

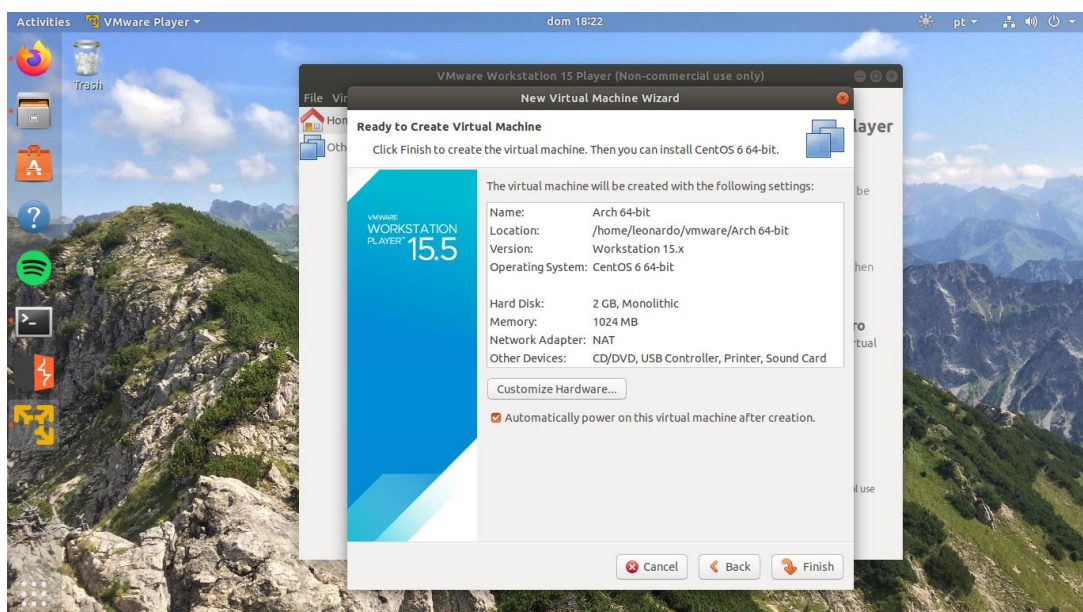
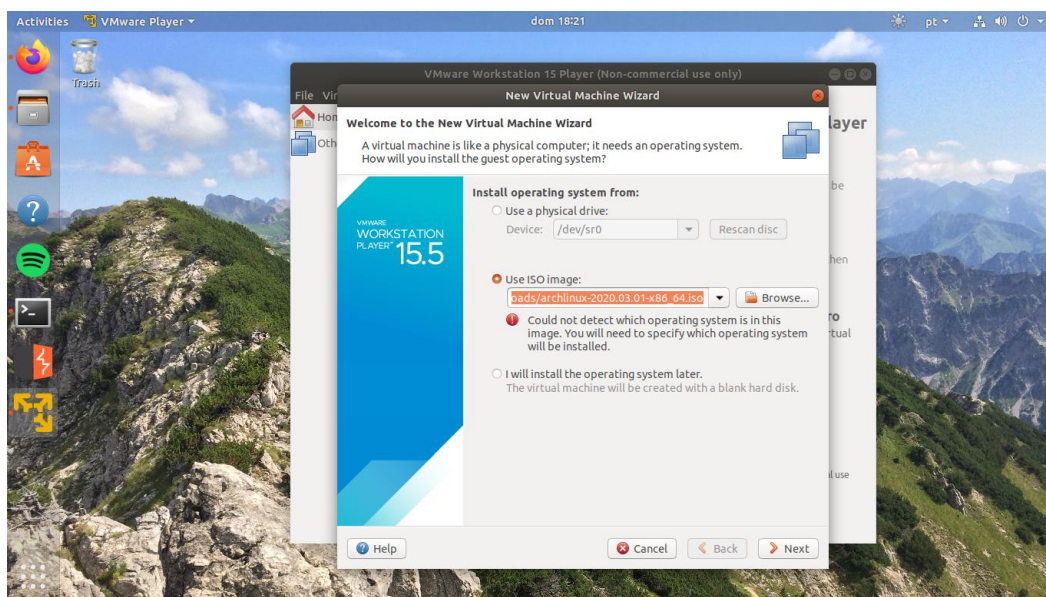
# #PROCESSO SELETIVO – GRIS 2020

**NOME:** Leonardo Andrade

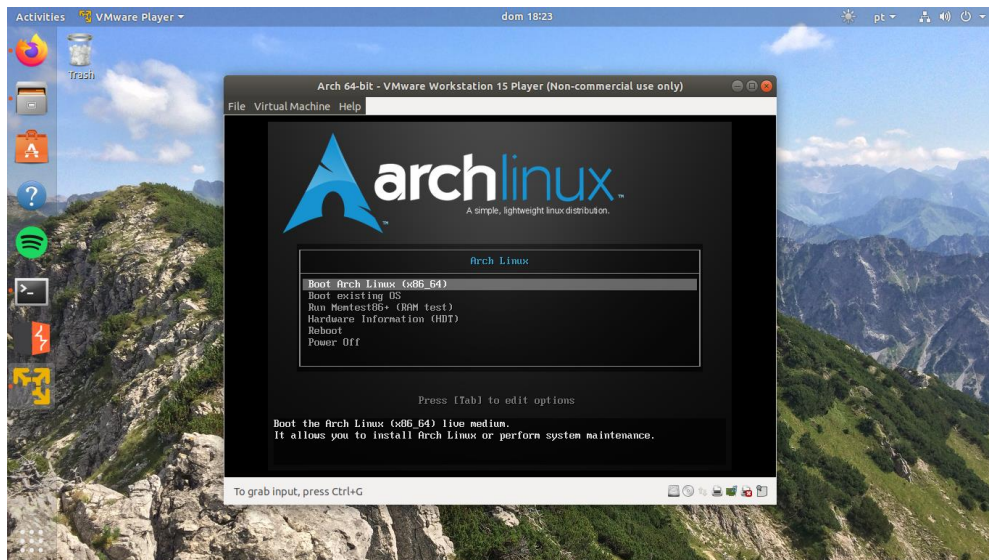
## TAG – DOCUMENTAÇÃO DE INSTALAÇÃO ARCH LINUX

Para fazer essa TAG eu decidi utilizar o VMware para instalar o Arch-Linux, tive algumas dificuldades em cima da hora as quais eu não consegui nem houve tempo suficiente para resolver, mas acredito que o todo o passo a passo, seguido corretamente.

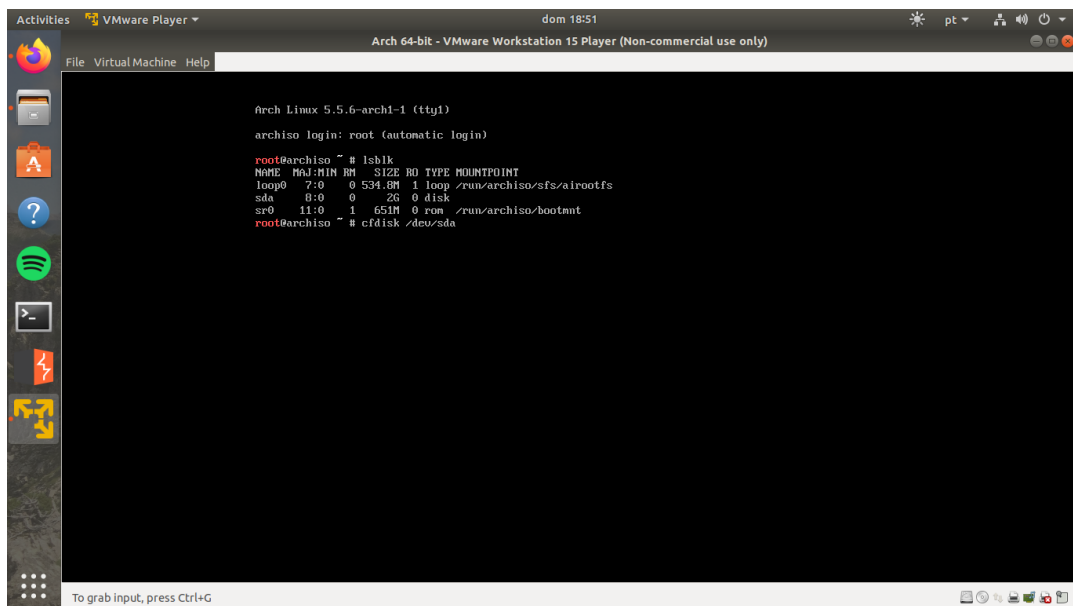
O primeiro passo é baixar a ISO da versão mais recente do Arch-Linux, podemos achar isso no próprio site <https://www.archlinux.org/download/>. Após isso, vamos criar a nossa máquina virtual com esta ISO.



Acima estão as configurações que eu designei para a máquina virtual. Por conta de alguns problemas futuros, eu precisei alterar o espaço da memória de 2GB para 3GB.

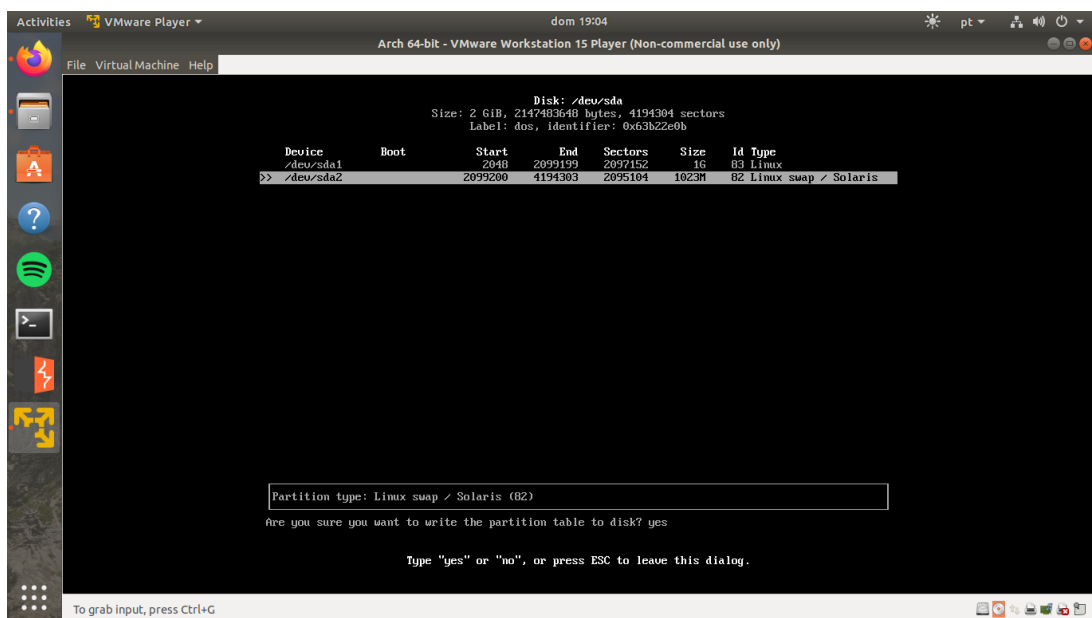
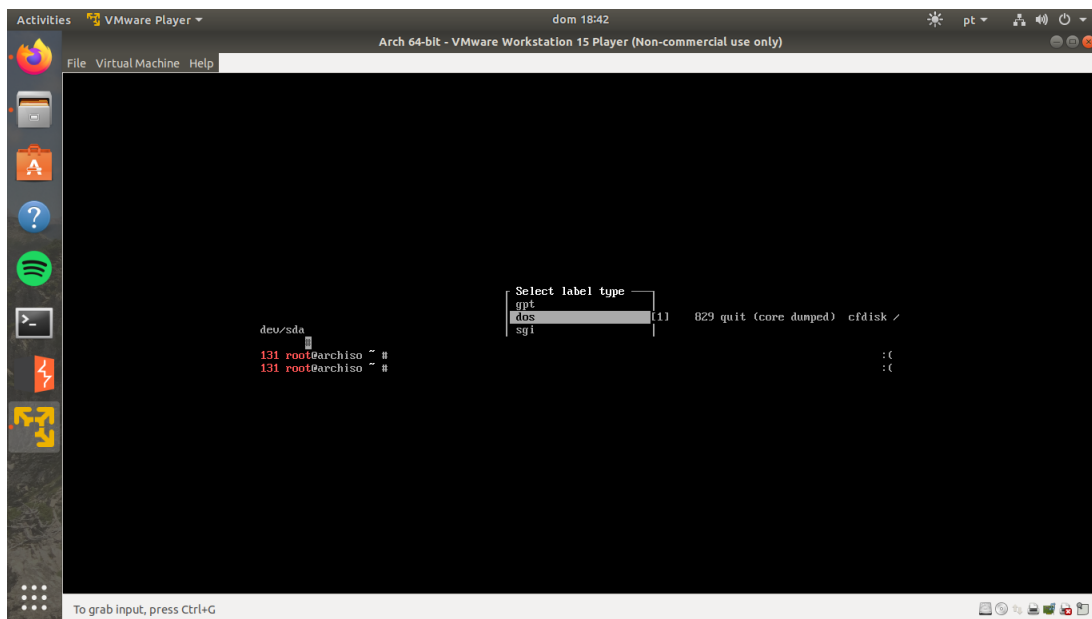


Pronto, já temos o nosso Arch Linux preparado para ser configurado, vamos iniciar e seguir adiante.



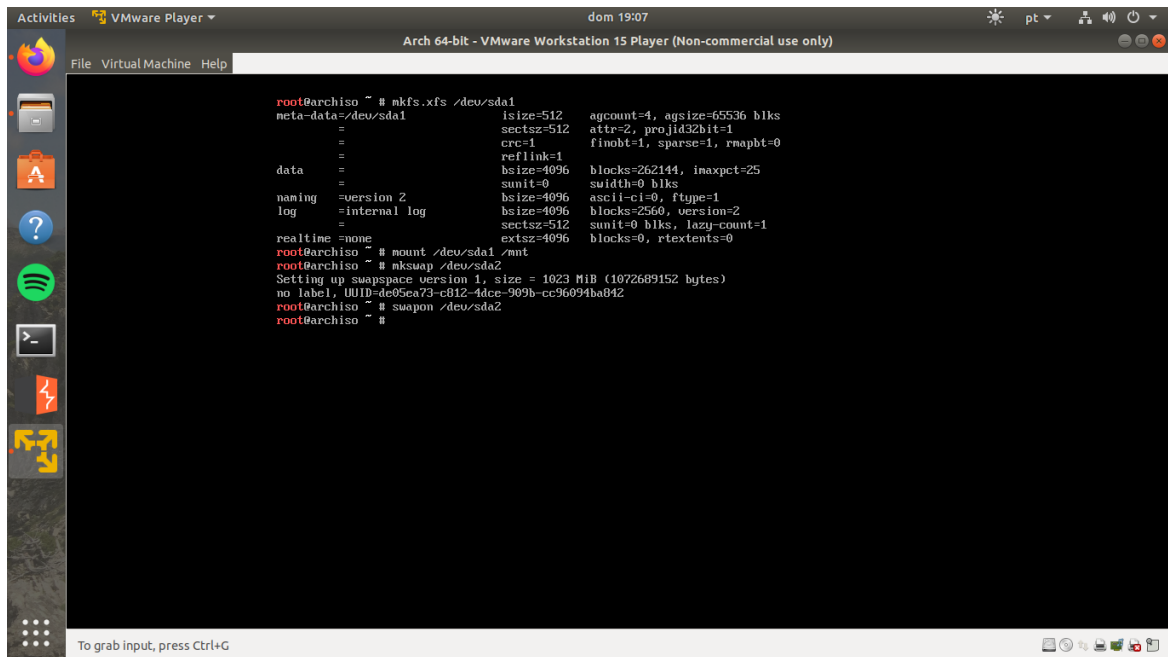
Usaremos o “lsblk” para listar dispositivos de bloco e identificar seus discos rígidos. Com o “cfdisk /dev/sda” vamos criar partições.

Vamos escolher um disco do tipo “dos” e “swap”.



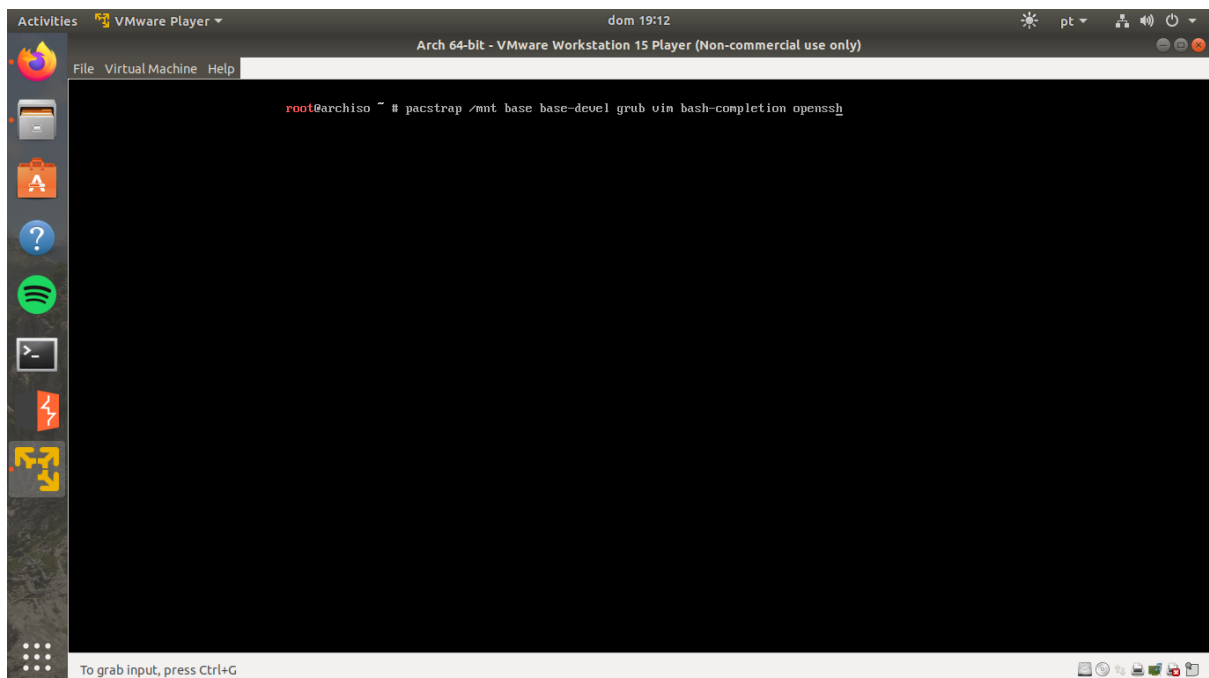
Criamos duas partições, uma “/dev/sda1” (a memória principal) e outra “/dev/sda2” (um “swap”, memória reserva). Assim que as partições tenham sido criadas, cada uma deve ser formatada com um sistema de arquivos adequado. Para uma partição para swap (por exemplo, /dev/sda2), inicializaremos com *mkswap*.

O *mkfs.xfs* constrói um sistema de arquivos XFS escrevendo em um arquivo especial usando os valores encontrados nos argumentos da linha de comando. O XFS é um sistema de arquivos de registro em diário de 64 bits e alto desempenho.

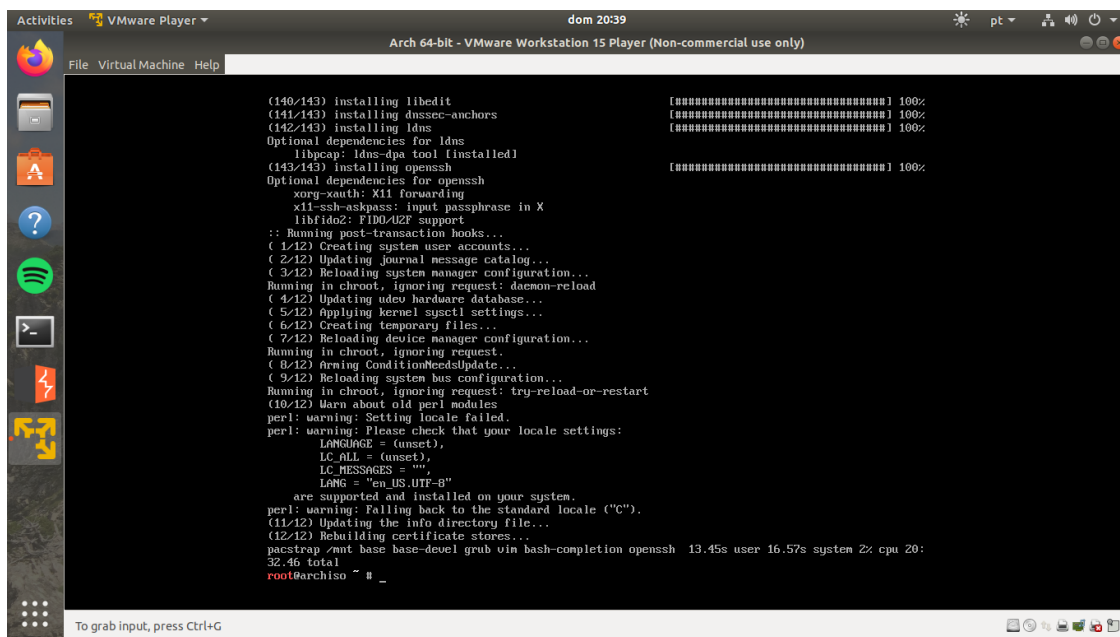


```
root@archiso ~ # mkfs.xfs /dev/sda1
meta-data=/dev/sda1          isize=512    agcount=4, agsize=65536 blks
                     =               sectsz=512   attr=2, projid32bit=1
                     =               crc=1        finobt=1, sparse=1, rmapbt=0
                     =               reflink=1
data           =               bsize=4096      blocks=262144, inapct=25
                     =               sunit=0      swidth=0 blks
naming         =version 2      bsize=4096   ascii-ci=0, ftype=1
log            =internal log   bsize=4096   blocks=2560, version=2
                     =               sectsz=512   sunit=0 blks, lazy-count=1
realtime       =none          extsz=4096      blocks=0, rtextents=0
root@archiso ~ # mount /dev/sda1 /mnt
root@archiso ~ # mkswap /dev/sda2
Setting up swapspace version 1, size = 1023 MiB (1072689152 bytes)
no label, UUID=de95ea73-c812-4dce-909b-cc96094ba042
root@archiso ~ # swapon /dev/sda2
root@archiso ~ #
```

Agora vamos instalar todos os pacotes essenciais para a utilização do sistema operacional. Pode demorar um pouco dependendo da sua conexão internet.



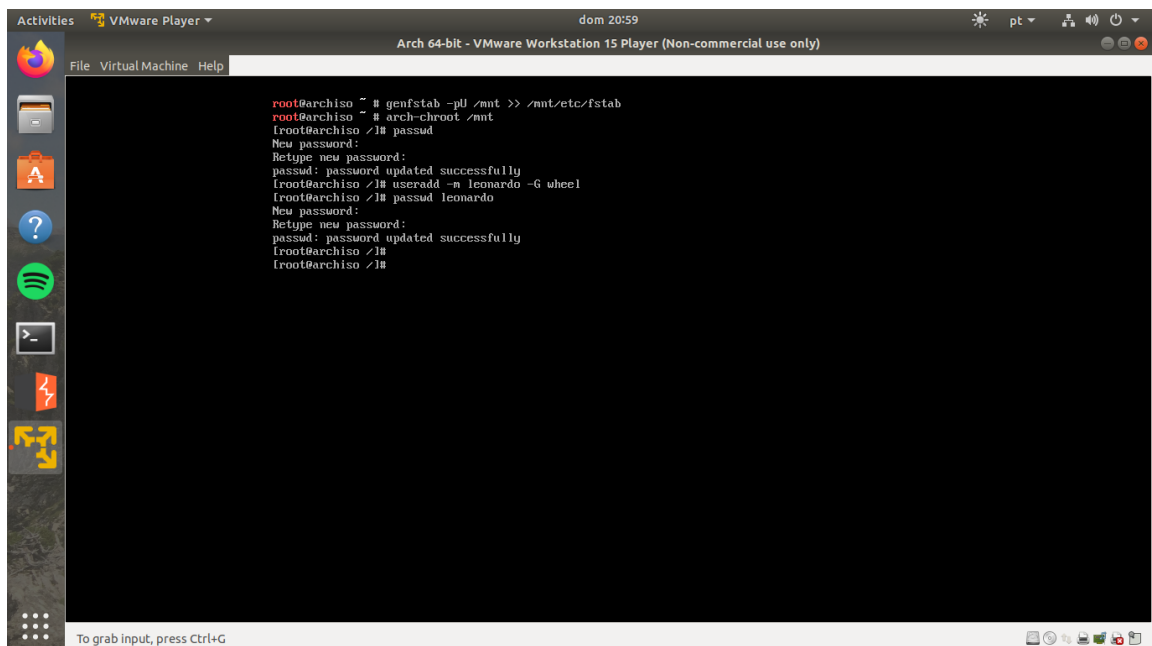
```
root@archiso ~ # pacstrap /mnt base base-devel grub uin bash-completion openssh
```



```
(140/143) installing libedit [=====] 100%
(141/143) installing dassec-anchors [=====] 100%
(142/143) installing ldns [=====] 100%
Optional dependencies for ldns
libncap: ldns-dns tool [installed]
(143/143) installing openssl [=====] 100%
Optional dependencies for openssl
xorg-xauth: X11 forwarding
x11-ssh-askpass: input passphrase in X
libfido2: FIDO/U2F support
:: Running post-transaction hooks...
(1/12) Creating system user accounts...
(2/12) Updating journal message catalog...
(3/12) Reloading system manager configuration...
Running in chroot, ignoring request: daemon-reload
(4/12) Updating udev hardware database...
(5/12) Applying kernel sysctl settings...
(6/12) Creating temporary files...
(7/12) Reloading device manager configuration...
Running in chroot, ignoring request...
(8/12) Updating ConditionNeedsUpdate...
(9/12) Reloading system bus configuration...
Running in chroot, ignoring request: try-reload-or-restart
(10/12) Warn about old perl modules
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
    LANGUAGE = (unset),
    LC_ALL = (unset),
    LC_MESSAGES = "",
    LANG = "en_US.UTF-8"
are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
(11/12) Updating the info directory file...
(12/12) Rebuilding certificate stores...
pacstrap /mnt base base-devel grub vim bash-completion openssl 13.45s user 16.57s system 2% cpu 20:
32.46 total
root@archiso ~ #
```

Agora vamos criar um “fstab” pode ser usado para definir como partições do disco, vários outros dispositivos de bloco, ou sistemas de arquivos remotos que devem ser montados no sistema. “Montagem” é a conexão de um sistema de arquivos adicional ao sistema de arquivos atualmente acessível de um computador.

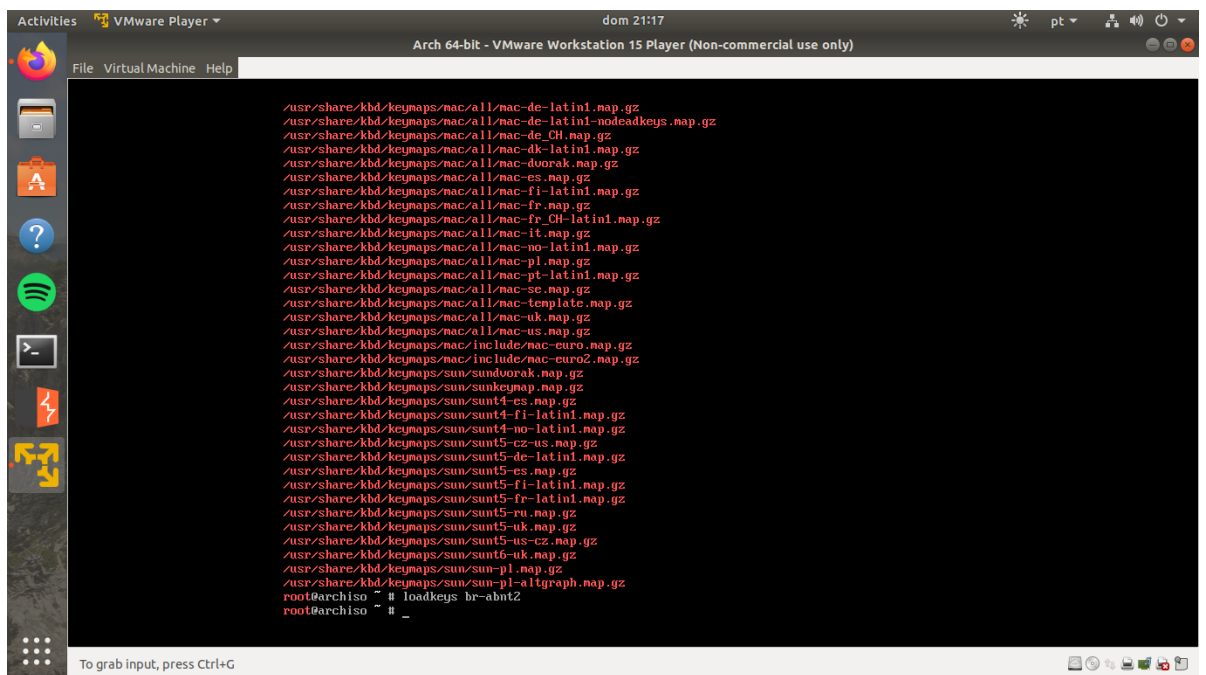
Além disso vamos definir a senha para o “root”, criar um novo usuário e senha para este.



```
root@archiso ~ # genfstab -pl /mnt >> /mnt/etc/fstab
root@archiso ~ # arch-chroot /mnt
[root@archiso /]# passwd
New password:
Retype new password:
passwd: password updated successfully
[root@archiso /]# useradd -m leonardo -G wheel
[root@archiso /]# passwd leonardo
New password:
Retype new password:
passwd: password updated successfully
[root@archiso /]#
```

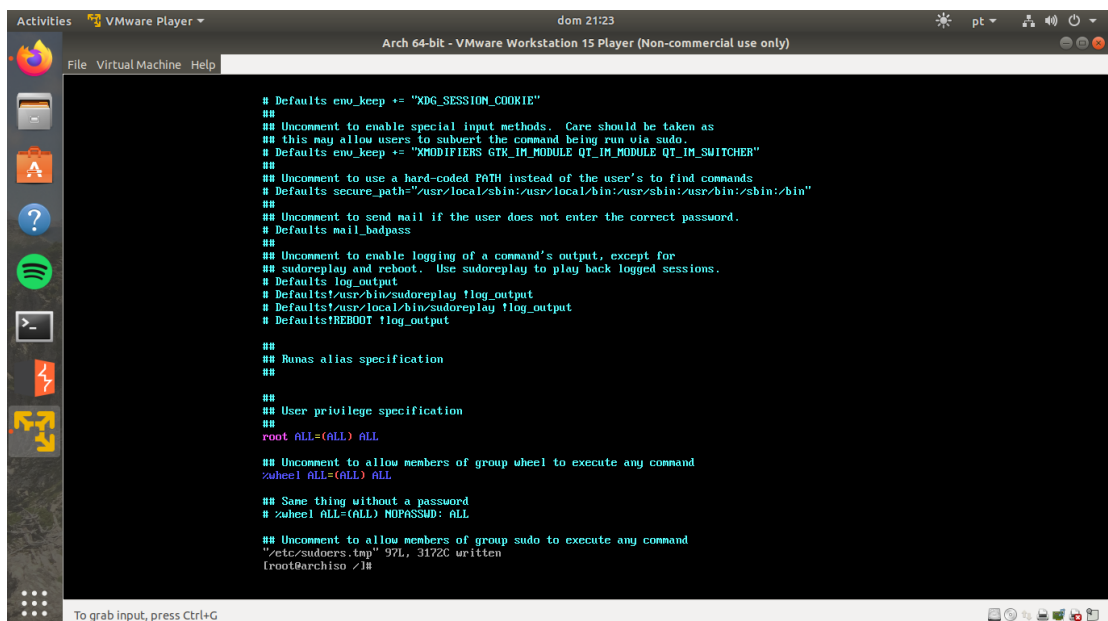
No próximo passo vamos configurar o teclado para o “abnt-2”, habitualmente utilizado no Brasil. O comando “loadkeys” bastará para isso.





```
/usr/share/kbd/keymaps/mac/all/mac-de-latin1.map.gz
/usr/share/kbd/keymaps/mac/all/mac-de-latin1-nodeadkeys.map.gz
/usr/share/kbd/keymaps/mac/all/mac-de_CH.map.gz
/usr/share/kbd/keymaps/mac/all/mac-dk-latin1.map.gz
/usr/share/kbd/keymaps/mac/all/mac-duorak.map.gz
/usr/share/kbd/keymaps/mac/all/mac-es.map.gz
/usr/share/kbd/keymaps/mac/all/mac-fi-latin1.map.gz
/usr/share/kbd/keymaps/mac/all/mac-fr.map.gz
/usr/share/kbd/keymaps/mac/all/mac-fr_CH-latin1.map.gz
/usr/share/kbd/keymaps/mac/all/mac-it.map.gz
/usr/share/kbd/keymaps/mac/all/mac-no-latin1.map.gz
/usr/share/kbd/keymaps/mac/all/mac-pl.map.gz
/usr/share/kbd/keymaps/mac/all/mac-pt-latin1.map.gz
/usr/share/kbd/keymaps/mac/all/mac-se.map.gz
/usr/share/kbd/keymaps/mac/all/mac-template.map.gz
/usr/share/kbd/keymaps/mac/all/mac-uk.map.gz
/usr/share/kbd/keymaps/mac/all/mac-us.map.gz
/usr/share/kbd/keymaps/mac/include/mac-euro.map.gz
/usr/share/kbd/keymaps/mac/include/mac-euro2.map.gz
/usr/share/kbd/keymaps/sun/sundoorak.map.gz
/usr/share/kbd/keymaps/sun/sunkeynap.map.gz
/usr/share/kbd/keymaps/sun/sunt4-es.map.gz
/usr/share/kbd/keymaps/sun/sunt4-fi-latin1.map.gz
/usr/share/kbd/keymaps/sun/sunt4-no-latin1.map.gz
/usr/share/kbd/keymaps/sun/sunt5-cz-us.map.gz
/usr/share/kbd/keymaps/sun/sunt5-de-latin1.map.gz
/usr/share/kbd/keymaps/sun/sunt5-es.map.gz
/usr/share/kbd/keymaps/sun/sunt5-fi-latin1.map.gz
/usr/share/kbd/keymaps/sun/sunt5-fr-latin1.map.gz
/usr/share/kbd/keymaps/sun/sunt5-ru.map.gz
/usr/share/kbd/keymaps/sun/sunt5-uk.map.gz
/usr/share/kbd/keymaps/sun/sunt5-us-cz.map.gz
/usr/share/kbd/keymaps/sun/sunt5-uk.map.gz
/usr/share/kbd/keymaps/sun/sun-pl.map.gz
/usr/share/kbd/keymaps/sun/sun-pl-altgraph.map.gz
root@archiso ~ # loadkeys br-abnt2
root@archiso ~ #
```

Agora vamos dar um “visudo” e permitir que os usuários do grupo “wheel” possam executar qualquer comando (lembre-se que o usuário que criamos pertence a este grupo). Para isso basta descomentar a linha que começa com %wheel.



```
# Defaults env_keep += "XDG_SESSION_COOKIE"
##
## Uncomment to enable special input methods. Care should be taken as
## this may allow users to subvert the command being run via sudo.
# Defaults env_keep += "XMODIFIERS GTK_IM_MODULE QT_IM_MODULE QT_IM_SWITCHER"
##
## Uncomment to use a hard-coded PATH instead of the user's to find commands
# Defaults secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
##
## 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
##

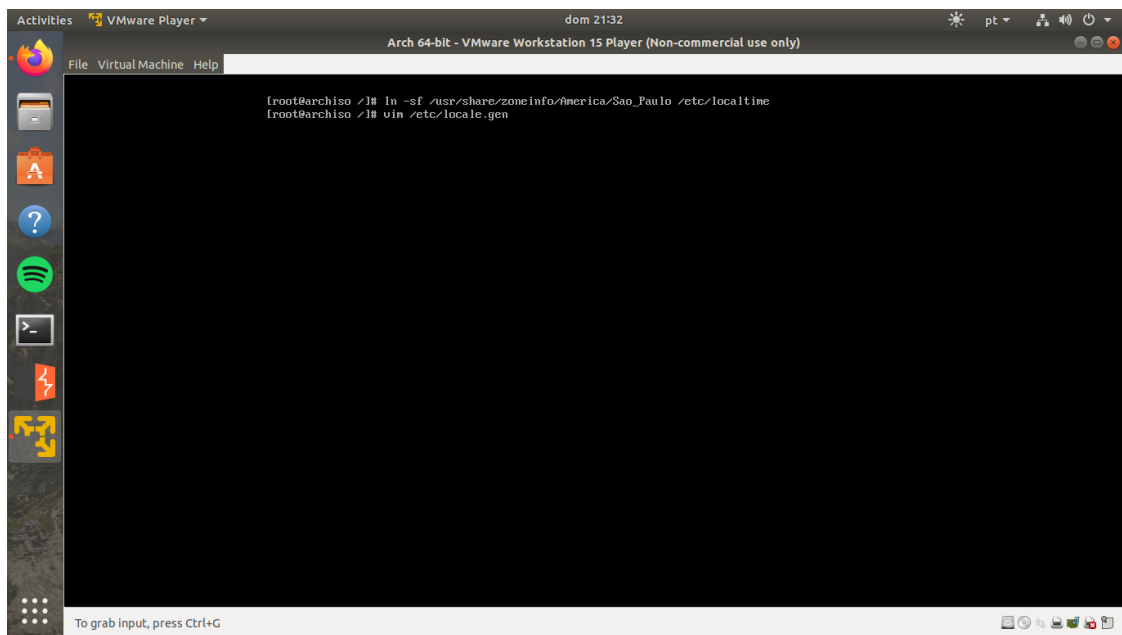
##
## User privilege specification
root ALL=(ALL) ALL

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

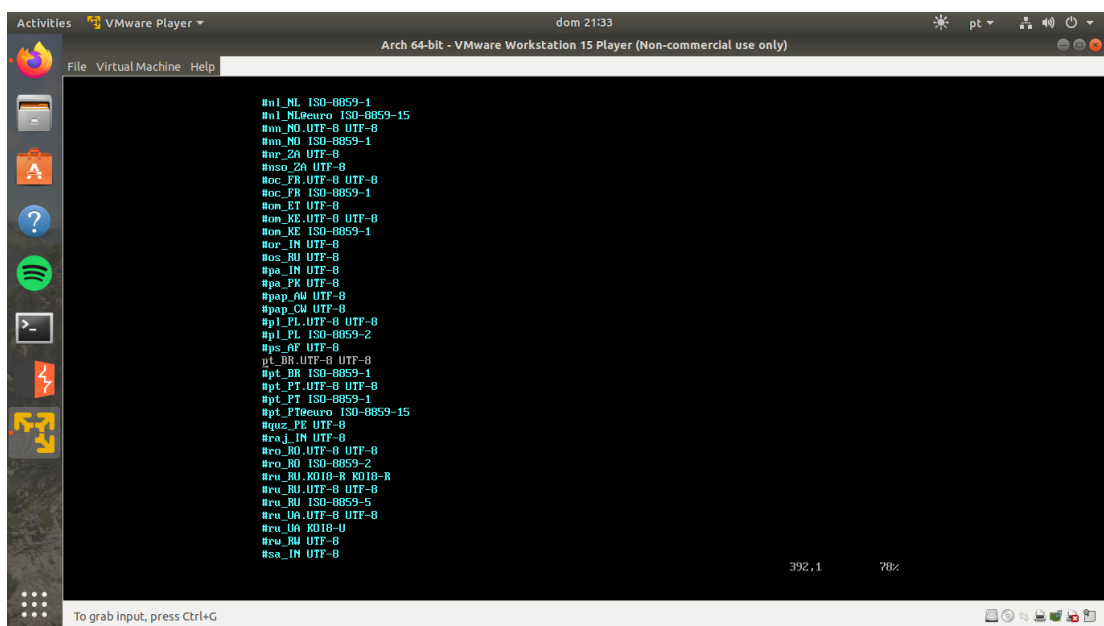
## Same thing without a password
# %wheel ALL=(ALL) NOPASSWD: ALL

## Uncomment to allow members of group sudo to execute any command
"/etc/sudoers.tmp" 97L, 3172C written
[root@archiso /#
```

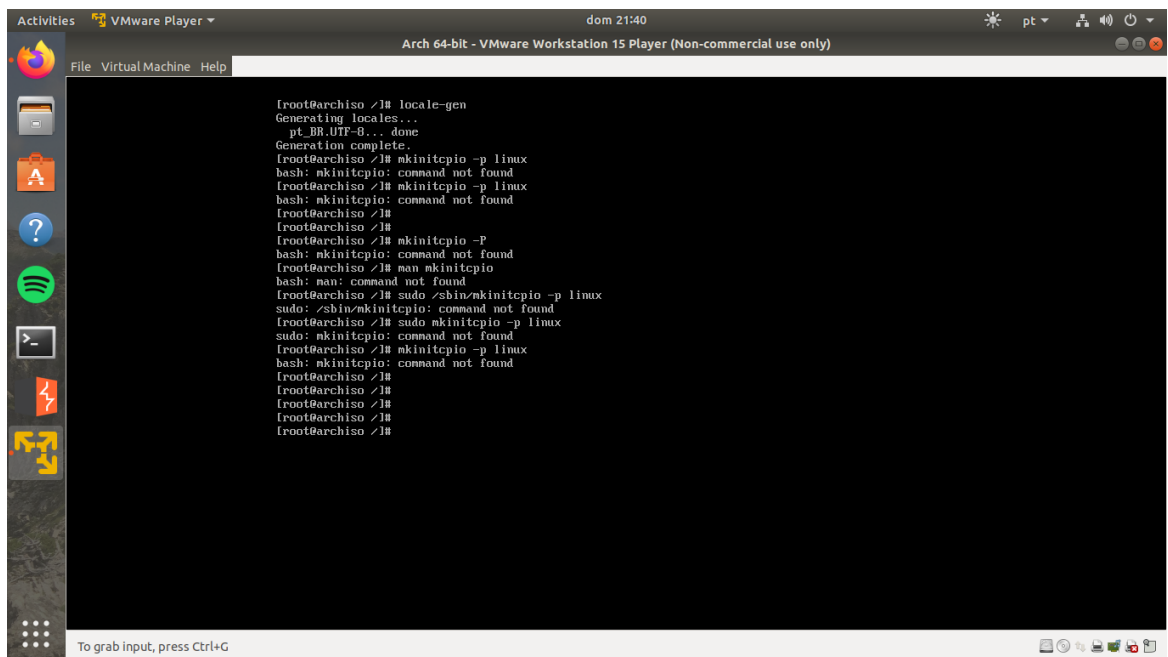
A seguir definiremos zona horária onde estamos. Definiremos São Paulo como o estado, mas o horário configurado será o de Brasília. Após isso vamos definir a linguagem que será utilizada como “locale-gen”



Basta descomentar a linha na qual está a linguagem e formatação que queremos.

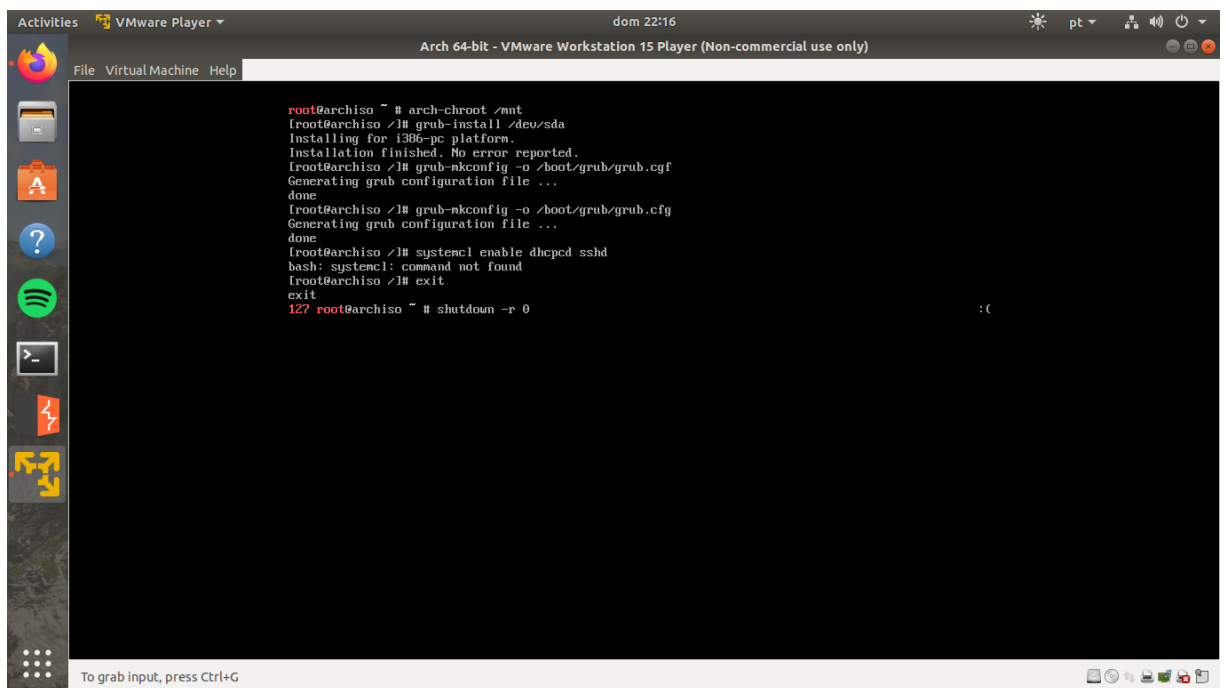


Tendo feito isso, basta utilizar o “locale-gen” para atualizar as configurações. A seguir estariam os passos finais da instalação, porém não consegui executar o “mkinitcpio”, que cria um ambiente de disco RAM para inicialização. Logo, a inicialização do sistema não foi totalmente configurada, segue os prints:



```
root@archiso /# locale-gen
Generating locales...
  pt_BR.UTF-8... done
Generation complete.
root@archiso /# mkinitcpio -p linux
bash: mkinitcpio: command not found
root@archiso /# mkinitcpio -p linux
bash: mkinitcpio: command not found
root@archiso /#
root@archiso /#
root@archiso /# mkinitcpio -P
bash: mkinitcpio: command not found
root@archiso /# man mkinitcpio
bash: man: command not found
root@archiso /# sudo /sbin/mkinitcpio -p linux
sudo: /sbin/mkinitcpio: command not found
root@archiso /# sudo mkinitcpio -p linux
sudo: mkinitcpio: command not found
root@archiso /# mkinitcpio -p linux
bash: mkinitcpio: command not found
root@archiso /#
root@archiso /#
root@archiso /#
root@archiso /#
```

Após algumas tentativas frustradas, decidi tentar configurar o grub assim mesmo.



```
root@archiso ~ # arch-chroot /mnt
root@archiso /# grub-install /dev/sda
Installing for i386-pc platform.
Installation finished. No error reported.
root@archiso /# grub-mkconfig -o /boot/grub/grub.cfg
Generating grub configuration file ...
done
root@archiso /# grub-mkconfig -o /boot/grub/grub.cfg
Generating grub configuration file ...
done
root@archiso /# systemctl enable dhcpcd sshd
bash: systemctl: command not found
root@archiso /# exit
exit
127 root@archiso ~ # shutdown -r 0
```

O grub foi instalado e configurado corretamente, após isso eu dei um reboot na máquina para ver se funcionaria normalmente, mas não foi o que aconteceu. Não me deparei com o grub, e sim com um “terminal” dele. Meu histórico de experiências com grub não são boas, mas creio que o problema está no fato de não ter executado corretamente o comando “mkinitcpio -p linux” para preparar uma memória de disco RAM para boot.



