

Arch Linux instalación con UEFI en Virtualbox



Table of Contents

1. Preparación en Virtualbox.....	3
2. Instalación del sistema Arch Linux con UEFI.....	6
1.1 Primeros comandos.....	7
1.2 GDISK.....	7
1.3 Formatear particiones.....	9
1.4 Crear puntos de montajes y montar particiones.....	12
1.5 activar SWAP.....	13
1.6 Instalación del sistema básico.....	13
1.7 Primer configuración.....	13
1.7.1 Crear archivo fstab.....	13
1.7.2 Crear archivo hostname.....	14
1.7.3 Cambiar a nuestro sistema nuevo con arch-chroot.....	14
1.7.4 Crear arranque efi.....	14
1.7.5 Modificar archivo de configuración loader.conf.....	15
1.7.6 Crear archivo de configuración arch.conf.....	15
1.7.7 Crear contraseña root.....	15
1.7.8 Terminamos por fin.....	16

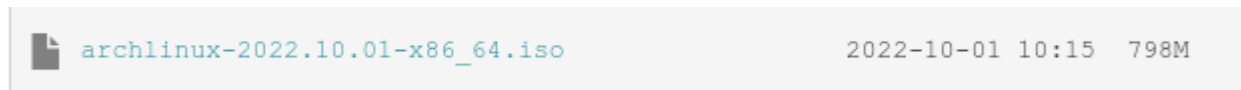
1. Preparación en Virtualbox

Descargar un iso file desde un servidor valido.

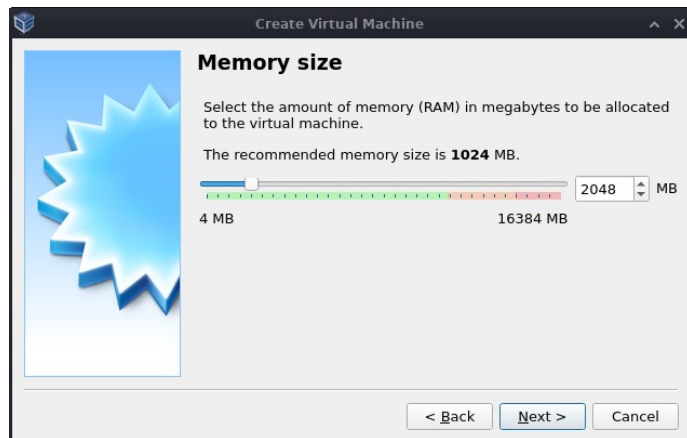
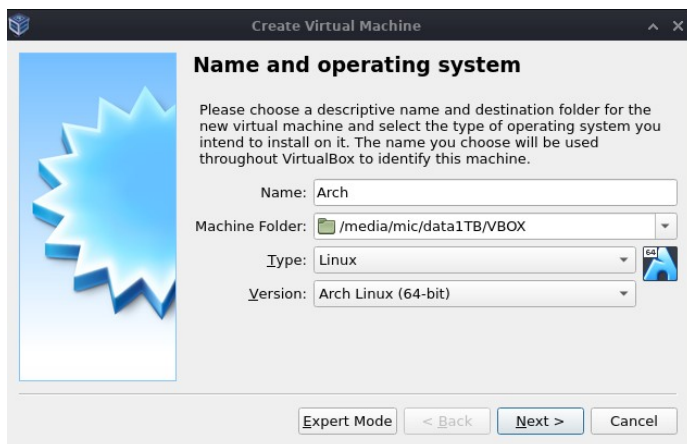
<https://archlinux.org/download/>

Servidor en españa: 18/10/2022

<https://ftp.rediris.es/mirror/archlinux/iso/2022.10.01/>



Virtualbox: Creamos una maquina virtual nueva. “ Create”



☒ Create a virtual hard disk now

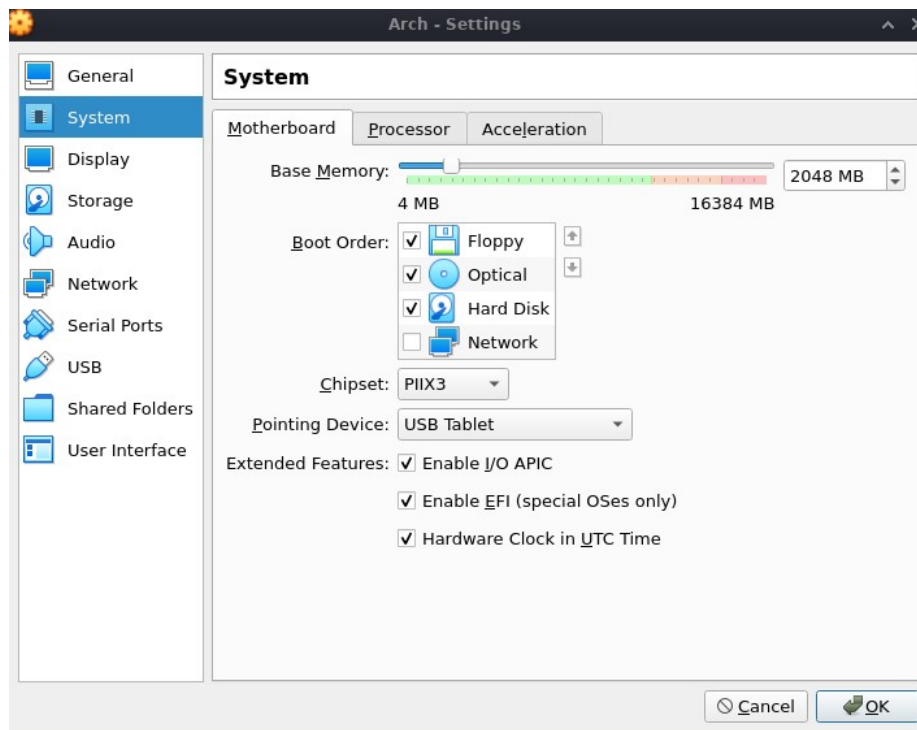
☒ VDI (VirtualBox Disk Image)

☒ **D**ynamically allocated
☐ **F**ixed size

Elije una opción que mas te conviene.

Recomiendo tamaño dinámico o fijo con mas de 20GB. Pero esto epende de tu uso.

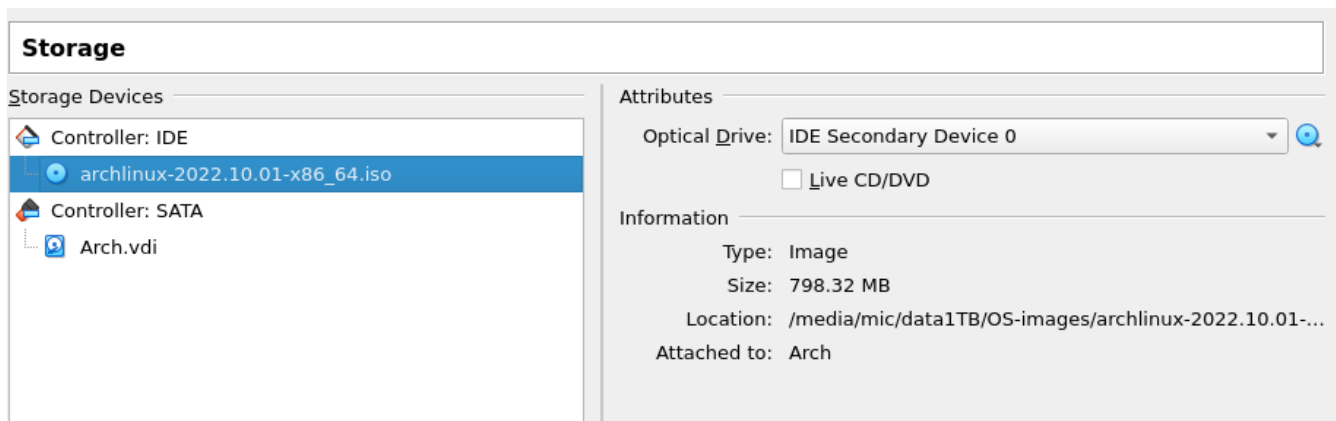
Configuración de la maquina:



Ojo: Activamos Enable EFI para esta instalación.

Algunos sistemas no permiten el arranque con EFI en Virtualbox. Te va a salir el error: "Kernel driver not installed." Este problema no forma parte de este guia.

Assignamos el archivo iso anteriormente bajado al “Storage”.

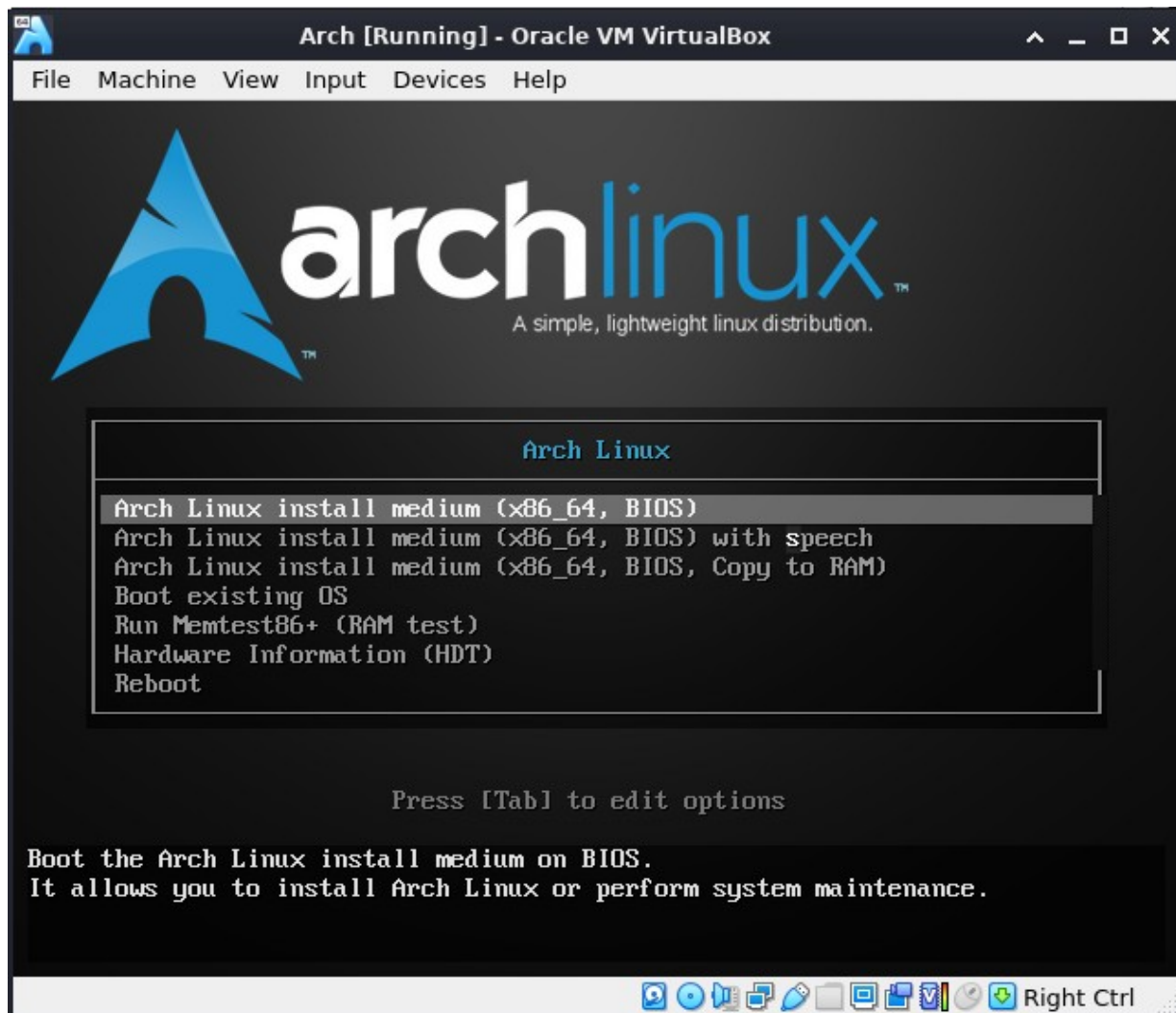


Para los nuevos en Virtualbox: Pincha el disco azul redondito a la derecha.



¡Arrancamos la maquina virtual!

2. Instalación del sistema Arch Linux con UEFI



```
Arch Linux 5.19.12-arch1-1 (tty1)

archiso login: root (automatic login)

To install Arch Linux follow the installation guide:
https://wiki.archlinux.org/title/Installation_guide

For Wi-Fi, authenticate to the wireless network using the iwctl 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.

root@archiso ~ # loadkeys es
```

1.1 Primeros comandos

Empezamos con los siguientes ordenes:

1. **# loadkeys es** → teclado español
2. **# ls /sys/firmware/efi/efivars** → archivo existe, entonces EFI activado, sino es BIOS
3. **# ip link** → controlar conexión internet , ping, etc
4. **# timedatectl set-ntp true** → hora del sistema
5. **# fdisk -l** → comprobar nombre del disco duro, normalmente */dev/sda*
6. **# gdisk /dev/sda** → arranca Manager de particiones

1.2 GDISK

n → nueva partición

```
Command (? for help): n
Partition number (1-128, default 1):
First sector (34-5860533134, default = 2048) or {+-}size{KMGTp}:
Last sector (2048-5860533134, default = 5860533134) or {+-}size{KMGTp}:
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300):
```

partición 1: +1024M (para EFI, 512 es suficiente, dicen) HEX-code: ef00
partición 2: +2048M (SWAP, tamaño de memoria RAM, con 2G va bien) HEX-code: 8200
partición 3: +10G (root partición, minimo 4G), HEX-code: 8303
partición 4: +4G (var partición, no es necesario), HEX-code: 8310
partición 5: Resto del espacio libre (home partición) , HEX-code: 8302

Instalación mínima con EFI son 3 partitiones. Aqui los primer 3.

```
Command (? for help): n
Partition number (1-128, default 1): 1
First sector (34-64049438, default = 2048) or {+-}size{KMGTp}:
Last sector (2048-64049438, default = 64049151) or {+-}size{KMGTp}: +1024M
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): 2
First sector (34-64049438, default = 2099200) or {+-}size{KMGTp}:
Last sector (2099200-64049438, default = 64049151) or {+-}size{KMGTp}: +2048M
Current type is 8300 (Linux filesystem)
Hex code or GUID (L to show codes, Enter = 8300): 8200
Changed type of partition to 'Linux swap'

Command (? for help): n
Partition number (3-128, default 3): 3
First sector (34-64049438, default = 6293504) or {+-}size{KMGTp}:
Last sector (6293504-64049438, default = 64049151) or {+-}size{KMGTp}: +20G
Current type is 8300 (Linux filesystem)
Hex code or GUID (L to show codes, Enter = 8300): 8303
Changed type of partition to 'Linux x86 root (/)'
```

```
Command (? for help): n
Partition number (4-128, default 4): 4
First sector (34-64049438, default = 48236544) or {+-}size{KMGTP}:
Last sector (48236544-64049438, default = 64049151) or {+-}size{KMGTP}: +4G
Current type is 8300 (Linux filesystem)
Hex code or GUID (L to show codes, Enter = 8300): 8310
Changed type of partition to 'Linux /var'

Command (? for help): n
Partition number (5-128, default 5): 5
First sector (34-64049438, default = 56625152) or {+-}size{KMGTP}:
Last sector (56625152-64049438, default = 64049151) or {+-}size{KMGTP}:
Current type is 8300 (Linux filesystem)
Hex code or GUID (L to show codes, Enter = 8300): 8302
Changed type of partition to 'Linux /home'
```

Finalmente terminamos el particionado:

```
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_
```

1.3 Formatear particiones

```
mkfs.fat -F 32 -n BOOT /dev/sda1 ('n' es Label en caso de fat, normalmente es 'L')
mkswap -L SWAP /dev/sda2
mkfs.ext4 -L ROOT /dev/sda3
mkfs.ext4 -L VAR /dev/sda4
mkfs.ext4 -L HOME /dev/sda5
```

```
root@archiso ~ # mkfs.fat -F 32 -n BOOT /dev/sda1
mkfs.fat 4.2 (2021-01-31)
root@archiso ~ # mkswap -L SWAP /dev/sda2
Setting up swap space version 1, size = 2 GiB (2147479552 bytes)
LABEL=SWAP, UUID=8662d07c-cabf-40b6-861d-d1be6dacbade
root@archiso ~ # mkfs.ext4 -L ROOT /dev/sda3
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 5242880 4k blocks and 1310720 inodes
Filesystem UUID: 141de366-d3c8-43d6-b80e-c0f9646d3b2a
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

root@archiso ~ # mkfs.ext4 -L VAR /dev/sda4
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 1048576 4k blocks and 262144 inodes
Filesystem UUID: 38186403-1d02-4ad4-b2d0-794465a3061c
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

root@archiso ~ # mkfs.ext4 -L HOME /dev/sda5
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 928000 4k blocks and 232000 inodes
Filesystem UUID: 706d654b-0451-418c-92c4-fa8383364626
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```


1.4 Crear puntos de montajes y montar particiones

Trabajamos con Labels 'L':

`mount -L ROOT /mnt` → Root partition directamente montado en /mnt

`mkdir /mnt/boot` → crear carpeta boot

`mount -L BOOT /mnt/boot` → montar partition efi en boot.

`mkdir /mnt/var`

`mount -L VAR /mnt/var`

`mkdir /mnt/home`

`mount -L HOME /mnt/home`

1.5 activar SWAP

`swapon -L SWAP` (o `swapon dev/sda2`)

1.6 Instalación del sistema básico

Comando `pacstrap` instala paquetes.

root@archiso ~# `pacstrap /mnt base base-devel linux linux-firmware gptfdisk efibootmgr bash-completion vim lvm2 networkmanager`

(en vez de `networkmanager` p.E. `dhcpcd`, `lvm2` depende.

No puedo vivir sin vim. Nano ya está integrado en el paquete “base”.)

¡Arch Linux está instalado!



Pero, ... no va aún.



1.7 Primer configuración

1.7.1 Crear archivo fstab

`genfstab -Lp /mnt >> /mnt/etc/fstab`

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

# <file system> <dir> <type> <options> <dump> <pass>
# /dev/sda3 UUID=141de366-d3c8-43d6-b80e-c0f9646d3b2a
LABEL=ROOT / ext4 rw,relatime 0 1

# /dev/sda1 UUID=E774-F88F
LABEL=BOOT /boot ufat rw,relatime,fmask=0022,dm
=mixed,utf8,errors=remount-ro 0 2

# /dev/sda4 UUID=38186403-1d02-4ad4-b2d0-794465a3061c
LABEL=VAR /var ext4 rw,relatime 0 2

# /dev/sda5 UUID=706d654b-0451-418c-92c4-fa8383364626
LABEL=HOME /home ext4 rw,relatime 0 2

# /dev/sda2 UUID=8662d07c-cabf-40b6-861d-d1be6dadbade
LABEL=SWAP none swap defaults 0 0
```

1.7.2 Crear archivo hostname

`echo myhost > /mnt/etc/hostname`

1.7.3 Cambiar a nuestro sistema nuevo con arch-chroot

`# arch-chroot /mnt`

```
root@archiso /mnt/etc # arch-chroot /mnt
[root@archiso /]#
```

Estamos ya en nuestra consola del nuevo sistema.

1.7.4 Crear arranque efi

No hace falta instalar un bootloader. Tenemos que decir al systemd que arranca en modo efi.

Commando: `#bootctl install`.

```
[root@archiso /]# bootctl install
Created "/boot/EFI".
Created "/boot/EFI/systemd".
Created "/boot/EFI/BOOT".
Created "/boot/loader".
Created "/boot/loader/entries".
Created "/boot/EFI/Linux".
Copied "/usr/lib/systemd/boot/efi/systemd-bootx64.efi" to "/boot/EFI/systemd/systemd-bootx64.efi".
Copied "/usr/lib/systemd/boot/efi/systemd-bootx64.efi" to "/boot/EFI/BOOT/BOOTX64.EFI".
Random seed file /boot/loader/random-seed successfully written (32 bytes).
Not installing system token, since we are running in a virtualized environment.
Created EFI boot entry "Linux Boot Manager".
[root@archiso /]#
```

1.7.5 Modificar archivo de configuración loader.conf

`vim /boot/loader/loader.conf`

```
[root@archiso /]# vim /boot/loader/loader.conf
```

borrar el contenido y reemplazar con

```
default arch
timeout 3
editor 0
```

```
default arch
timeout 3
editor 0
```

editor 0 → Es importante para no pasar la contraseña de root al arrancar el sistema.

1.7.6 Crear archivo de configuración arch.conf

`vim boot/loader/entries/arch.conf`

```
title      Arch Linux
linux      /vmlinuz-linux
initrd     /initramfs-linux.img
options    root=LABEL=ROOT rw
```

```
[root@archiso ~]# cat /boot/loader/entries/arch.conf
title      Arch Linux
linux      /vmlinuz-linux
initrd     /initramfs-linux.img
options    root=LABEL=ROOT rw
[root@archiso ~]#
```

1.7.7 Crear contraseña root

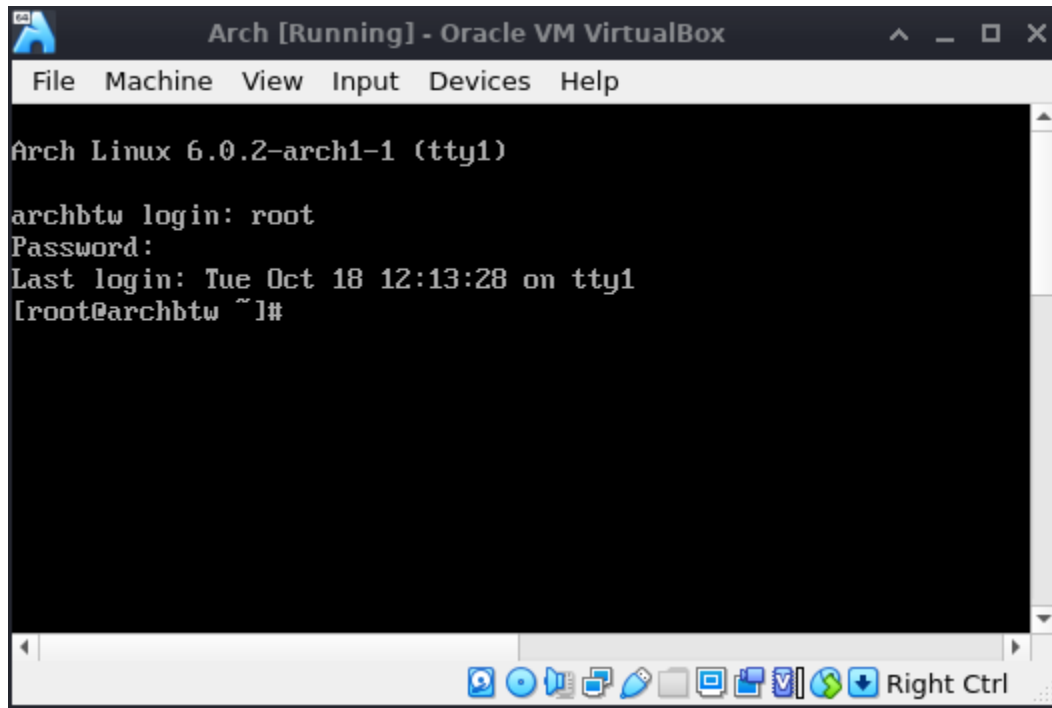
Poner root password: `passwd` → No necesario pero mejor haberlo hecho ya. Si no se pone, se queda vacío y el usuario puede conectarse sin contraseña.

Ojo: no eliges nada raro de contraseña al principio. Tenemos que configurar el teclado definitivamente mas adelante.

1.7.8 Terminamos por fin

`# exit`

`# reboot`



Ahora estamos en nuestra nueva instancia de Arch Linux.

No olvides de nuevo el teclado: `loadkeys` es
y empezamos a configurar. Teclado, red, desktop, server,

Mas aquí:

[https://wiki.archlinux.org/title/Arch_Linux_\(Español\)](https://wiki.archlinux.org/title/Arch_Linux_(Español))