

LUKS + LVM

INTRODUÇÃO

Este documento é um guia para a instalação do Arch Linux usando o sistema live inicializado a partir de uma mídia de instalação feita a partir de uma imagem de instalação oficial. Todo o conteúdo desse documento foi feito com base no site oficial do Arch Linux na guia de instalação e o canal Learning Linux, caso seja necessário o autor desse documento fará a alteração necessária quando for preciso.

O meio de instalação desse documento é o modo criptografado com LVM.

O Arch Linux deve ser executado em qualquer máquina compatível com processador 64bits com um mínimo de 512 MiB RAM, embora seja necessária mais memória para inicializar o sistema depois da instalação, no caso utilizando recursos gráficos. Uma instalação básica deve ocupar menos de 2 GiB de espaço em disco. Como o processo de instalação precisa recuperar pacotes de um repositório remoto, este guia assume que uma conexão de Internet está disponível.

VERIFIQUE O MODO DE INICIALIZAÇÃO

Depois de iniciar o Arch Linux com um pendrive deve verificar a inicialização, para sabermos se temos suporte ao UEFI.

```
# efivar -l
```

Se o comando mostrar o diretório sem erros, o sistema será inicializado no modo UEFI. Se o diretório não existir, o sistema pode ser inicializado no modo BIOS (ou CSM). Se o sistema não inicializou no modo desejado, consulte o manual da placa-mãe.

LAYOUT DO TECLADO

O mapa de teclado padrão do console é US. Os layouts disponíveis podem ser listados com:

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

Para modificar o layout, anexe um nome de arquivo correspondente a loadkeys, omitindo o caminho e a extensão do arquivo. Por exemplo, para definir um layout de teclado brasileiro:

```
# loadkeys br-abnt2
```

Se o seu teclado for como o meu e for o americano internacional você usa:

```
# loadkeys us-acentos
```

CONEXÃO COM A INTERNET

Para configurar uma conexão de rede no ambiente de instalação, siga as seguintes etapas:

Certifique-se de que sua interface de rede esteja listada e ativada, por exemplo, com ip-link:

```
# ip link
```

Para conectar-se à rede:

- Ethernet - conecte o cabo.
- Wi-Fi - autentique-se na rede sem fio usando iwctl.

CONECTANDO COM WI-FI

```
# iwctl
```

Primeiro, se você não souber o nome do seu dispositivo sem fio, liste todos os dispositivos Wi-Fi:

```
# device list
```

Em seguida, para procurar redes:

```
# station dispositivo scan
```

OBS: note que em dispositivo é o seu dispositivo de rede, no meu caso seria wlan0, caso o seu seja diferente é só substituir.

```
# station wlan0 scan
```

Você pode listar todas as redes disponíveis:

```
# station wlan0 get-networks
```

Por fim, para conectar-se a uma rede:

```
# station dispositivo connect SSID
```

OBS: note que SSID refere-se ao nome de sua rede, no meu caso Kairo Vox.

```
# station wlan0 connect "Kairo Vox"
```

Se uma senha for necessária, você será solicitado a digitá-la.

VERIFICANDO A CONEXÃO

Antes de instalar o Arch Linux, verifique se o computador está de fato conectado à internet.

```
# ping -c 10 archlinux.org
```

Caso retorne algum tipo de erro, significa que você não está conectado na internet.

PARTICIONAR O DISCO

Primeiro vamos listar os discos conectados:

```
# lsblk
```

Vamos particionar o disco

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

Vamos ver o layout da partição atual

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

Vamos criar um layout de partição GPT.

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

Vamos criar uma tabela de partição:

```
Command (m for help): n
Partition number (1-128, default 1): {ENTER}
First sector (...): {ENTER}
```

Vamos definir o tamanho da partição de boot.

```
Last sector (...): +500M
```

OBS: Caso apareça alguma mensagem dizendo que você já possui alguma assinatura ou partição já criada, você pode remover ela.

Vamos definir o tipo da partição que acabamos de criar:

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

Vamos definir o tipo da partição como EFI System:

```
Partition type (type L to list all types): 1
```

Agora vamos criar a partição de boot

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

Vamos escolher o número da partição.

```
Partition number (2-128, default 2): {ENTER}
First sector (...): {ENTER}
```

Agora vamos definir o tamanho da partição boot:

```
Last sector (...): +500M
```

Vamos criar a última partição, onde vamos criar os volumes lógicos.

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

Definir o número da partição:

```
Partition number (3-128, default 3): {ENTER}
First sector (...): {ENTER}
```

Definir o tamanho da partição:

```
Last sector (...): {ENTER}
```

OBS: apertamos ENTER para poder utilizar todo o restante do disco.

Vamos definir o tipo de partição das unidades criadas.

```
Command (m for help): t
Partition number (1-3, default 3): {ENTER}
```

Vamos agora definir o tipo para 'Linux LVM'

```
Partition type (type L to list all types): 30
```

Vamos ver as partições criadas e seus tipos:

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

Vamos escrever as mudanças no disco:

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

Se usarmos o lsblk, vamos ver as partições criadas no disco:

```
# lsblk /dev/sda
```

OBS: lembre se que sda é o disco no qual eu vou instalar o S.O, caso o seu seja outro é só substituir.

FORMATAR O DISCO

Vamos formatar a partição EFI:

```
# mkfs.fat -F32 /dev/sda1
```

OBS: lembre se que sda1 é a primeira partição EFI do MEU disco, caso o seu seja outro é só substituir.

FORMATAR O BOOT

```
# mkfs.ext4 -L B00T /dev/sda2
```

CONFIGURAR A CRIPTOGRAFIA DO DISCO

Agora vamos configurar a criptografia do disco

```
# cryptsetup -v luksFormat /dev/sda3
```

Esse comando vai excluir todo o conteúdo da partição então vamos escrever YES para continuar.

```
Are you sure? (Type uppercase yes): YES
```

Agora vamos criar uma senha para encriptar o disco:

```
Enter passphrase for /dev/sda3: escolhaumaseinha
```

OBS: caso você se esqueça dessa senha, você não vai conseguir dar boot no sistema.

Confirme a senha que foi digitada anteriormente:

```
Verify passphrase: escolhaumaseinha
```

Agora vamos abrir o disco que acabou de ser criptografado e definir um nome, para nos referir a ele posteriormente.

```
# cryptsetup open --type luks /dev/sda3 lvm
```

OBS: eu dei o nome de lvm para o disco, mas isso fica a sua escolha.

Digite a senha que você tinha criado anteriormente:

```
Enter passphrase for /dev/sda3: escolhaumaseinha
```

CONFIGURAR LVM

Vamos criar o volume físico do nosso disco:

```
# pvcreate --dataalignment 1m /dev/mapper/lvm
```

OBS: Esse comando é recomendado caso você utilize um ssd, caso você utilize um HD mecânico use o próximo comando:

```
# pvcreate /dev/mapper/lvm
```

OBS: note que lvm é o nome dado anteriormente ao nosso /dev/sda3.

Agora vamos criar o nosso volume de grupo:

```
# vgcreate kairovox /dev/mapper/lvm
```

OBS: note que kairovox é o nome que eu escolhi para o nosso volume de grupo, escolha outro se quiser.

Agora vamos começar a criar o volume lógico root:

```
# lvcreate -L 30GB kairovox -n root
```

OBS: lvcreate -L {tamanho do volume} {nome do volume de grupo} -n {nome do volume logico}

Agora vamos criar o volume logico home:

```
# lvcreate -l 100%FREE kairovox -n home
```

OBS: note que no tamanho do volume fizemos algo diferente, no caso estamos dizendo que queremos utilizar todo o restante livre do disco.

Agora vamos formatar o volume lógico do root:

```
# mkfs.ext4 -L ROOT /dev/kairovox/root
```

Formatar o volume home:

```
# mkfs.ext4 -L HOME /dev/kairovox/home
```

MONTAGEM DOS VOLUMES

Montar o volume root:

```
# mount /dev/kairovox/root /mnt
```

Criar a partição home:

```
# mkdir /mnt/home
```

Montar volume home:

```
# mount /dev/kairovox/home /mnt/home
```

Vamos criar a partição boot:

```
# mkdir /mnt/boot
```

Vamos montar a partição boot:

```
# mount /dev/sda2 /mnt/boot
```

Vamos criar agora a partição etc.:

```
# mkdir /mnt/etc
```

Agora nós podemos criar o nosso arquivo de fstab.

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

Agora podemos ver como o nosso arquivo ficou com o less:

```
# less /mnt/etc/fstab
```

INSTALAR O SISTEMA BASE

Agora podemos instalar os programas base do nosso sistema.

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

CONFIGURAÇÃO DO SISTEMA

Agora vamos configurar o Sistema.

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

Vamos instalar pacotes essenciais:

```
# nvim vim nano git wget man linux linux{-firmware,-headers}
```

Vamos instalar pacotes para se conectar à internet:

```
# pacman -S wpa_supplicant wireless_tools dialog iw networkmanager netctl
```

GERENCIADOR DE REDE

Caso você utilize somente internet cabeada:

```
# systemctl enable dhcpcd
```

Caso voce utilize o Wi-Fi:

```
# systemctl enable NetworkManager
```

Agora vamos instalar o lvm2 para adicionar o suporte apropriado para a nossa configuração lvm.

```
# pacman -S lvm2
```

Agora vamos editar o arquivo mkinitcpio.

```
# nano /etc/mkinitcpio.conf
```

Vamos descer e procurar onde está o “HOOKS”, e editar essa linha e deixar desse jeito:

```
HOOKS=(base udev autodetect modconf block encrypt lvm2 filesystems keyboard fsck)
```

Vamos rodar essa configuração que alteramos:

```
# mkinitcpio -p linux
```

LOCALIZAÇÃO

Edite /etc/locale.gen e descomente pt_BR.UTF-8 UTF-8 caso você queira o sistema em português. Caso queira o sistema em inglês, deve descomentar: en_US.UTF-8 UTF-8.

Agora é só gerar os locais:

```
# locale-gen
```

Crie o arquivo locale.conf e defina a variável LANG de acordo:

```
# nano /etc/locale.conf  
LANG=en_US.UTF-8
```

OBS: caso queira o sistema em português é só alterar para: pt_BR.UTF-8.

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

LAYOUT DO TECLADO

Se você definir o layout do teclado, faça as alterações persistentes em vconsole.conf:

```
# nano /etc/vconsole.conf  
KEYMAP=us-acentos
```

OBS: caso queira o teclado em português é só alterar para: br-abnt2.

CONFIGURAR O REPOSITÓRIO

Vamos habilitar o repositório multilib para podermos instalar aplicativos de arquitetura de 32bits. Vamos entrar no arquivo /etc/pacman.conf e procurar no final do arquivo [multilib] e descomentar. Salvar e sair.

Atualizar os repositórios:

```
# pacman -Sy
```

CONFIGURAR A REDE

Crie o arquivo de nome de host:

```
# nano /etc/hostname  
archlinux
```

OBS: eu escolhi archlinux como meu nome de host, caso prefira outro é só alterar.

Adicione entradas correspondentes aos hosts:

```
# nano /etc/hosts  
127.0.0.1    localhost  
::1         localhost  
127.0.1.1    archlinux.localdomain archlinux
```

Para alterar o DNS do nosso sistema:

```
# nano /etc/resolv.conf  
nameserver 1.1.1.1  
nameserver 1.0.0.1  
nameserver 208.67.222.222  
nameserver 208.67.220.220
```

CRIAÇÃO DO USUÁRIO

```
# useradd -m -g users -G wheel,sys,storage,video -s /bin/bash kairovox
```

OBS: eu escolhi kairovox como meu nome de usuário, caso prefira outro é só alterar.

Agora vamos mudar a senha de administrador:

```
# passwd
```

Agora vamos criar a senha do nosso novo usuário.

```
# passwd kairovox
```

Agora vamos permitir que os usuários no grupo wheel, sejam capazes de executar tarefas de administrador.

```
# visudo
```

Vamos procurar por `wheel ALL=(ALL) ALL` e descomentar.

aperta esc, :x e enter. (vai sair salvando as alterações.)

GRUB

Vamos primeiro instalar o grub:

```
# pacman -S grub efibootmgr dosfstools os-prober mtools
```

Vamos editar o arquivo do grub:

```
# nano /etc/default/grub
```

Temos que ter muito cuidado com a mudança que vamos fazer aqui, senão o nosso sistema simplesmente não vai dar boot.

Vamos tirar o comentário da linha:

```
GRUB_ENABLE_CRYPTODISK=y
```

Vamos modificar a linha:

```
GRUB_CMDLINE_LINUX_DEFAULT="loglevel=3 cryptdevice=/dev/sda3:kairovox:allow-discards quiet"
```

Vamos criar agora o diretório EFI:

```
# mkdir /boot/EFI
```

Vamos montar a partição EFI:

```
# mount /dev/sda1 /boot/EFI
```

Vamos instalar o GRUB de fato no dispositivo:

```
# grub-install --target=x86_64-efi --bootloader-id=ArchLinux --recheck
```

[Agora vamos copiar o arquivo para essa pasta \(caso você esteja usando a linguagem inglês\):](#)

```
# cp /usr/share/locale/en\@quot/LC_MESSAGES/grub.mo /boot/grub/locale/en.mo
```

Agora vamos gerar o arquivo de configuração do GRUB:

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

SWAP

Criar o arquivo:

```
# fallocate -l 4G /swapfile
```

Ajustar as permissões do arquivo

```
# chmod 600 /swapfile
```

Criar o swap

```
# mkswap /swapfile
```


Vamos fazer uma cópia do arquivo de fstab original para caso de algum problema.

```
# cp /etc/fstab /etc/fstab.backup
```

Vamos colocar o swapfile no arquivo de fstab.

```
# echo '/swapfile none swap sw 0 0' | tee -a /etc/fstab
```

Vamos ativar o swap no sistema:

```
# swapon /swapfile
```

Vamos instalar o microcode do nosso CPU. Caso seja Intel:

```
# pacman -S intel-ucode
```

Caso seja AMD:

```
# pacman -S amd-ucode
```

Vamos sair e reiniciar o sistema:

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

PÓS INSTALAÇÃO

Bem-vindo ao "Guia de pós instalação do Arch Linux " Neste documento, vou te ensinar a como configurar o seu Arch depois da instalação, instalando programas, personalizando e criando aliases.

Lembre-se que este documento foi feito com foco em desenvolvimento e diversas configurações e personalizações não servem para usuários comuns. Deve-se salientar também que esse documento não vai abordar instalação e personalização de uma DE, pois vamos utilizar uma WM (bspwm).

CONFIGURAÇÕES INICIAIS

Entre como super usuário

```
$ su
```

Conecte-se a sua rede Wi-Fi

```
# nmtui
```

Verificar a conectividade com a net:

```
# ping -c 10 www.google.com
```

INSTALAR DRIVERS GRÁFICOS

```
# pacman -S xorg-server xorg-xinit xorg-apps gvfs-mtp
```

Descobrir qual placa de vídeo o seu computador está usando.

```
# lspci -k | grep -A 2 -i "VGA"
```

Instalar pacote referente a sua placa de vídeo.

Virtual Box:

```
# pacman -S virtualbox-guest-utils
```

AMD:

```
# pacman -S xf86-video-amdgpu
```

Intel:

```
# pacman -S xf86-video-intel
```

Nvidia:

```
# pacman -S xf86-video-nouveau
```

DRIVERS GRÁFICOS PROPRIETÁRIOS

NVIDIA

Instale o driver apropriado para a sua placa:

Para placas da série GeForce 400 ou mais recentes:

```
# pacman -S nvidia
```

Para placas da série GeForce 8/9 e 100-300 (NV5x, NV8x, NV9x e NVAx):

```
# pacman -S pacote nvidia-340xx
```

Para placas da série GeForce 6/7 (NV4x e NV6x):

```
# pacman -S nvidia-304xx
```

Para as placas da série GeForce 400/500 (NVCx e NVDx) de 2010-2011:

```
# pacman -S nvidia-390xx
```

OBS: para cada pacote mencionado também há seu respectivo pacote LTS.

Se você estiver com sistema 64 bits você também precisa de um suporte OpenGL de 32 bits, você também deve instalar o pacote lib32 equivalente do repositório multilib:

```
# pacman -S lib32-nvidia-utils lib32-nvidia-390xx-utils lib32-nvidia-340xx-utils
```

Optimus Bumblebee

```
# pacman -S nvidia nvidia-utils bumblebee primus xf86-video-intel mesa  
# systemctl enable bumblebeed.service
```

Optimus Bumblebee 340xx_legacy:

```
# pacman -S nvidia-340xx nvidia-340xx-utils bumblebee primus xf86-video-intel mesa  
# systemctl enable bumblebeed.service
```

Optimus Bumblebee 304xx_legacy:

```
# pacman -S nvidia-304xx nvidia-304xx-utils bumblebee primus xf86-video-intel mesa  
# systemctl enable bumblebeed.service
```

ATI

```
# pacman -S xf86-video-ati lib32-mesa-dri lib32-mesa-libgl
```

DRIVERS PARA PLACA DE SOM

```
# pacman -S alsa-utils pulseaudio pavucontrol
```

Plugins para reprodução de mp3 e outras mídias:

```
# pacman -S alsa-oss alsa-lib
```

CODECS DE ÁUDIO

```
# pacman -S gst-plugins-base gst-plugins-base-libs gst-plugins-good gst-plugins-ugly gst-libav
```

ARCH USER REPOSITORY

Editar o arquivo:

```
# vim /etc/pacman.conf
```

Configuração no final do arquivo:

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

Depois atualize os pacotes do sistema:

```
# pacman -Syuu
```

YAY

Instalar o pacote de dependência:

```
$ sudo pacman -S go
```

Baixar o código fonte do git:

```
$ git clone https://aur.archlinux.org/yay.git
```

Acessar a pasta criada:

```
$ cd yay
```

Finalmente, use o comando abaixo para compilar e instalar o programa:

```
$ makepkg -si
```

WINDOWS MANAGER

INSTALAÇÃO DE PACOTES

```
$ sudo pacman -S bspwm sxhkd termite dmenu picom nitrogen
```

CRIAÇÃO DOS DIRETÓRIOS

BSPWM

```
$ mkdir ~/.config/bspwm
```

SXHKD

```
$ mkdir ~/.config/sxhkd
```

Copiar os dados de configuração padrão:

```
$ cp /usr/share/doc/bspwm/examples/bspwmrc ~/.config/bspwm/
$ cp /usr/share/doc/bspwm/examples/sxhkdrc ~/.config/sxhkd/
```

Entrar no arquivo de configuração do sxhkd para mudar o terminal.

```
$ vim ~/.config/sxhkd/sxhkdrc
```

E substituir o urxvt pelo termite e substituir a teclado e atalho do dmenu.

```
$ cp /etc/X11/xinit/xinitrc ~/.xinitrc
```

Adicionar o bspwm no final do arquivo para ele ser iniciado com startx.

```
$ vim ~/.xinitrx
picom -f &
exec bspwm
```

caso você esteja em uma máquina virtual altere o arquivo picom.conf

```
$ sudo vim /etc/xdg/picom.conf
```

Procurar a opção vsync = true; e comentar essa linha.

para iniciarmos o bspwm:

```
$ startx
```

Vamos alterar agora o cursor do mouse.

```
$ xsetroot -cursor_name left_ptr
```

ADICIONAR UM BACKGROUND

1. Baixar a imagem da internet.
2. Salvar na pasta.
3. Usar o nitrogen para definir a imagem de fundo.

Definir o papel de parede para inicialização.

```
$ vim ~/.xinitrc
```

E adicionar no final no arquivo (antes do exec):

```
nitrogen -restore &
```

vamos agora tirar as bordas das janelas no arquivo de configuração do bspwm.

```
$ vim ~/.config/bspwm/bspwmrc
bspc config border_width 0
```

OBS: o programa xprop pode ser usado para saber a classe de determinado programa. Para se inserir na configuração do bspwmrc.

POLYBAR

Instalando as dependências:

```
$ sudo pacman -S polybar pacman-contrib ttf-font-awesome siji-git pulseaudio alsa-utils
```

Criar a pasta de configuração do polybar:

```
$ mkdir ~/.config/polybar
```

Pegar a configuração do polybar do site:

<https://github.com/KairoVox/Dotfiles>

e colocar ele em ~/.config/polybar/config

TECLADO

Criar e editar o arquivo:

```
# vim /etc/vconsole.conf
```

Configuração:

```
KEYMAP=us-acentos
```

TECLADO NO AMBIENTE X

Criar e editar o arquivo:

```
# vim /etc/X11/xorg.conf.d/00-keyboard.conf
```

Configuração:

```
Section "InputClass"
Identifier "system-keyboard"
MatchIsKeyboard "on"
Option "XkbLayout" "us"
Option "XkbVariant" "intl"
EndSection
```

CRIANDO AS PASTAS DE TRABALHO

Baixar o pacote

```
# pacman -S xdg-user-dirs
```

Rodar o commando:

```
$ xdg-user-dirs-update
```

PACOTES ADICIONAIS PARA NOTEBOOKS

Instalar o pacote para o touchpad e cpu:

```
# pacman -S xf86-input-libinput acpi acpid
```

CONFIGURAÇÃO DO LIBINPUT

Criar e editar o arquivo:

```
# vim /etc/X11/xorg.conf.d/30-touchpad.conf
```

Configuração:

```
Section "InputClass"
Identifier "touchpad"
Driver "libinput"
MatchIsTouchpad "on"
Option "Tapping" "on"
Option "NaturalScrolling" "true"
Option "TappingButtonMap" "1mr"
EndSection
```

Creio que essas configurações irão servir para a maioria dos usuários de notebooks, para mais configurações visite a página da wiki: <https://wiki.archlinux.org/index.php/Libinput#Configuration>

CONFIGURAÇÃO DO BLUETOOTH

Download dos pacotes:

```
# pacman -S bluez blueman bluez-utils
```

Iniciar o módulo btusb:

```
# modprobe btusb
```

Habilitar e iniciar o serviço de bluetooth:

```
# systemctl enable bluetooth && sudo systemctl start Bluetooth
```

Fazer os dispositivos se conectarem automaticamente depois do login:

```
$ sudo vim /etc/bluetooth/main.conf  
[General]  
DiscoverableTimeout = 0
```

CONFIGURAÇÃO DO SERVIÇO DE IMPRESSÃO

Instalação dos pacotes:

```
# pacman -S ghostscript cups gsfonts gutenprint libcups hplip system-config-printer
```

Configuração:

```
# systemctl enable org.cups.cupsd.service  
# systemctl start org.cups.cupsd.service
```

INSTALAÇÃO DO SSH

```
# pacman -S openssh  
# systemctl start sshd.service  
# systemctl enable sshd.service
```

INSTALAÇÃO DO FIREWALL

```
# pacman -S gufw  
# ufw enable  
# systemctl enable ufw.service
```

PERSONALIZAÇÃO

INSTALAÇÃO DE FONTES

```
$ sudo pacman -S noto-fonts ttf-ubuntu-font-family ttf-dejavu gnu-free-fonts ttf-liberation ttf-droid  
ttf-inconsolata ttf-roboto terminus-font ttf-font-awesome
```

INSTALAÇÃO DE TEMA

Baixar programa para aplicar o tema:

```
$ sudo pacman -S lxappearance
```

LXAppearance define o cursor padrão criando um arquivo ~/.icons/default/index.theme: se você editou esse arquivo manualmente, LXAppearance irá sobrescrevê-lo. Lembre-se de editar também ~/.config/gtk-3.0/settings.ini manualmente conforme especificado na especificação XDG, aplicativos como o Firefox usam essa configuração.

```
$ vim ~/.config/gtk-3.0/settings.ini
[Settings]
gtk-cursor-theme-name=cursor_theme_name
```

TEMA DO MOUSE

Eu optei por instalar o tema do ícone para todos os usuários pois assim evita diversos problemas com renderização do mouse em WM.

Editar o arquivo:

```
$sudo nvim /usr/share/icons/default/index.theme
[Icon Theme]
Inherits=capitaine-cursors
```

No meu caso o capitaine-cursors seria o tema do mouse que estou utilizando.

INICIALIZAR AUTOMATICAMENTE O X NO LOGIN

Certifique-se de que startx esteja apropriadamente configurado.

Para o Bash, adicione o seguinte ao final do ~/.bash_profile. Se o arquivo não existir, copie uma versão esqueleto de /etc/skel/.bash_profile. Para Zsh, adicione-o a ~/.zprofile.

```
if systemctl -q is-active graphical.target && [[ ! $DISPLAY && $XDG_VTNR -eq 1 ]]; then
    exec startx
fi
```

Você pode substituir a comparação -eq por uma como -le 3 (de vt1 a vt3) se quiser usar logins gráficos em mais de um terminal virtual.

INSTALAÇÃO DE PROGRAMAS

INTERNET

```
$ sudo pacman -S openvpn tor w3m firefox torbrowser-launcher apache nginx wget filezilla qbittorrent
irssi
```

MULTIMÍDIA

```
$ sudo pacman -S lame a52dec libdca libmad libmpcdec speex libvorbis faac faad2 jasper libwebp libdv
schroedinger libtheora xvidcore ogmtools feh python-cairosvg converseen gimp inkscape fontforge blender
gpick mpd ncmpcpp mpc scrot soundconverter audacity vlc handbrake kdenlive perl-image-exiftool android-
file-transfer mesa
```


GERENCIADOR DE ARQUIVOS

```
$ yay -S vifm nemo nemo-fileroller p7zip-gui kompare catfish
```

OUTROS

```
$ sudo pacman -S gnome-characters gxkb htop neofetch blueman simplescreenrecorder ghex
```