

iniciando a instalação do arch:

comando 1 → `ping google.com` (verifica se há conexão de internet)

comando 2 → `ls /usr/share/kbd/keymaps/i386/qwerty` (vai até o diretório do layout do teclado. Por padrão, o arch linux usa o default o layout US, como meu teclado também segue esse layout, não realizar nenhuma alteração)

comando 3 → `loadkeys us.map.gz` (escolhendo um layout para exemplificar. Por padrão, o arch linux usa o default o layout US, como meu teclado também segue esse layout, não realizar nenhuma alteração)

comando 4 → `sudo pacman -Sy reflector openssh` (instalando os dois pacotes “reflector” e “openssh”)

comando 5 → `systemctl start sshd` (escolhendo a sua para a iso)

comando 6 → `passwd` (digitamos duas vezes a senha que queremos setar)

senha definida como 010203

comando 7 → `timedatectl set-ntp true` (comando que me permite pegar a data e horário correto dos servidores)

comando 8 → `reflector -c Brazil -a 6 --sort rate --save` (para fazer download de pacotes, mas seguindo os parametros do comando que define o país como brasil, escolhe um servidor ativo nas últimas 5 horas e ordena-os “rate” e salva em `/etc/pacman.d/mirrorlist`)

comando 9 → `pacman -Sy` (atualiza os servidores)

precisamos criar uma partição EFI e uma partição home:

comando 10 → `gdisk /dev/sda`

digitamos → “n”

digitamos → enter

digitamos → enter

digitamos → “+200M”

digitamos → “ef00”

partição EFI criada

digitamos → “n”

digitamos → enter

digitamos → enter

digitamos → enter

digitamos → enter

partição para os arquivos do sistema linux criada

digitamos → “w”

digitamos → “y”

mensagem de operação realizada com sucesso

```
root@archiso ~ # gdisk /dev/sda
GPT fdisk (gdisk) version 1.0.8

Partition table scan:
  MBR: not present
  BSD: not present
  APM: not present
  GPT: not present

Creating new GPT entries in memory.

Command (? for help): n
Partition number (1-128, default 1):
First sector (34-20971486, default = 2048) or {+-}size{KMGT}:
Last sector (2048-20971486, default = 20971486) or {+-}size{KMGT}: +200M
Current type is 8300 (Linux filesystem)
Hex code or GUID (L to show codes, Enter = 8300): ef00
Changed type of partition to 'EFI system partition'

Command (? for help): n
Partition number (2-128, default 2):
First sector (34-20971486, default = 411648) or {+-}size{KMGT}:
Last sector (411648-20971486, default = 20971486) or {+-}size{KMGT}:
Current type is 8300 (Linux filesystem)
Hex code or GUID (L to show codes, Enter = 8300):
Changed type of partition to 'Linux filesystem'

Command (? for help): w

Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING
PARTITIONS!!

Do you want to proceed? (Y/N): y
OK; writing new GUID partition table (GPT) to /dev/sda.
The operation has completed successfully.
root@archiso ~ #
```

comando 11 → lsblk (lista as partições geradas)

comando 12 → mkfs.fat -F32 /dev/sda1 (nesse comando adicionamos um formata para a partição “/dev/sda1” anteriormente definida como um sistema EFI)

comando 13 → mkfs.ext4 /dev/sda2 (nos passos anteriores, definimos a partição “/dev/sda2” responsável pelos arquivos do sistema linux, então no comando atual definimos essa mesma partição para o formato ext4, pois ext4 é responsável por partições linux nativas)

agora precisamos montar as partições:

comando 14 → `mount /dev/sda2 /mnt` (esse diretório já existe)

agora precisamos montar a outra partição em um diretório que ainda não existe, então devemos criá-lo:

comando 15 → `mkdir -p /mnt/boot/efi` (criamos os diretórios)

comando 16 → `mount /dev/sda1 /mnt/boot/efi` (montamos a partição no diretório criado no comando anterior)

comando 17 → `lsblk` (não obrigatório, mas é útil para verificar se os comandos foram feitos com sucesso)

```
NAME      MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
loop0     7:0    0 653.4M  1 loop /run/archiso/airootfs
sda        8:0    0    10G   0 disk
├─sda1     8:1    0   200M   0 part /mnt/boot/efi
└─sda2     8:2    0   9.8G   0 part /mnt
sr0       11:0    1 807.3M  0 rom  /run/archiso/bootmnt
```

instalação de pacotes básicos:

comando 18 → `pacstrap /mnt base linux linux-firmware vim`((instalando pacotes básicos, o kernel e o editor de texto vim)

comando 19 → `genfstab -U /mnt >> /mnt/etc/fstab`

comando 20 → `cat /mnt/etc/fstab` (não obrigatório, mas serve para acompanhar se os comandos estão sendo executados com exito)

```
root@archiso ~ # cat /mnt/etc/fstab
# Static information about the filesystems.
# See fstab(5) for details.

# <file system> <dir> <type> <options> <dump> <pass>
# /dev/sda2
UUID=8b27ca1e-1d54-4b8a-bd63-7f4aeb5db2ab      /          ext4          rw,relatime    0 1

# /dev/sda1
UUID=2806-22D0          /boot/efi    vfat          rw,relatime,fmask=0022,dmask=0022,codepage=4
37,iocharset=ascii,shortname=mixed,utf8,errors=remount-ro    0 2

root@archiso ~ # _
```

comando 21 → `arch-chroot /mnt`

precisamos criar o arquivo de swap:

comando 22 → tem espaço “ dd if=/dev/zero of=/swapfile bs=1G count=2 status=progress” (cria um arquivo de swap de 2gb)

```
root@archiso ~ # arch-chroot /mnt
[root@archiso /]# dd if=/dev/zero of=/swapfile bs=1G count=2 status=progress
2147483648 bytes (2.1 GB, 2.0 GiB) copied, 3 s, 809 MB/s
2+0 records in
2+0 records out
2147483648 bytes (2.1 GB, 2.0 GiB) copied, 2.6632 s, 806 MB/s
[root@archiso /]# _
```

agora que criamos o arquivo de swap precisamos mudar as suas permissões:

comando 23 → chmod 600 /swapfile

comando 24 → mkswap /swapfile (faz o swap)

comando 25 → swapon /swapfile (ativa o swap)

comando 26 → vim /etc/fstab

abrindo o editor de texto “vim” digitamos nas ultimas linhas → /swapfile none swap default 0 0

apertamos esc e digitamos → :wp e apertamos enter

comando 27 → timedatectl list-timezones | grep Brazil (serve para buscar o horário de acordo com a regioao escolhida)

```
[root@archiso /]# timedatectl list-timezones | grep Sao_Paulo
America/Sao_Paulo
```

comando → ln -sf /usr/share/zoneinfo/America/Sao_Paulo /etc/localtime (seta a localidade e armazena em /etc/localtime. Como moro no Rio de Janeiro, o fuso horário de são paulo serve)

comando 29 → hwclock --systohc

comando 30 → vim /etc/locale.gen (devemos remover o comentário das linguagens que queremos, no meu caso, optei pelo en_us.UTF-8 UTF8, devido ao layout do meu teclado)

apertamos esc → digitamos :wq

comando 31 → locale-gen

comando 32 → `echo LANG=en_US.UTF-8 >> /etc/locale.conf`

comando 33 → `echo KEYMAP=<layout_setado> >> /etc/vconsole.conf`

explicacao do comando 33 se o usuário tem alguma preferência no `/etc/vconsole.conf`

ler a preferência do usuário para definir teclado

caso contrário

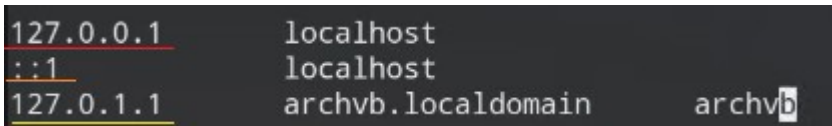
ler do arquivo default do sistema(no caso, eu não preciso executar, devido ao layout do meu teclado)

comando 34 → `vim /etc/hostname` (nomeamos o nosso sistema, no meu caso nomeei de “archvb”)

apertamos esc → digitamos :wq

configurando um endereço ip pra loopback. Sendo esses os padrões

comando 35 → `vim /etc/hosts`



```
127.0.0.1    localhost
::1         localhost
127.0.1.1    archvb.localdomain    archvb
```

em vermelho temos o endereço ipv4

em laranja temos o endereço ipv6

comando 36 → `passwd` (definindo a senha para o root usr)

instalando o boot loader e alguns pacotes:

comando 37 → `pacman -S grub efibootmgr networkmanager network-manager-applet dialog os-prober mtools dosfstools base-devel linux-headers cups reflector openssh git xdg-utils xdg-user-dirs virtualbox-guest-utils`

apertamos enter para aceitar o download dos pacotes

instalando o grub:

comando 38 → `grub-install --target=x86_64-efi --efi-directory=/boot/efi --bootloader-id=GRUB`

comando 39 → `grub-mkconfig -o /boot/grub/grub.cfg` (cria um arquivo de configuração)

habilitando serviços:

comando 40 → `systemctl enable NetworkManager`

comando 41 → `systemctl enable sshd`

criando um novo usuário:

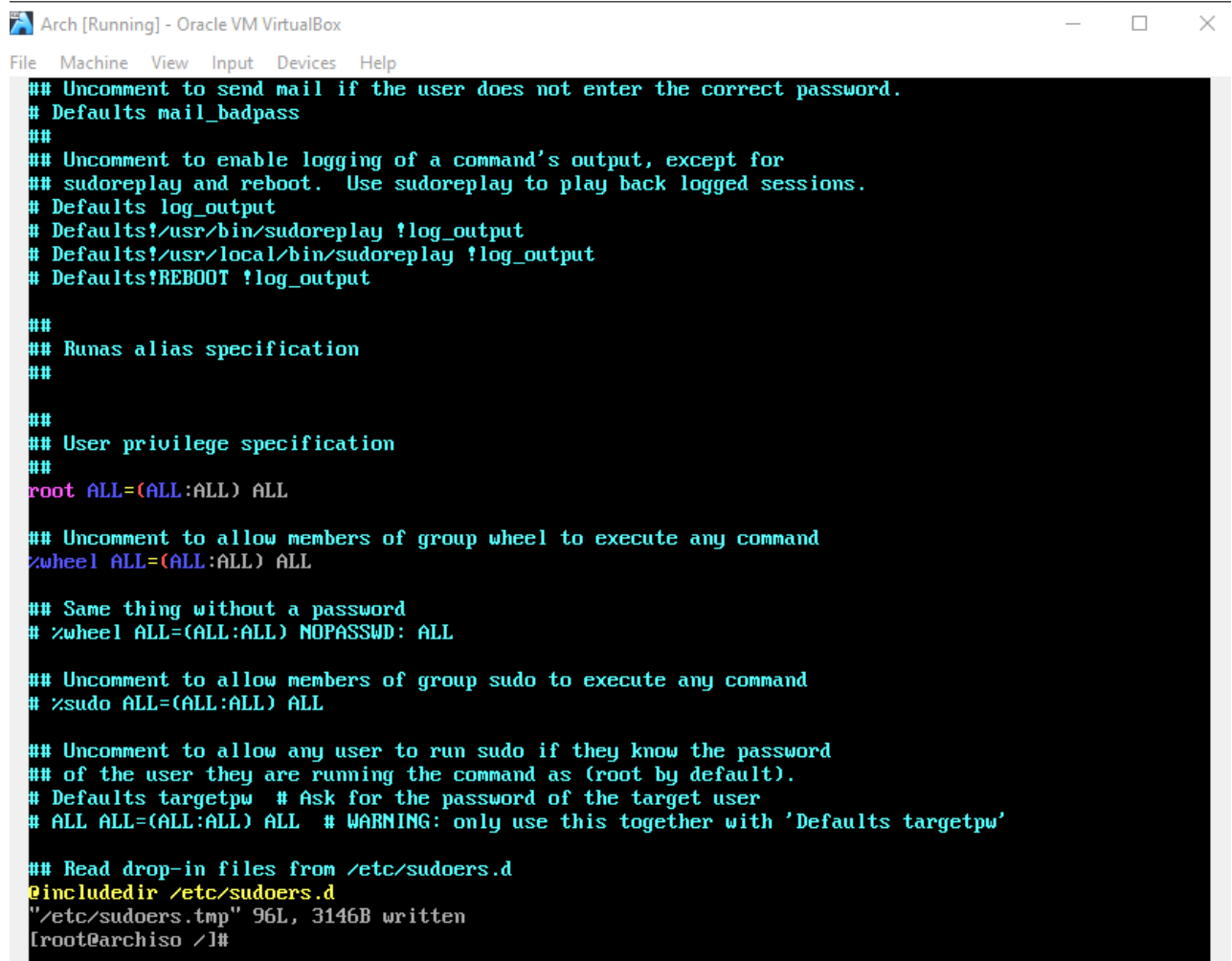
comando 42 → `useradd -mG wheel romio` (criando um novo usuário de nome “romio”)

comando 43 → `passwd romio` (cria uma senha para o usuário escolhido no comando anterior)

digita a senha

repete a senha digitada

comando 43 → `EDITOR=vim visudo` (editamos o arquivo e removemos o comentário de “%wheel ALL=(ALL) ALL”)



```
Arch [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
## Uncomment to send mail if the user does not enter the correct password.
# Defaults mail_badpass
##
## Uncomment to enable logging of a command's output, except for
## sudoreplay and reboot. Use sudoreplay to play back logged sessions.
# Defaults log_output
# Defaults! /usr/bin/sudoreplay !log_output
# Defaults! /usr/local/bin/sudoreplay !log_output
# Defaults! REBOOT !log_output
##
## Runas alias specification
##
##
## User privilege specification
##
root ALL=(ALL:ALL) ALL

## Uncomment to allow members of group wheel to execute any command
%wheel ALL=(ALL:ALL) ALL

## Same thing without a password
# %wheel ALL=(ALL:ALL) NOPASSWD: ALL

## Uncomment to allow members of group sudo to execute any command
# %sudo ALL=(ALL:ALL) ALL

## Uncomment to allow any user to run sudo if they know the password
## of the user they are running the command as (root by default).
# Defaults targetpw # Ask for the password of the target user
# ALL ALL=(ALL:ALL) ALL # WARNING: only use this together with 'Defaults targetpw'

## Read drop-in files from /etc/sudoers.d
@include_dir /etc/sudoers.d
"/etc/sudoers.tmp" 96L, 3146B written
[root@archiso /]#
```

comando 44 → `exit`

comando 45 → `umount -a`

comando 46 → `reboot` (reinicia o sistema)

fazemos login com o usuário e senha escolhido

instalando os drives de video

comando 47 → `sudo pacman -S xf86-video-vmware xorg lightdm lightdm-gtk-greeter xfce4 xfce4-goodies chromium materia-gtk-theme papirus-icon-theme`
digitamos a senha e pressionamos enter para aceitar o download dos pacotes

comando 48 → `sudo systemctl enable lightdm`

comando 49 → `exit`

comando 50 → `reboot`

após todos esses comandos, a instalação foi problemática pois o áudio não foi instalado corretamente. Para corrigir o erro foram necessários os comandos abaixo:

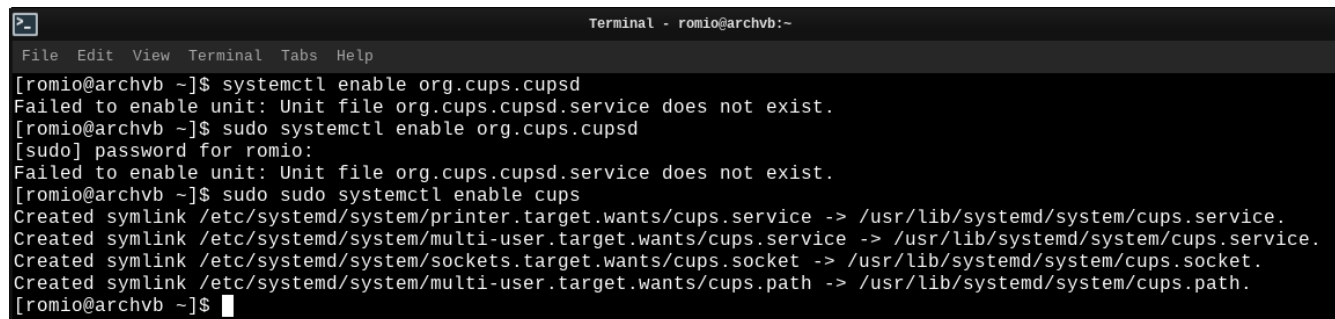
- 1) `pacman -S pulseaudio`
- 2) `rm -rf /tmp/pulse* ~/.pulse* ~/.config/pulse`
- 3) `pulseaudio -k`
- 4) `pulseaudio --start`

após esses comandos, o audio ficou funcionando, mas com um pequeno chiado, foi necessário instalar a versão atualizada do comando cups:

versão desatualizada → “`sudo systemctl enable org.cups.cupsd.service`”

versão atualizada → “`sudo systemctl enable cups`”

link que descreve o problema → <https://bbs.archlinux.org/viewtopic.php?id=264591>



```
Terminal - romio@archvb:~  
File Edit View Terminal Tabs Help  
[romio@archvb ~]$ systemctl enable org.cups.cupsd  
Failed to enable unit: Unit file org.cups.cupsd.service does not exist.  
[romio@archvb ~]$ sudo systemctl enable org.cups.cupsd  
[sudo] password for romio:  
Failed to enable unit: Unit file org.cups.cupsd.service does not exist.  
[romio@archvb ~]$ sudo sudo systemctl enable cups  
Created symlink /etc/systemd/system/printer.target.wants/cups.service -> /usr/lib/systemd/system/cups.service.  
Created symlink /etc/systemd/system/multi-user.target.wants/cups.service -> /usr/lib/systemd/system/cups.service.  
Created symlink /etc/systemd/system/sockets.target.wants/cups.socket -> /usr/lib/systemd/system/cups.socket.  
Created symlink /etc/systemd/system/multi-user.target.wants/cups.path -> /usr/lib/systemd/system/cups.path.  
[romio@archvb ~]$
```

link para o video no youtube:

<https://youtu.be/EOpj4UEb3HM>