DOCUMENTATION > INSTALLATION > INSTALLING-IMAGES > LINUX

INSTALLING OPERATING SYSTEM IMAGES ON LINUX

Etcher is typically the easiest option for most users to write images to SD cards, so it is a good place to start. If you're looking for more advanced options on Linux, you can use the standard command line tools below.

Note: use of the `dd` tool can overwrite any partition of your machine. If you specify the wrong device in the instructions below, you could delete your primary Linux partition. Please be careful.

DISCOVERING THE SD CARD MOUNTPOINT AND UNMOUNTING IT

- Run `df -h` to see which devices are currently mounted.

- If your computer has a slot for SD cards, insert the card. If not, insert the card into an SD card reader, then connect the reader to your computer.

BLOG DOWNLOADS COMMUNITY HELP FORUMS EDUCATION

device appears, then your system is not automounting devices. In this case, you will need to search for the device name using another method. The `dmesg | tail` command will display the most recent system messages, which should contain information on the naming of the SD card device. The naming of the device will follow the format described in the next paragraph. Note that if the SD card was not automounted, you do not need to unmount later.

- The left column of the results from `df -h` command gives the device name of your SD card. It will be listed as something like `/dev/mmcblk0p1` or `/dev/sdX1`, where X is a lower case letter indicating the device. The last part (`p1` or `1` respectively) is the partition number. You want to write to the whole SD card, not just one partition. You therefore need to remove that section from the name. You should see something like `/dev/mmcblk0` or `/dev/sdX` as the device name for the whole SD card. Note that the SD card can show up more than once in the output of `df`. It will do this if you have previously written a Raspberry Pi image to this SD card, because the Raspberry Pi SD images have more than one partition.

- Now you have noted the device name, you need to unmount it so that files can't be read or written to the SD card while you are copying over the SD image.

- Run `umount /dev/sdX1`, replacing `sdX1` with whatever your SD card's device name is, including the partition number.

- If your SD card shows up more than once in the output of `df`, this shows that the card has multiple partitions. You should unmount all of these partitions.

COPYING THE IMAGE TO THE SD CARD

- In a terminal window, write the image to the card with the command below, making sure you replace the input file `if=` argument with the path to your `.img` file, and the `/dev/sdX` in the output file `of=` argument with the correct device name. This is very important, as you will lose all the data on the hard drive if you provide the wrong device name. Make sure the device name is the name of the whole SD card as described above, not just a partition. For example: `sdd`, not `sdds1` or `sddp1`, and `mmcblk0`, not `mmcblk0p1`.

  ```
  dd bs=4M if=2017-09-07-raspbian-stretch.img of=/dev/sdX
  conv=fsync
  ```

- Please note that block size set to `4M` will work most of the time. If not, try `1M`, although this will take considerably longer.

- Also note that if you are not logged in as root you will need to prefix this with `sudo`.

COPYING A ZIPPED IMAGE TO THE SD CARD

In Linux it is possible to combine the unzip and SD copying process into one command, which avoids any issues that might occur when the unzipped image is larger than 4GB. This can happen on certain filesystems that do not support files larger than 4GB (e.g. FAT), although it should be noted that most Linux installations do not use FAT and therefore do not have this limitation.

The following command unzips the zip file (replace 2017-09-07-raspbian-stretch.zip with the appropriate zip filename), and pipes the output directly to the dd command. This in turn copies it to the SD card, as described in the previous section.

```
unzip -p 2017-09-07-raspbian-stretch.zip | sudo dd of=/dev/sdX
bs=4M conv=fsync
```

CHECKING THE IMAGE COPY PROGRESS

- By default, the `dd` command does not give any information about its progress, so it may appear to have frozen. It can take more than five minutes to finish writing to the card. If your card reader has an LED, it may blink during the write process.

- To see the progress of the copy operation, you can run the dd command with the status option.

```
dd bs=4M if=2017-09-07-raspbian-stretch.img of=/dev/sdX
status=progress conv=fsync
```

- If you are using an older version of `dd`, the status option may not be available. You may be able to use the `dcfldd` command instead, which will give a progress report showing how much has been written. Another method is to send a USR1 signal to `dd`, which will let it print status information. Find out the PID of `dd` by using `pgrep -l dd` or `ps a | grep dd`. Then use `kill -USR1 PID` to send the USR1 signal to `dd`.

CHECKING WHETHER THE IMAGE WAS CORRECTLY WRITTEN TO THE SD CARD

- After `dd` has finished copying, you can check what has been written to the SD card by `dd`-ing from the card back to another image on your hard disk; truncating the new image to the same size as the original; and then running `diff` (or `md5sum`) on those two images.

- If the SD card is bigger than the original image size, `dd` will make a copy of the whole card. We must therefore truncate the new image to the size of the original image. Make sure you replace the input file `if=` argument with the correct device name. `diff` should report that the files are identical.

```
dd bs=4M if=/dev/sdX of=from-sd-card.img
truncate --reference 2017-09-07-raspbian-stretch.img from-sd-
card.img
diff -s from-sd-card.img 2017-09-07-raspbian-stretch.img
```

- Run `sync`. This will ensure the write cache is flushed and that it is safe to unmount your SD card.

- Remove the SD card from the card reader.

VIEW/EDIT THIS PAGE ON GITHUB

READ OUR USAGE AND CONTRIBUTIONS POLICY

RASPBERRY PI FOUNDATION
UK REGISTERED CHARITY 1129409

Cookies Trademark rules and brand guidelines