

# How to install Arch Linux on Raspberry Pi

Luckily, there is an Arch Linux version designed to work with ARM processors. Let's see how you can install Arch Linux on Raspberry Pi through the following article!

The Raspberry Pi is a versatile, credit-card-sized calculator that can be used for a variety of electronics projects. The great thing about Raspberry Pi is that you have the option to install a variety of operating systems and are not limited to the Raspberry Pi OS. These operating systems include Arch Linux, which is loved for its simplicity.

Luckily, there is an Arch Linux version designed to work with ARM processors. Let's see how you can install Arch Linux on Raspberry Pi through the following article!

## Request

Before you get started, you need the following:

1. Raspberry Pi
2. 8GB micro SD card (or above)
3. Arch Linux ARM
4. Stable Internet connection
5. The computer system can read the SD card

## Prepare the SD card

First, you will need to make a list of the storage devices attached to your device to determine which device is the SD card. Do this with the following command:

```
sudo fdisk -l
```

```
Disk /dev/sdb: 28.91 GiB, 31016878080 bytes, 60579840 sectors
Disk model: USB DISK 2.0
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: 0xf4f22d33

Device      Boot Start      End  Sectors  Size Id Type
/dev/sdb1           8064 60579839 60571776 28.9G c W95 FAT32 (LBA)

Disk /dev/sdc: 7.41 GiB, 7948206080 bytes, 15523840 sectors
Disk model: STORAGE DEVICE
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: 0x5dea878b

Device      Boot Start      End  Sectors  Size Id Type
/dev/sdc1           63 15518789 15518727  7.4G b W95 FAT32
william@ud114am-14rtualBox: ~
```

Make a list of all storage devices attached to the machine

The SD card that we are using is `/dev/sdc`.

You need to format the SD card. To do this run the following command, note that you will need to replace `/dev/sdc` with the name of your SD card:

```
sudo fdisk /dev/sdc
```

You will need to delete any existing partitions on the drive. To do this, type `o` and press **Enter** in Terminal.

Type `p` into Terminal to check if there are any partitions left.

If there are no more partitions left, go ahead and create a boot partition by typing `n`, then `p`, followed by `1` into Terminal. `p` stands for primary (primary) and `1` stands for the first partition on the drive. You will need to press the **Enter** button after this sequence to continue.

```
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): o
Created a new DOS disklabel with disk identifier 0x64ab2eb1.

Command (m for help): p
Disk /dev/sdc: 7.41 GiB, 7948206080 bytes, 15523840 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: 0x64ab2eb1

Command (m for help): n
Partition type
  p   primary (0 primary, 0 extended, 4 free)
  e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-15523839, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-15523839, default 15523839):
```

Create a boot partition

When prompted for the last sector, type + **100M** and press **Enter**.

Type **t** into the Command Prompt, then **c** to set the first partition as type **W95 FAT32 (LBA)** .

Type **n** , followed by **p** , then enter **2** to create the root partition.

```
Command (m for help): n
Partition type
  p   primary (0 primary, 0 extended, 4 free)
  e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-15523839, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-15523839, default 15523839): +100M

Created a new partition 1 of type 'Linux' and of size 100 MiB.

Command (m for help): t
Selected partition 1
Hex code (type L to list all codes): c
Changed type of partition 'Linux' to 'W95 FAT32 (LBA)'.

Command (m for help): n
Partition type
  p   primary (1 primary, 0 extended, 3 free)
  e   extended (container for logical partitions)
Select (default p): p
Partition number (2-4, default 2): 2
First sector (206848-15523839, default 206848):
```

Type **n**, followed by **p**, then enter **2**

Press **Enter** twice to accept the default settings for the first and last sectors.

Write the partition table and exit **fdisk** by typing **w** .

You need to mount **FAT & ext4** file system . To list your partitions, type the following:

```
sudo fdisk -l
```

The SD card will show up and you will be able to see the partitions. In our case, the partitions are / **dev / sdb1** and / **dev / sdb2**.

## Copy Arch Linux file to SD card

Next, the boot and root partitions need to be mounted. Do this with the following series of commands. Remember to replace the partition name in these commands with your partition name.

```
sudo mkfs.vfat /dev/sdb1 sudo mkdir boot sudo mount /dev/sdb1 boot sudo mkfs.ext4
```

```
Calling ioctl() to re-read partition table.
Syncing disks.

william@william-VirtualBox:~$ sudo mkfs.vfat /dev/sdd1
mkfs.fat 4.1 (2017-01-24)
william@william-VirtualBox:~$ sudo mkfs.ext4 /dev/sdd2
mke2fs 1.45.5 (07-Jan-2020)
Found a dos partition table in /dev/sdd2
Proceed anyway? (y,N) y
Creating filesystem with 1914624 4k blocks and 479552 inodes
Filesystem UUID: b9e3d6ea-d408-4411-8d3a-4d76fe15e15c
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

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

william@william-VirtualBox:~$
william@william-VirtualBox:~$ mkdir boot
mkdir: cannot create directory 'boot': File exists
william@william-VirtualBox:~$ sudo mkdir boot
mkdir: cannot create directory 'boot': File exists
william@william-VirtualBox:~$ sudo mount /dev/sdd1 boot
william@william-VirtualBox:~$ sudo mount /dev/sdd2 root
william@william-VirtualBox:~$
```

Copy Arch Linux file to SD card

Now, put the Arch Linux file you downloaded into the home directory and extract it to the root directory of the SD card with the following command:

```
sudo bsdtar -xpf ArchLinuxARM-rpi-2-latest.tar.gz -C root sync
```

The boot files will need to be moved to the boot partition of the SD card with:

```
sudo mv root/boot/* boot
```

You can unmount two partitions using:

```
umount boot root
```

Insert the SD card into the Raspberry Pi.

## Initial setup on Raspberry Pi

After inserting the SD card into the Raspberry Pi, go ahead and enable it. You will need an Internet connection via an Ethernet cable or WiFi network. To connect over WiFi, first log in with the default root account. The username for this account is **root** and the password is also **root**. Now run the following command:

```
wifi-menu
```

A menu will load and you can choose your WiFi network, then log in. Now complete the installation by launching the keyring **pacman** and entering the Arch Linux ARM package signing keys with:

```
pacman-key --init pacman-key --populate archlinuxarm
```

You can go ahead and update system packages with:

```
pacman -Syu
```

You should change the default username. Do this with the following command:

```
usermod -l newusername oldusername
```

Alternatively, change the password with:

```
passwd newusername
```

You will be asked to enter a new password and then confirm it. To change the name of the home directory and reflect the new username, run the following command:

```
usermod -d /home/newusername -m newusername
```

You should also change the root account's password. Do this with:

```
passwd
```

To grant sudo privileges to your user account, you will have to take the following steps to install the sudo package:

```
pacman -S sudo
```

You will have to edit the configuration file for sudo. Do this with:

```
EDITOR=nano visudo
```

Add **newusername ALL = (ALL) ALL** under the line **root ALL = (ALL) ALL** .

Close and save the file. You have completed all the work!

Now that you've installed Arch Linux on your Raspberry Pi, there's a ton of things you can do, including installing and playing Minecraft, turning it into a NAS or Plex server. etc .

Good luck.

You finished reading the article "**How to install Arch Linux on Raspberry Pi**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.