware/efi/efivars
ls: cannot access '/sys/firmware/efi/efivars': No such file or directory
{2}
```

Então no meu caso foi com *BIOS* ou *CSM*.

Podemos testar a internet com `ping -c 3 <seu.dominio.preferido>`, eu gosto do `8.8.8.8`, dá para digitar rapidinho (o `-c 3` é para não ter que dar *Ctrl-C* depois). Como estou em uma VM não preciso fazer muita coisa e já tenho internet.

```
root@archiso ~ # ping -c 3 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=116 time=20.9 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=116 time=19.8 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=116 time=18.6 ms
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 18.581/19.778/20.941/0.963 ms
root@archiso ~ #
```

Vamos usar `timedatectl set-ntp true` para ter certeza que o relógio do sistema está certo.

```
--- TL;DR ---
# loadkeys br-abnt2 --- Configurar o teclado
# ping -c 3 8.8.8.8 --- Testar se internet funciona
# timedatectl set-ntp true --- Acertar o relógio do sistema
```

## 4 Particionando o Disco

Primeiro temos que achar o nome do/escolher o disco. Não podemos escolher discos terminados em *rom*, *loop* ou *airoot*

```
# fdisk -l
```

```
root@archiso ~ # fdisk -l
Disk /dev/sda: 8 GiB, 8589934592 bytes, 16777216 sectors
Disk model: VBOX HARDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop0: 534.78 MiB, 560738304 bytes, 1095192 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
root@archiso ~ #
```

Nosso disco é `/dev/sda/` e tem *8 Gib.* :)

Temos que criar a tabela de partição, no nosso caso uma *GPT* (*Por quê, você pergunta? Porque é a opção comum*). Vamos usar o próprio `fdisk` para criar a tabela.

```
# fdisk /dev/sda
```

```
Command (m for help): m
Help:

GPT
M   enter protective/hybrid MBR

Generic
d   delete a partition
F   list free unpartitioned space
l   list known partition types
n   add a new partition
p   print the partition table
t   change a partition type
v   verify the partition table
i   print information about a partition

Misc
m   print this menu
x   extra functionality (experts only)

Script
l   load disk layout from sfdisk script file
O   dump disk layout to sfdisk script file

Save & Exit
w   write table to disk and exit
q   quit without saving changes

Create a new label
g   create a new empty GPT partition table
G   create a new empty SGI (IRIX) partition table
o   create a new empty DOS partition table
s   create a new empty Sun partition table

Command (m for help):
```

Vamos gerar nossa nova tabela com `g`, criar 3 partições: uma para `boot` com *260MiB*, uma para `/` com *7.2GiB* e outra para o `swap` com *512MiB*.

Então nosso input ficou assim (cada `;` é um `Enter`):

```
g;n;;+260M;t;4;n;;;+7.2G;n;;;t;;19;w
```

Olha que bonita nossa tabela!

```
# fdisk -l /dev/sda
```

```

root@archiso ~ # fdisk -l
Disk /dev/sda: 8 GiB, 8589934592 bytes, 16777216 sectors
Disk model: UBOX HARDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 7DAE2EB0-3EEC-AB45-B9C4-7E54CE074700

Device        Start      End  Sectors  Size Type
/dev/sda1      2048    534527    532480  260M BIOS boot
/dev/sda2     534528 15634431 15099904   7.2G Linux filesystem
/dev/sda3     15634432 16777182   1142751   558M Linux swap

Disk /dev/loop0: 620.81 MiB, 650969088 bytes, 1271424 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
root@archiso ~ # _

```

*Obs: Se estivéssemos usando UEFI precisaríamos criar uma partição para boot dela.*

Agora temos que formatar as partições e montá-las/ativá-la:

```

--- Formatar ---
# mkfs.ext4 /dev/sda2
# mkswap /dev/sda3
--- Montar ---
# mount /dev/sda2 /mnt
--- Ativar ---
# swapon /dev/sda3

```

## 5 Instalando

Os *mirror servers* (lugares de onde vai ser baixado o sistema) estão em `/etc/pacman.d/mirrorlist`. Podemos trocar a ordem, deletar os mais lentos, mas a ordem padrão já é boa então vamos só instalar usando o script:

```
# pacstrap /mnt base linux vim
```

*Obs: Se não estiver instalando num container ou virtual-machine passe `linux-firmware` como um argumento extra.*

Agora é sentar e esperar...

## 6 Configurando o sistema

Vamos gerar o Fstab:

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

Entrar no nosso novo sistema como *root*:

```
# arch-chroot /mnt
```

Escolher o *Time zone*:

```
# ln -sf /usr/share/zoneinfo/America/Sao_Paulo /etc/localtime
# hwclock --systohc
```

Para configurar a localização, temos que procurar o arquivo `/etc/locale.gen` e descomentar a linha com `pt_BR.UTF-8 UTF-8` (para isso que baixamos o `vim`!).

E depois de descomentar a linha:

```
# locale-gen
# echo LANG=pt_BR.UTF-8 >> /etc/locale.config
# echo KEYMAP=br-abnt2 >> /etc/vconsole.config
```

Configurar o network escolhemos um hostname e escrevemos em `/etc/hostname` e também precisamos escrever em `/etc/hosts` isso:

```
127.0.0.1      localhost
::1           localhost
127.0.1.1      vm-arch.localdomain  vm-arch

# echo vm-arch >> /etc/hostname
```

A última coisa que falta para darmos o primeiro boot é o boot loader.

```
# pacman -S grub
# grub-install /dev/sda
# grub-mkconfig -o /boot/grub/grub.cfg
```

Antes de dar reboot, vamos aproveitar o root para escolher sua senha super-secreta com `passwd` e para criar um usuário *shikiyo*:

```
# passwd
# useradd -m -g users -G wheel shikiyo
# passwd shikiyo
```

Vamos baixar e configurar utilitários de internet e também `sudo`, para o `sudo` é só descomentar a linha `%wheel ALL=(ALL) ALL`.

```
# pacman -S networkmanager sudo
# EDITOR=vim visudo
# systemctl enable NetworkManager
```

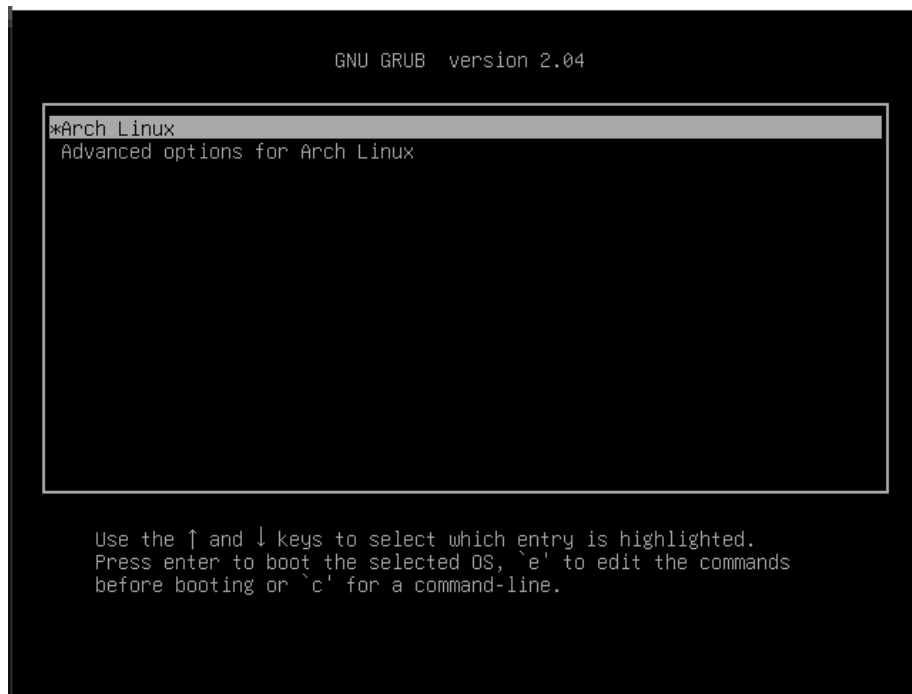
Agora vamos dar reboot

```
# exit
# umount -a
# reboot
```

*Obs: Lembrar de retirar o pendrive.*

*Obs: Agora é um bom momento para clonar a ISO da máquina virtual.*

## 7 Pós-instalando



Esse é o momento de instalar o que seu coração pedir! Vou instalar apenas o `xorg` e o `gnome`.

```
$ sudo pacman -S xorg gnome
$ sudo systemctl enable gdm
```

E no próximo reboot teremos uma interface gráfica bonitinha!

