

# My first Arch Linux rice

mvrleo

February 2019

## 1 Requirements

- [VirtualBox](#) virtual machine manager.
- [An Arch Linux ISO](#)

## 2 Resources

This guide is heavily based on the following tutorials and is what worked for me:

- [Arch Linux wiki page](#).
- [A youtube tutorial](#).
- [This](#) gist.
- [This](#) article
- [This](#) articles for things to do after the install.

## 3 System Setup

### 3.1 Disk partitioning

#### 3.1.1 Create partitions

I attempted setting up Arch Linux in a virtual machine (VM). After creating the hard disk and loading the ISO in the VM manager, start the Arch image and select Boot Arch Linux



Figure 1: Arch Linux boot menu.

It will log in as root so the setup can begin. First, verify that the hard disk has been created correctly. Type

```
fisk -l
```

This shows the hard disk partitions as device blocks under /dev. Since there is currently only one partition, it should show something like /dev/sda or /dev/sdb etc:

```
Disk /dev/sda: 21 GiB, 22548578304 bytes, 44040192 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
```

```
Disk /dev/loop0: 459.6 MiB, 481939456 bytes, 941288 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
```

```
Disk /dev/sdb: 7.3 GiB, 7851737088 bytes, 15335424 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
Disklabel type: dos
Disk identifier: 0x5bcc3706
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1	*	8064	15335423	15327360	7.3G	7	HPFS/NTFS/exFAT

So we want to partition /dev/sda.

```
fdisk /dev/sda
```

fdisk will bring up a menu. We will create three primary partitions for:

- root (things that linux installs incl. the OS itself) - 12/21 Gb
- home (where files and data are stored) - 8/21 Gb
- swap (space in hard driver that can be used as backup RAM memory when the computer runs out of it) - 1/21 Gb

Follow fdisk's instructions and remember that o initialises a new partition, n followed by either p or e sets its type (primary or extended) and size can be specified e.g. by +12G. Also, make sure to make the root partition bootable by pressing a in fdisk after selecting its size. For the swap partition, press **type** after selecting its size. We want this to be used for swap memory so enter 82. All types can be listed from the help menu. Finally, press w to finalise the changes.

Now the output of fdisk -l should contain something similar to the following:

```
Disk /dev/sda: 21 GiB, 22548578304 bytes, 44040192 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
Disklabel type: dos
Disk identifier: 0x22b02a40
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sda1	*	2048	25167871	25165824	12G	83	Linux
/dev/sda2		25167872	27265023	2097152	1G	82	Linux swap / Solaris
/dev/sda3		27265024	44040191	16775168	8G	83	Linux

```
Disk /dev/loop0: 459.6 MiB, 481939456 bytes, 941288 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
```

```
Disk /dev/sdb: 7.3 GiB, 7851737088 bytes, 15335424 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
Disklabel type: dos
Disk identifier: 0x5bcc3706
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1	*	8064	15335423	15327360	7.3G	7	HPFS/NTFS/exFAT

### 3.1.2 Format partitions

We're back at the root shell and 3 partitions have been created. We want /dev/sda1 and /dev/sda3 to be ext4 filesystems. Simply do

```
mkfs.ext4 /dev/sda1
mkfs.ext4 /dev/sda3
```

Next, format partition /dev/sda2 to be swap memory:

```
mkswap /dev/sda2
swapon /dev/sda2
```

### 3.1.3 Internet setup

Check if an external host can be reached:

```
ping -c 4 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=122 time=9.72 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=122 time=8.50 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=122 time=9.26 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=122 time=9.04 ms

--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 9ms
rtt min/avg/max/mdev = 8.501/9.128/9.717/0.442 ms
```

Ping was working so I skipped this setup. If not, see the [Arch wiki](#). Another useful command, to find out the name of the internet interface, e.g. wlan0, enp0s3 etc. is:

```
ip link
```

It is suggested to note it in case the internet stops working. Check also the `wifi-menu` command which automated a lot on internet-related things.

## 3.2 Essential package installation

To install things, we need to temporarily mount the partition device blocks into the actual filesystem. Do

```
mkdir /mnt/home  
mount /dev/sda1 /mnt  
mount /dev/sda3 /mnt/home
```

The output of the mount command should now look similar to this

```
/dev/sda1 on /mnt type ext4 (rw,relatime)  
/dev/sda3 on /mnt/home type ext4 (rw,relatime)
```

Go to root and call the `pacstrap` package manager to install the minimum packages for the system. When prompted, install them all.

```
pacstrap -i /mnt base
```

That resulted in files with unknown signature. Update key ring and refresh local package database and try again:

```
pacman -Sy archlinux-keyring && pacman -Syyu  
pacstrap -i /mnt base
```

To automatically what's in `/mnt` from now on, write the changes in the `fstab` file (found at `<root>/etc/fstab`):

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

Now we can preview and modify the install by entering into it:

```
arch-chroot /mnt
```

## 3.3 Installing the packages

We're still in the chroot environment. Install some basic packages with `pacman`:

```
pacman -S openssh grub-bios linux-headers linux-lts linux-lts-headers wpa_supplicant wireless_tools
```

## 3.4 Post-install configuration

### 3.4.1 Locale configuration

see also: [https://www.youtube.com/watch?v=U3nnu\\_PGfy8](https://www.youtube.com/watch?v=U3nnu_PGfy8)

Open the language configuration to select locale(s) that can be used:

```
vi /etc/locale.gen
```

Uncomment the desired ones, in my case `#en_GB.UTF-8 UTF-8 -> en_GB.UTF-8 UTF-8`. Now the default system language needs to be set. Take a note of the language to use, in this case `en_GB.UTF-8 UTF-8` and export it to file `/etc/locale.conf` (needs to be created):

```
echo LANG=en_GB.UTF-8 > /etc/locale.conf  
export LANG=en_GB.UTF-8
```

Finally, make the desired keyboard layout default and permanent:

```
vi /etc/vconsole.conf  
KEYMAP=uk
```

### 3.4.2 Time configuration and password

Create a shortcut for the right time zone under `/etc/localtime` and then save the clock:

```
ln -s /usr/share/zoneinfo/Europe/Berlin /etc/localtime  
hwclock --systohc --utc
```

Also, automatically start ssh service (optional):

```
systemctl enable sshd.service
```

Finally, create a password for the persistent system:

```
passwd
```

Before adding a new user, the `sudo` package needs to be installed in order to be able to give them `sudo` permissions:

```
pacman -S sudo
```

A new user should also be added and allocated a password, as follows:

```
useradd -m username -G wheel  
passwd username
```

To complete the user registration, add them in the `sudoers` list:

```
vi /etc/sudoers
```

Uncomment the line that allows the wheel group (where the user was added) to run sudo commands, so vi /etc/sudoers file should contain the following line:

```
%wheel ALL=(ALL) ALL
```

Finally, set the machine's hostname (in this case it's "archlinux"):

```
echo archlinux > /etc/hostname  
echo -e "127.0.1.1\tarchlinux" >> /etc/hosts
```

The reason the last IP was chosen is because 127.0.0.1 is already taken.

### 3.5 Install bootloader

We use grub which was installed through grub-bios. Configure it and make it:

```
grub-install --target=i386-pc --recheck /dev/sda  
cp /usr/share/locale/en\@quot/LC_MESSAGES/grub.mo /boot/grub/locale/en.mo  
grub-mkconfig -o /boot/grub/grub.cfg
```

If everything is successful, grub should have found the linux, initrd and their fallback images. Now, don't forget to exit chroot:

```
exit
```

Then unmount the partitions and reboot, and login as root with the previously given password

```
umount /mnt/home  
umount /mnt  
reboot
```

On a USB boot, don't forget to remove the USB after executing "reboot". Done! If the internet is down, an easy way to bring it up for the current session is:

```
ip link # note wireless interface name, e.g. enp0s3  
dhcpcd enp0s3
```

Now ricing can begin!

## 4 Ricing

### 4.1 Moar packages!

Before installing a window manager, make sure the system is equipped with some basic packages – some required for the window manager itself and some note:

- xorg-server, xorg-xinit, xorg-apps – window server
- git, gcc, gdb, cmake, qemu, python, python-pip, vim – for development
- firefox, wget, ffmpeg, neofetch, xf86-input-synaptics, evince, or okular, or zathura for pdfs, feh, sudo – utilities
- intel-ucode – CPU firmware updates, [needs grub update, i.e. grub-mkconfig -o /boot/grub/grub.cfg](#), alsa-utils – control audio volume
- rxvt-unicode, termite – terminal emulators (choose which one you want later) supporting unicode and the latter supports emojis
- i3status, dmenu, xbacklight, conky – the window manager accessories, they're required.
- i3-wm

### 4.2 Start i3

**NOTE:** Never run X and other GUI programs as root!

Before attempting the following sections, **log in as the user** that was created during setup. We need to tell xorg what to do when it starts. xorg is started by typing startx. To start i3 as soon as X starts, just edit the ~/.xinitrc file to do this:

```
vim ~/.xinitrc
#!/bin/bash
exec i3
```

Then type startx to start the graphical window manager. Accept the two i3 prompts to create a new configuration. To start a terminal, press <Mod>+<Enter>, where the Mod key, or “super” key is usually the windows key. More shortcuts explained [here](#). Press <Mod>+<L/R Arrow> to switch windows, <Mod>+q to close a window. We'd like to X (startx command) after logging in in the default terminal tty1. Open file /etc/profile and append this:

```
# autostart systemd default session on tty1
if [[ "$(tty)" = '/dev/tty1' ]]; then
    exec startx
fi
```

Note that other tty sessions won't be affected.

### 4.3 Modifying locale again

Select language e.g. UK English by setting the /etc/locale.conf as follows:

```
LANG=en_GB.UTF-8
LANGUAGE=en_GB.UTF-8
LC_ADDRESS=en_GB.UTF-8
LC_COLLATE=en_GB.UTF-8
LC_CTYPE=en_GB.UTF-8
LC_IDENTIFICATION=en_GB.UTF-8
LC_MEASUREMENT=en_GB.UTF-8
LC_MESSAGES=en_GB.UTF-8
LC_MONETARY=en_GB.UTF-8
LC_NAME=en_GB.UTF-8
LC_NUMERIC=en_GB.UTF-8
LC_PAPER=en_GB.UTF-8
LC_TELEPHONE=en_GB.UTF-8
LC_TIME=en_GB.UTF-8
```

Then do `locale-gen` again.

## 4.4 (For VBox only) Increase screen size

Use `xrandr` installed before by `xorg` package

```
xrandr # to list available configs
xrandr --output VGA-1 --mode 1600x900 --rate 60 # to set
```

## 4.5 Editing the status bar

`i3` looks at these files by the following order when configuring the status bar:

```
~/.config/i3status/config (or $XDG_CONFIG_HOME/i3status/config if set)
/etc/xdg/i3status/config (or $XDG_CONFIG_DIRS/i3status/config if set)
~/.i3status.conf
/etc/i3status.conf
```

Edit the first one. [Here](#) are some instructions on how to set the status bar.

## 4.6 customize terminal emulator

`termite` and `urxvt` have been installed as terminal emulators. We want the first one as the default. Open the `i3` config file at:

```
~/.config/i3/config
```

show changes here

to add extra customization to command line, particularly transparency since `termite` doesn't support it anymore, install `compton`.

```
echo 'opacity-rule = ["85:class_g = 'Termite'"]; ' >> ~/.compton.conf
compton -b --config ~/.compton.conf
```

The 16 terminal colours can be changed from `Termite`'s config.



## 4.7 Change theme, icons and install a file explorer

lxappearance can be installed as the theme selector. A nice dark theme is arc-dark, which requires the gtk2 backend. Now lxappearance can be launched to select theme.

A lightweight graphical file explorer is thunar. However, it doesn't render thumbnails on its own so it needs tumbler.

A terminal file explorer that support Vim key bindings, image/pdf previews, etc. is ranger.

## 4.8 Pretty fonts

For system-wide instructions, follow. [https://www.reddit.com/r/archlinux/comments/5r5ep8/make\\_your\\_arch\\_fonts\\_beautiful\\_easily/](https://www.reddit.com/r/archlinux/comments/5r5ep8/make_your_arch_fonts_beautiful_easily/) Otherwise, terminal font can be modified from the i3 config file.

### 4.8.1 System monitoring on the desktop

Install conky using pacman. To display a sample config file, do:

```
conky -C
```

This can be tailed to run in the background and be attached to the desktop. When done, the config changes can take place by issuing

```
conky -c ~/.config/conky/conky.conf
```

## 4.9 Other issues

### 4.9.1 Troubleshooting display

If GTK or another backend cannot open the firefox or another program that runs graphically with a message similar to "Display not found", make sure to set the display correctly and have something like

```
export DISPLAY=:0
```

Next, allow the user to have authority in the X session by:

```
cp /root/.Xauthority /home/username/.Xauthority
```

That should resolve the issue.

### 4.9.2 Troubleshooting audio; no audio

Sometimes, post-install, Arch mutes audio by default. To check if that's the case, launch alsamixer and unmute it. Otherwise, try the following.

```
# make sure pulseaudio daemon running
pulseaudio -D
# lists sounds cards, shouldn't be empty
aplay -l
# load sound controller module
modprobe snd-hda-intel
useradd -a -G audio <username>
```

snd-hda-intel is a universal audio driver and more about it can be found [here](#).

## A Appendices

### A.1 Useful packages

This is a table of neat packages I found, some installed with the standard package manager and some from the [AUR](#) (Arch User Repository). To install from AUR, a package manager such as yaourt is required.

Package list			
Package Name	Installed by	Description	Documentation/ Examples
yaourt	from git	installs packages not in the official repository	<ul style="list-style-type: none"> <li>• <a href="https://www.ostechnix.com/install-yaourt-arch-linux/">https://www.ostechnix.com/install-yaourt-arch-linux/</a></li> </ul>
python-eyed3	yaourt	music tag editing	<ul style="list-style-type: none"> <li>• <a href="https://eyed3.readthedocs.io/en/latest/">https://eyed3.readthedocs.io/en/latest/</a></li> </ul>
cmatrix	pacman	Matrix-themed terminal screen-saver	<ul style="list-style-type: none"> <li>• Just run cmatrix</li> </ul>
neofetch	pacman	Fetches system info	<ul style="list-style-type: none"> <li>• Just run neofetch</li> </ul>
mplayer	pacman	play music	
gksu	yaourt	Handles GUI and d-bus permissions	<ul style="list-style-type: none"> <li>• <a href="https://bbs.archlinux.org/viewtopic.php?id=240261">https://bbs.archlinux.org/viewtopic.php?id=240261</a></li> </ul>
thunar	pacman	graphical file manager	<ul style="list-style-type: none"> <li>• Just run thunar, may need thumbnail manager or they won't show up.</li> <li>• <a href="https://bbs.archlinux.org/viewtopic.php?id=240261">https://bbs.archlinux.org/viewtopic.php?id=240261</a></li> </ul>
tumbler	yaourt	Thumbnail manager for graphical file managers	
lxappearance	pacman	selected theme from list of installed ones	<ul style="list-style-type: none"> <li>• Just run the command, needs graphical window manager.</li> </ul>

mpd	pacman	Music Player Daemon – an audio player that has a server-client architecture. That's the backend of ncmpcpp music player.	<ul style="list-style-type: none"> <li>• <a href="https://wiki.archlinux.org/index.php/Music_Player_Daemon">https://wiki.archlinux.org/index.php/Music_Player_Daemon</a></li> </ul>
ncmpcpp	pacman	mpd front-end client that organises music playlists, edits metadata etc.	<ul style="list-style-type: none"> <li>• <a href="https://wiki.archlinux.org/index.php/ncmpcpp">https://wiki.archlinux.org/index.php/ncmpcpp</a></li> <li>• <a href="https://pkgbuild.com/~jelle/ncmpcpp/">https://pkgbuild.com/~jelle/ncmpcpp/</a></li> </ul>
alsa-utils	pacman	Sound card and hardware interface, at kernel level	<ul style="list-style-type: none"> <li>• <a href="https://www.archlinux.org/packages/extra/x86_64/alsa-utils/">https://www.archlinux.org/packages/extra/x86_64/alsa-utils/</a></li> <li>• Includes utilities such as alsamixer which controls sound volume.</li> </ul>
pulseaudio	pacman	Sound server that runs as a middleware between applications and hardware devices, either using ALSA or OSS. Runs as daemon.	<ul style="list-style-type: none"> <li>• <a href="https://wiki.archlinux.org/index.php/PulseAudio">https://wiki.archlinux.org/index.php/PulseAudio</a></li> <li>• <a href="https://askubuntu.com/questions/581128/what-is-the-relation-between-alsa-and-pulseaudio">https://askubuntu.com/questions/581128/what-is-the-relation-between-alsa-and-pulseaudio</a></li> </ul>
htop	yaourt	terminal process manager	<ul style="list-style-type: none"> <li>• Just run the command.</li> </ul>
arc-gtk-theme	pacman	dark window theme, requires gtk	<ul style="list-style-type: none"> <li>• <a href="https://www.archlinux.org/packages/community/any/arc-gtk-theme/">https://www.archlinux.org/packages/community/any/arc-gtk-theme/</a></li> </ul>
adwaita-icon-theme	pacman	icon theme, can be set from lxappearance	

## A.2 Difference between MBR and EFI

(from [Arch wiki](#))

**Master Boot Record** The Master Boot Record (MBR) is the first 512 bytes of a storage device. It contains an operating system bootloader and the storage device's partition table. It plays an important role in the boot process under BIOS systems.

**Note:** The MBR is not located in a partition; it is located at the first sector of the device (physical offset 0), preceding the first partition. The boot sector present on a partitionless device or within an individual partition is called a volume boot record (VBR) instead.

**Master Boot Record (bootstrap code):** The first 440 bytes of MBR are the bootstrap code area. On BIOS systems it usually contains the first stage of the boot loader. The bootstrap code can be backed up, restored from backup or erased using `dd`.