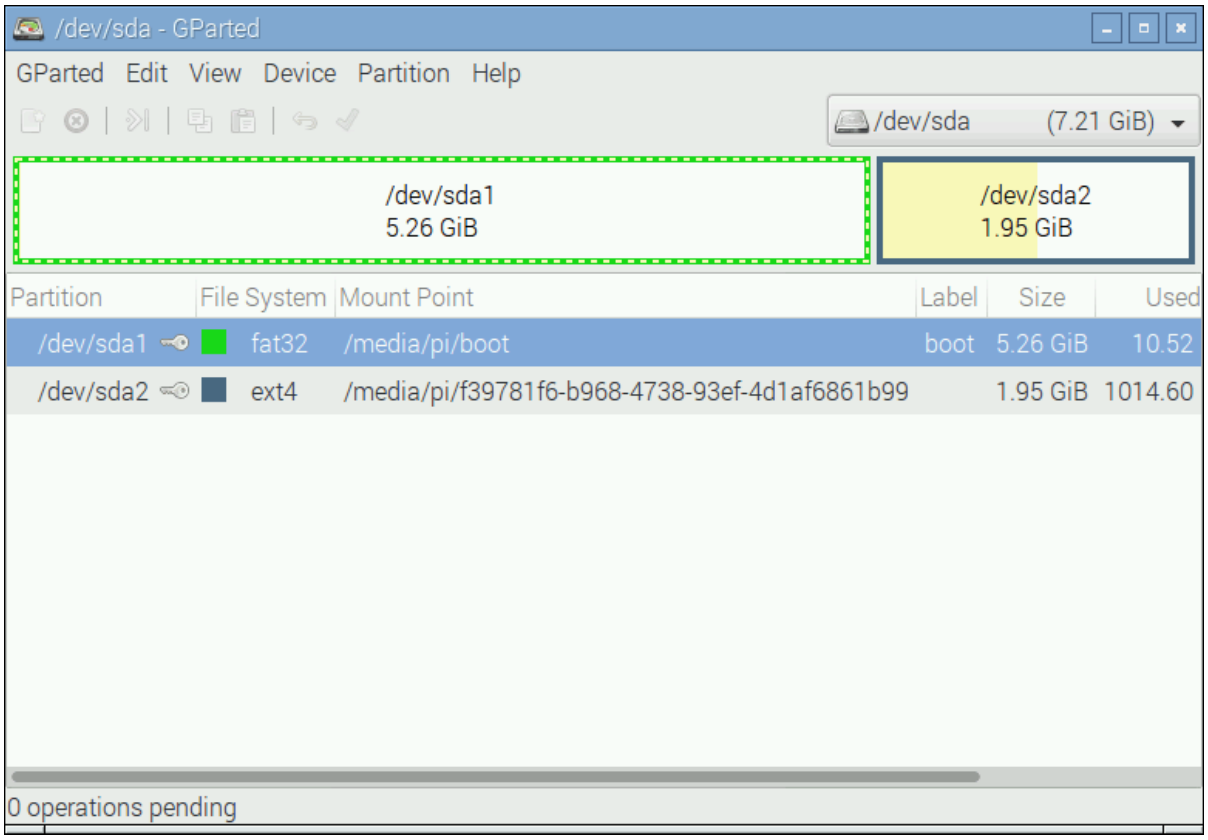
Resizing the Raspberry Pi Boot Partition

Created by Phillip Burgess



Partition	File System	Mount Point	Label	Size	Used
/dev/sda1	🟩 fat32	/media/pi/boot	boot	5.26 GiB	10.52
/dev/sda2	■ ext4	/media/pi/f39781f6-b968-4738-93ef-4d1af6861b99		1.95 GiB	1014.60

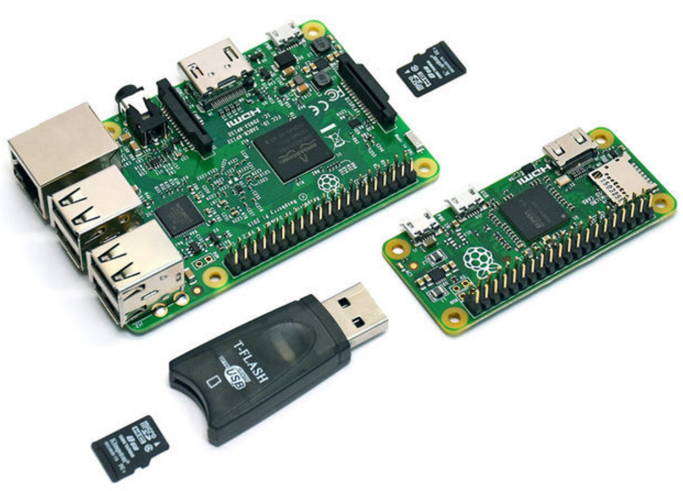
0 operations pending

https://learn.adafruit.com/resizing-raspberry-pi-boot-partition

Last updated on 2024-06-03 01:56:00 PM EDT

Table of Contents

Overview



Though the Raspberry Pi computer is eminently networkable, some projects still just work best by physically moving the SD card to a desktop system to exchange data… but normally only a small section of the card is accessible to Windows and Mac computers. This guide explains one way of making more space available to both the Pi and other systems.

Some examples of projects that work this way include cameras, data loggers and handheld game consoles…anywhere a network connection can't be assumed. This is the decades-old tradition of "Sneakernet" — once done with floppy disks or tape, but still occasionally relevant in our age of flash memory.

Utilities exist to allow Windows and Mac systems to access the main SD partition, but I prefer the following method as it allows a stock PC to read and write the card, vital when working in the field or on a borrowed computer.

Items Needed

In most cases, **two Raspberry Pi computers** are required…

One is the "**target**" system — usually embedded into a project, this is the camera or data logger or game console, whatever you're building. Sometimes it might be a "headless" system with no display.

The other is the "**setup**" system — this runs a full version of the Raspbian desktop operating system, has **networking** capability, with **keyboard**, **mouse** and **monitor** attached. A Raspberry **Pi 3**, **Pi 2**, or **Model B+** are ideal for this, since they have ample USB ports (and built-in WiFi on the Pi 3).

There are some exceptions where this can be done with a single system, or an alternate setup system…more on that in a moment.

Also required:

- USB flash card reader.
- **Two** micro SD cards: the **setup** system requires an **8 GB or larger** card, the **target** can be most any size, though **4 to 32 GB** is ideal…**whatever your target operating system requires**, plus space for the data to be moved between systems.
- Depending on your network and setup system features, an Ethernet connection or a USB WiFi adapter may be required (WiFi is built-in on Pi 3).

Some prior Raspberry Pi experience is assumed; writing an SD card, connecting to networks, etc. If you're new to Raspberry Pi and any of this is unfamiliar, begin with our starter tutorials in the Learn Raspberry Pi (https://adafru.it/dpe) series.

What if I've Only Got One Raspberry Pi?

If your target system is a "full featured" Pi — something like a Pi 3, Pi 2 or Model B+, easily networked and with many USB ports — it's no problem, the target system can also act as the setup system, just switch SD cards when needed.

If your target is a "feature-limited" Pi — like a Model A+ or Pi Zero — you can still do your setup on this system, but unless you already own a collection of the required dongles (powered USB hub, WiFi or Ethernet adapter, mini-to-full-size HDMI adapter for Pi Zero, etc.), it's easier and often cheaper just to order the latest Pi board to do the setup than to deal with all these parts. And when you're done, now you have an extra Pi to play around with.

But there's one more option…if you have a desktop or laptop computer running any of the popular Linux desktop distributions (e.g. Ubuntu, Debian, etc.)…often you can use this as the "setup" system, the necessary software (gparted) is usually already installed. You'll only need one Raspberry Pi (the target system) and one SD card in this situation.

Setup

Download an **operating system SD card image** for each system (**setup** and **target**). For the "setup" system, this will be Raspbian Jessie **(https://adafru.it/fQi)** (the full version, not “Lite”)…2016-05-10 or later, whatever the current release is. For the “target” system…this depends on the project…sometimes it’ll use the same Raspbian Jessie image, others will use Raspbian Jessie Lite (which lacks the GUI desktop and other features), or an entirely different SD card image from another project.

“**Burn**” each OS to an SD card using a tool such as Etcher (https://adafru.it/nNb) or your utility of choice. For the “setup” system running full Raspbian, this requires an **8 GB or larger** card. For the “target” system, this depends…whatever the operating system requires, plus some extra working space for exchanging data…this could be as small as **2 GB** for something like Raspbian Jessie Lite, or up to **32 GB** maximum.

Keep track of which card is which…setup or target.

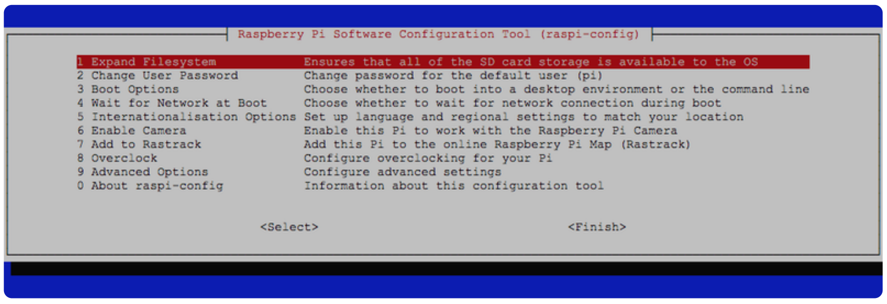
Boot and Configure Target System

Let’s set up the **target** system first, though I realize this is a bit counter-intuitive.

If your target system OS is based on a recent version of **Raspbian Jessie** (or Lite), one nice feature is that it will boot on **any Raspberry Pi board**. So if your target hardware is a feature-limited Model A+ or a Pi Zero, you can boot on a more capable system (Pi 3, etc.) which is faster and more easily networked, then move the card later.

Go through the usual **first-boot configuration** process. For Raspbian, that means logging in with the default “**pi**” / “**raspberry**” username and password and running the **raspi-config** utility…

```
sudo raspi-config
```



```
Raspberry Pi Software Configuration Tool (raspi-config)

1 Expand Filesystem            Ensures that all of the SD card storage is available to the OS
2 Change User Password         Change password for the default user (pi)
3 Boot Options                 Choose whether to boot into a desktop environment or the command line
4 Wait for Network at Boot     Choose whether to wait for network connection during boot
5 Internationalisation Options Set up language and regional settings to match your location
6 Enable Camera                Enable this Pi to work with the Raspberry Pi Camera
7 Add to Rastrack              Add this Pi to the online Raspberry Pi Map (Rastrack)
8 Overclock                    Configure overclocking for your Pi
9 Advanced Options             Configure advanced settings
0 About raspi-config           Information about this configuration tool

                 <Select>                                 <Finish>
```

At the very least, you should expand the filesystem and set up the Internationalisation Options (especially the keyboard…if you’re getting unexpected characters when typing, this is why). Depending on the project this is going into, you may want to enable the camera, disable overscan (under Advanced Options), etc. Totally depends on the project…choose your own adventure.

With the basic setup done (it may prompt you to reboot...this is fine), you may want to get the Pi connected to your network to download additional software, apply updates, etc. That's beyond the scope of this guide, but you'll find pointers in the Learn Raspberry Pi (https://adafru.it/jcW) series.

Basically...your goal is to get the target system card **as close to finished as possible**, all software installed and configured.

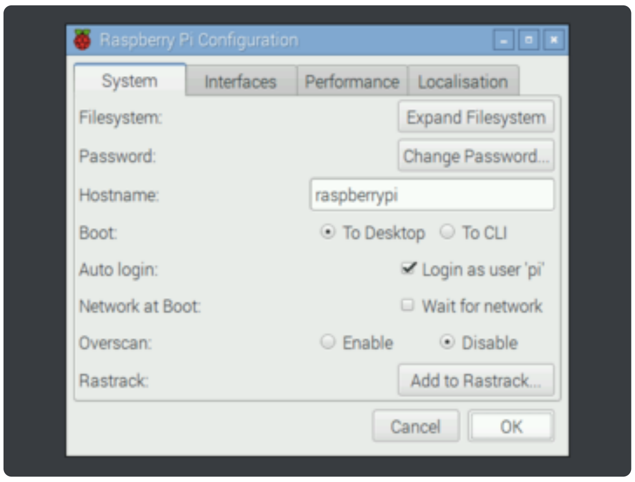
Once that's ready, shut the system down gracefully...if working from the Raspbian command line, that's...

```
sudo shutdown -h now
```

After the system has halted (it takes about 20 seconds or so), disconnect power, remove the SD card and **put it into the USB card reader**.

Boot and Configure the Setup System

Now we'll repeat that process for the **setup** system...except this one will definitely be running the **full Raspbian Jessie** OS, so you'll have a mouse and GUI desktop for all of this.



Filesystem:
Password:
Hostname: raspberrypi
Boot: [✓ To Desktop] To CLI
Auto login: [✓ Login as user 'pi']
Network at Boot: Wait for network
Overscan: Enable [✓ Disable]
Rastrack:

The system configuration utility can be accessed at **Pi Menu→Preferences→Raspberry Pi Configuration**.

Once the basic system is configured to your liking, set up networking. Sometimes this is as easy as plugging in an Ethernet cable or using the WiFi icon near the top-right of the screen.

This Pi system must have a **working internet connection** to complete this guide.

Back Up Target SD Card

Put the target system’s SD card in a USB reader and plug it into your normal desktop computer, or wherever you originally “burned” the SD card images.

Most utilities for burning SD cards also have a feature for working the other direction…they can back up an SD card to a file. (Etcher doesn’t have this feature yet, but other utilities do.)

Make a backup image of your target SD card before proceeding. We’ll be doing some serious shenanigans on the next page. There’s the possibility of messing things up, so this “save point” is insurance against having to start completely from scratch.

Once you have a backup, eject the USB reader and plug it into the setup system.

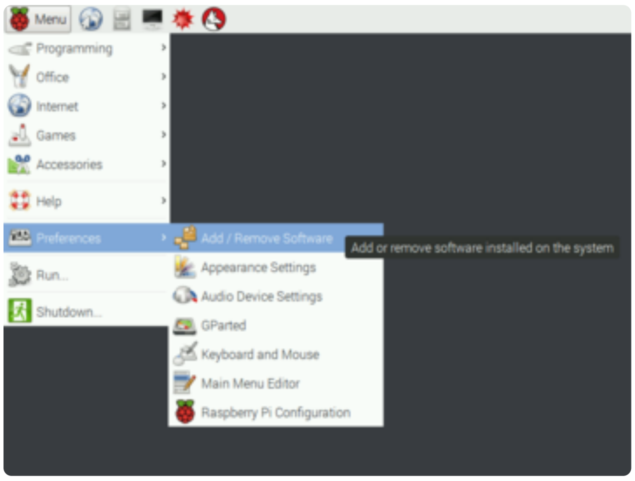
Edit Partitions

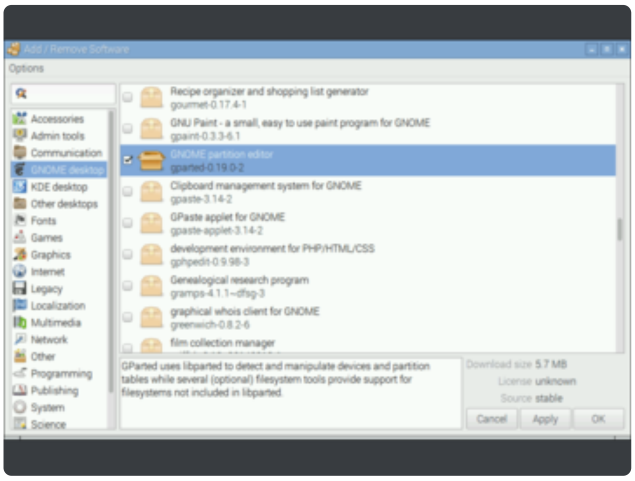
At this point, your “setup” system should be up and running the Raspbian graphical desktop, with networking, keyboard and mouse connected, with the “target” system's SD card in a USB reader connected to one of the setup system’s USB ports.

We’ll use the **gparted** utility to edit the partition table. This is not installed with Raspbian by default, hence the need for a network connection.

It’s usually fastest and easiest to open a Terminal window and type:

```
sudo apt-get install -y gparted
sudo gparted
```

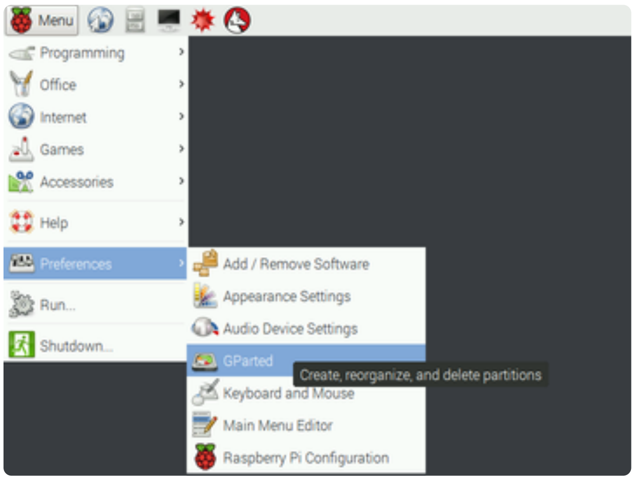




Recipe organizer and shopping list generator
gourmet-0.17.4-1
GNU Paint - a small, easy to use paint program for GNOME
gpaint-0.3.3-6.1
GNOME partition editor
gparted-0.19.0-2
Clipboard management system for GNOME
gpaste-3.14-2
GPaste applet for GNOME
gpaste-applet-3.14-2
development environment for PHP/HTML/CSS
gphpedit-0.9.98-3
Genealogical research program
gramps-4.1.1~dfsg-3
graphical whois client for GNOME
greenwich-0.8.2-6
film collection manager

GParted uses libparted to detect and manipulate devices and partition tables while several (optional) filesystem tools provide support for filesystems not included in libparted.

Download size 5.7 MB
License unknown
Source stable

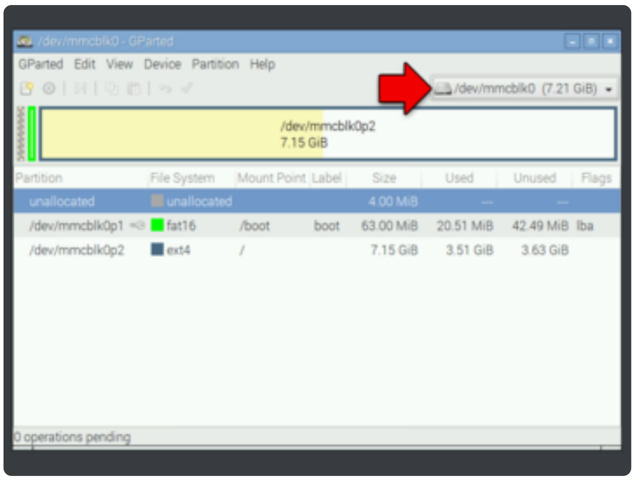


If you insist on using a GUI to install the software, click **Pi Menu→Preferences→Add / Remove Software**. You'll find gparted in the "Gnome Desktop" section, or use the search field to locate it. Tick the check box and click "Apply" to install.

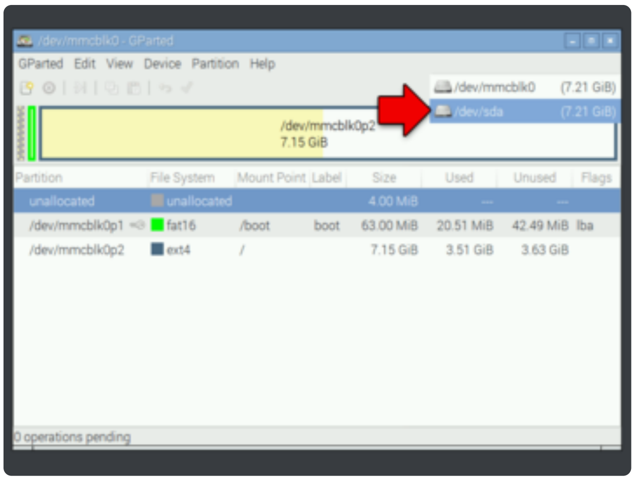
You can then run gparted via **Pi Menu→Preferences→GParted**. You'll be asked for the root password, since this is a system-level operation.

But really, the Terminal way is much quicker.

Select Device



Partition	File System	Mount Point	Label	Size	Used	Unused	Flags
unallocated	unallocated			4.00 MiB	---	---	
/dev/mmcblk0p1	fat16	/boot	boot	63.00 MiB	20.51 MiB	42.49 MiB	lba
/dev/mmcblk0p2	ext4	/		7.15 GiB	3.51 GiB	3.63 GiB	



Partition	File System	Mount Point	Label	Size	Used	Unused	Flags
unallocated	unallocated			4.00 MiB	---	---	
/dev/mmcblk0p1	fat16	/boot	boot	63.00 MiB	20.51 MiB	42.49 MiB	lba
/dev/mmcblk0p2	ext4	/		7.15 GiB	3.51 GiB	3.63 GiB	

When you first run gparted, it will select the **internal** SD card by default.

We want to edit the card in the **USB reader**. To do this, use the pop-up menu near the top-right of the window, or the **GParted→Devices** menu.

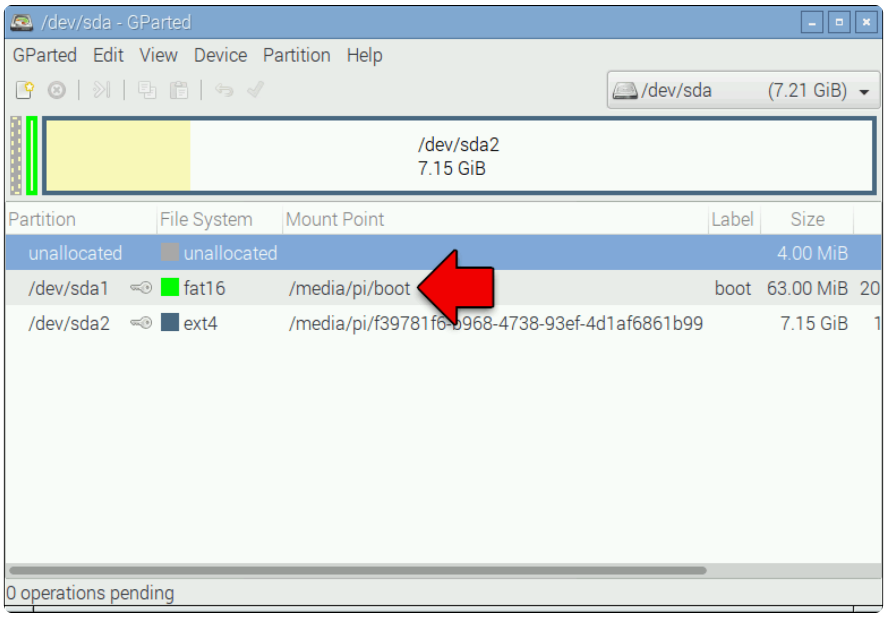
It should appear as **/dev/sda** or similar…the name's not important, just make sure you're **not** selecting /dev/mmcblk0, which is the internal card.

Save /boot Partition Contents

You'll see a graph and list of at least two partions on the card — the tiny boot partition, and a much larger partition taking up the bulk of the card — and perhaps one or more small "unallocated" spaces.

The unallocated spaces are used to align partitions to specific start and end points on the card. You can ignore these.

The first "real" partition is a FAT16 filesystem…that's the boot partition on the target system SD card. Note the mount point of this partition…it's probably **/media/pi/boot** or similar. We'll need this exact path in a moment.



Partition	File System	Mount Point	Label	Size
unallocated	unallocated			4.00 MiB
/dev/sda1	fat16	/media/pi/boot	boot	63.00 MiB
/dev/sda2	ext4	/media/pi/f39781f6-b968-4738-93ef-4d1af6861b99		7.15 GiB

Open a Terminal window if you don't already have one running, and type the following commands...

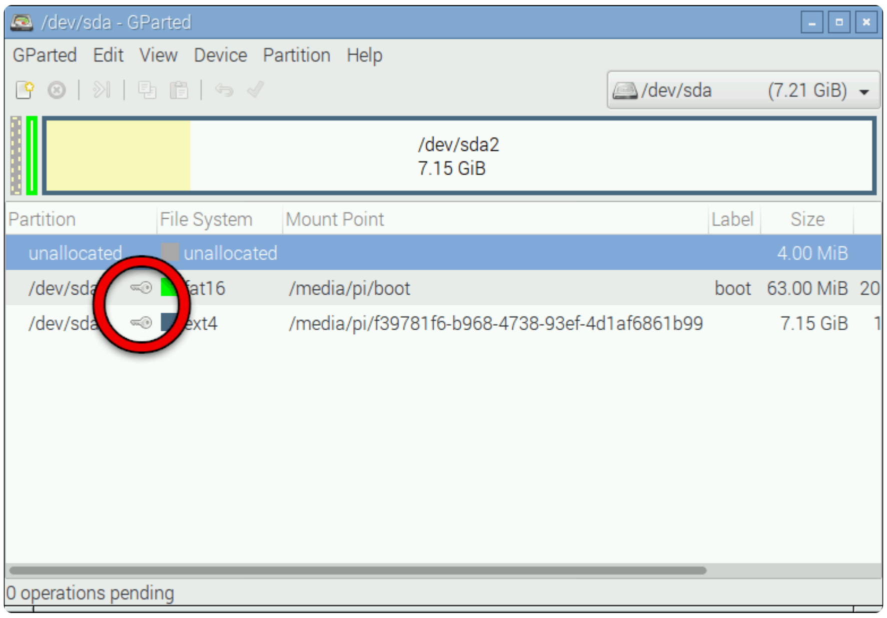
```
cd
sudo cp -r /media/pi/boot .
```

You may need to change "/media/pi/boot" if a different mount point was shown in gparted. Also, there should be a space between "boot" and the final period.

This copies the full contents of the /boot partition (from the card in the USB reader, not the setup system's card) to a directory called "boot" in your home folder. This is a temporary measure...we'll move these files back later.

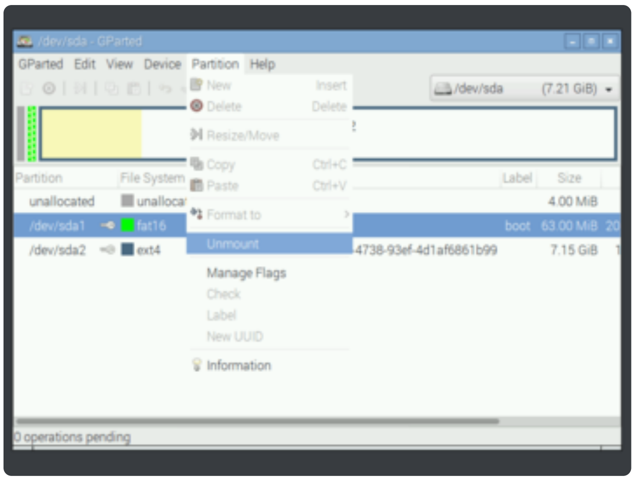
Unmount Partitions Before Editing

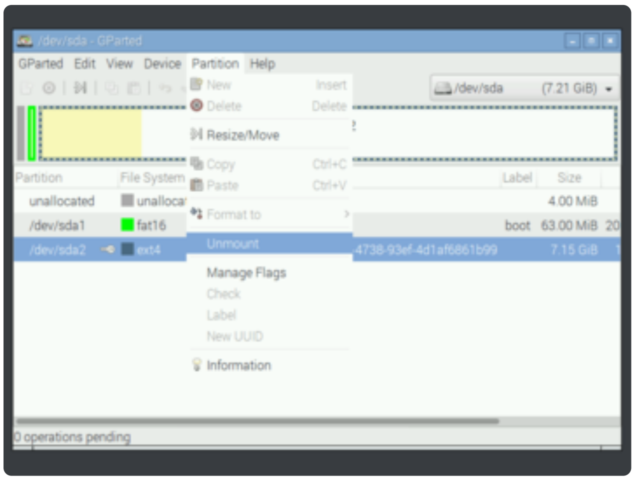
You can not edit partitions while mounted (e.g. accessing files). If there's a **key icon** next to one or both partitions, it's necessary to unmount these filesystems first...



Partition	File System	Mount Point	Label	Size	
unallocated	unallocated			4.00 MiB	
/dev/sda	fat16	/media/pi/boot	boot	63.00 MiB	20
/dev/sda	ext4	/media/pi/f39781f6-b968-4738-93ef-4d1af6861b99		7.15 GiB	1

0 operations pending



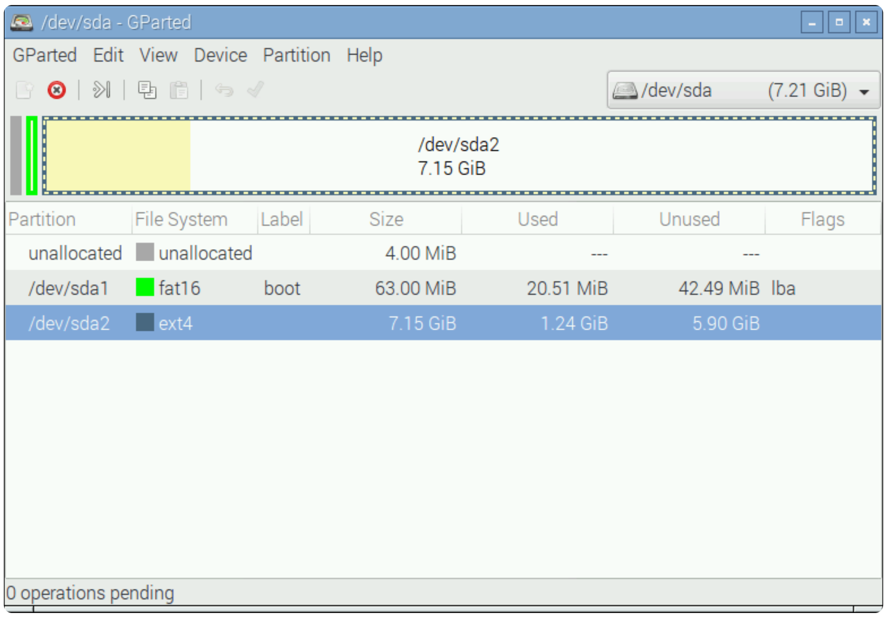


Highlight the /boot partition, then select **Partition→Unmount**.

As a shortcut, you can also right-click on a partition in the graph or list and select Unmount there.

Repeat for the second (root) partition. There should be **no key icons** in the list… everything’s editable now.

When everything’s ready, it should look something like this…



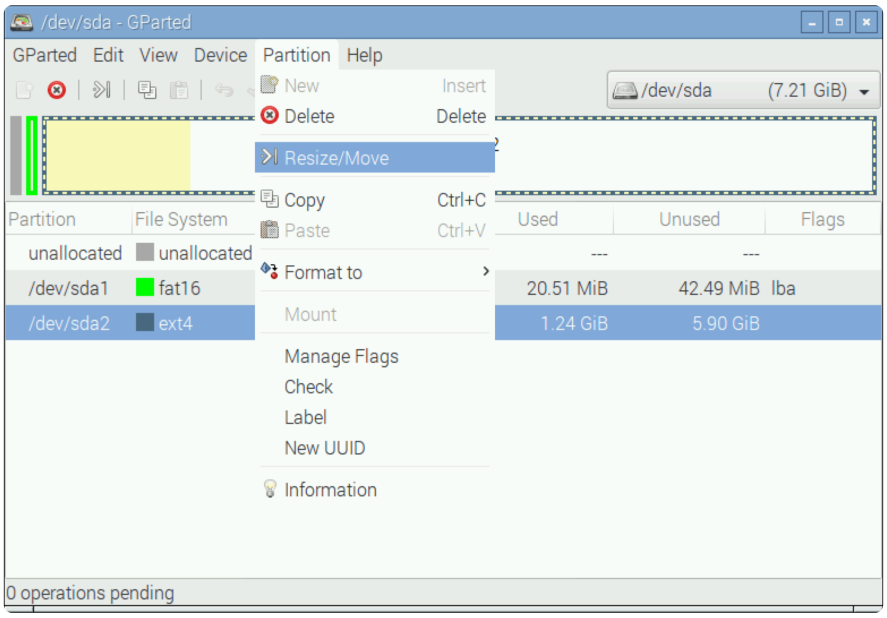
Partition	File System	Label	Size	Used	Unused	Flags
unallocated	unallocated		4.00 MiB	---	---	
/dev/sda1	fat16	boot	63.00 MiB	20.51 MiB	42.49 MiB	lba
/dev/sda2	ext4		7.15 GiB	1.24 GiB	5.90 GiB	

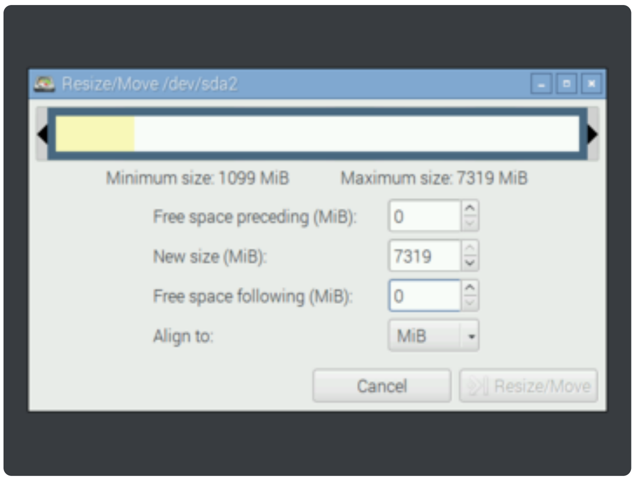
0 operations pending

Resize the Root Partition

To increase the /boot partition, we first need to shrink the root partition to make space.

Highlight the **root** partition (**/dev/sda2**) and select **Partition→Resize/Move** (or use the right-click shortcut).


/dev/sda - GParted
GParted Edit View Device Partition Help
/dev/sda (7.21 GiB)
New Insert
Delete Delete
Resize/Move
Copy Ctrl+C
Paste Ctrl+V
Format to
Mount
Manage Flags
Check
Label
New UUID
Information
Partition File System Used Unused Flags
unallocated unallocated --- ---
/dev/sda1 fat16 20.51 MiB 42.49 MiB lba
/dev/sda2 ext4 1.24 GiB 5.90 GiB
0 operations pending

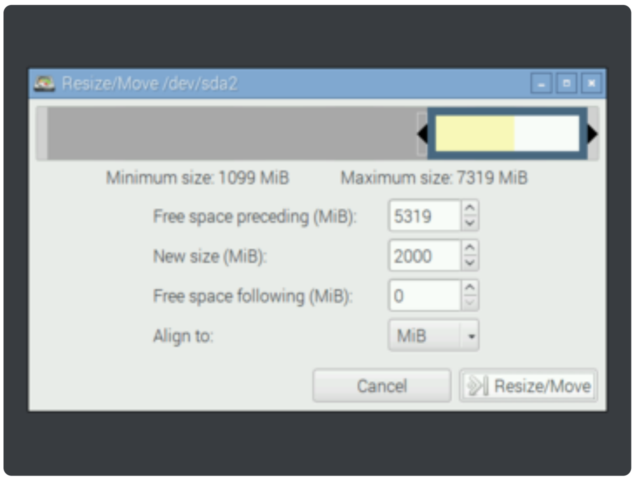

Resize/Move /dev/sda2

Minimum size: 1099 MiB   Maximum size: 7319 MiB

Free space preceding (MiB): 0
New size (MiB): 7319
Free space following (MiB): 0
Align to: MiB

The Resize/Move dialog visually shows the full partition size (white) and the amount of data actually present within (yellow).

Because this is a Linux system, there needs to be some space left for caches and log files…or for future updates and additions…we can't shrink it down to an exact fit.

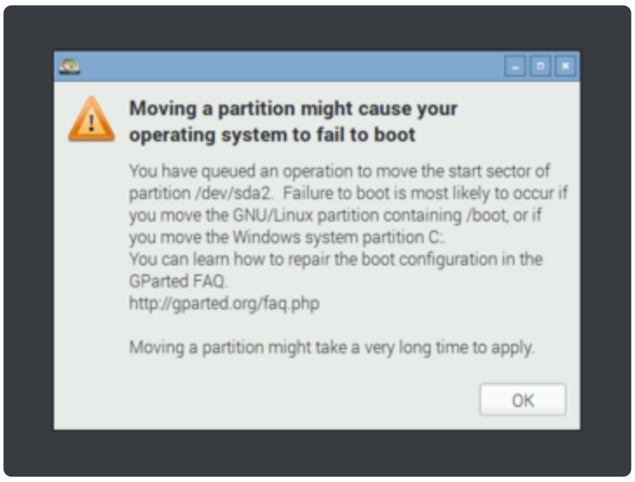

Resize/Move /dev/sda2

Minimum size: 1099 MiB   Maximum size: 7319 MiB

Free space preceding (MiB): 5319
New size (MiB): 2000
Free space following (MiB): 0
Align to: MiB

There's no hard-and-fast rule for exactly how much should be kept free. One third to one half felt "safe"…with Raspbian Jessie Lite, I rounded to an even 2000 MB.

Any surplus should go to the Free space preceding field…Free space following should be 0.


Moving a partition might cause your operating system to fail to boot

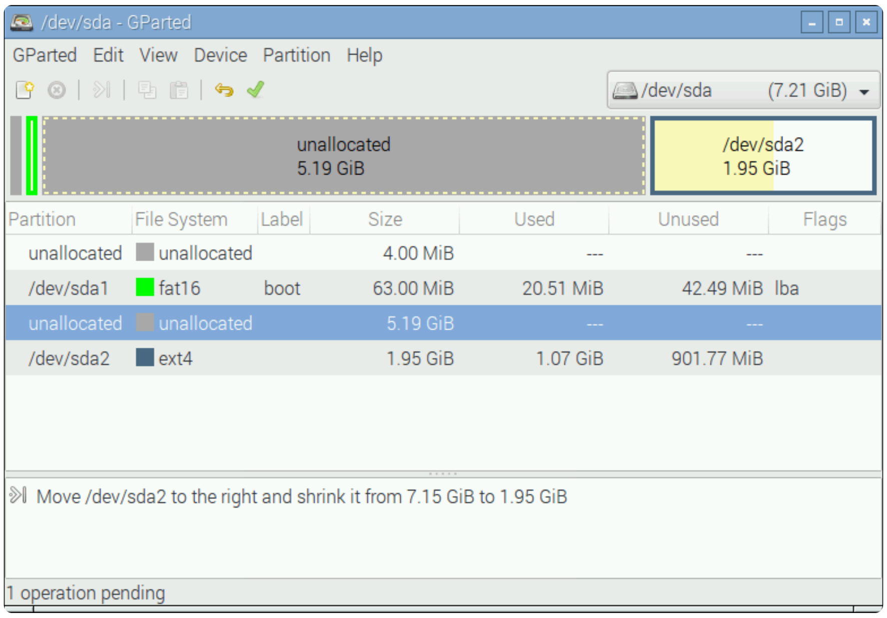
You have queued an operation to move the start sector of partition /dev/sda2. Failure to boot is most likely to occur if you move the GNU/Linux partition containing /boot, or if you move the Windows system partition C:.
You can learn how to repair the boot configuration in the GParted FAQ.
http://gparted.org/faq.php

Moving a partition might take a very long time to apply.

You can do this either with the numeric fields or by dragging and moving the partition graph with the mouse. When it's adjusted to your liking, click the "Resize/ Move" button.

A pop-up dialog will warn about moving partitions. It's all good, just click "OK."

Afterward, it should look something like this (though your partition sizes may vary with card capacity and installed operating system version):



Partition	File System	Label	Size	Used	Unused	Flags
unallocated	unallocated		4.00 MiB	---	---	
/dev/sda1	fat16	boot	63.00 MiB	20.51 MiB	42.49 MiB	lba
unallocated	unallocated		5.19 GiB	---	---	
/dev/sda2	ext4		1.95 GiB	1.07 GiB	901.77 MiB	

Move /dev/sda2 to the right and shrink it from 7.15 GiB to 1.95 GiB

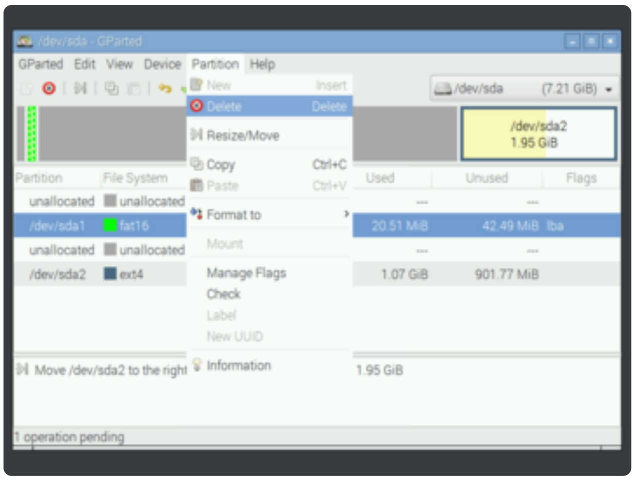
1 operation pending

The partition map isn't actually changed on the card yet. Several operations can be queued up and applied all at once…

Create a Larger Boot Partition

This next sequence is the secret sauce that makes this guide work…

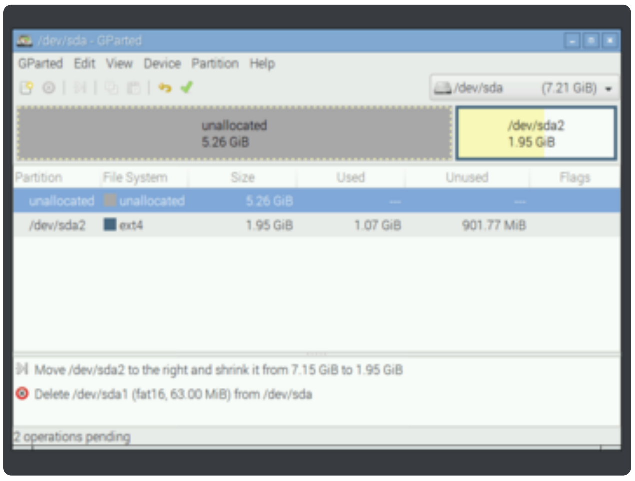
Using gparted's Resize/Move feature on the boot partition renders the card unreadable on Windows and Mac systems (though it still works with Linux). So the trick here is to **delete the existing boot partition**, **create a new one** with the desired size, then **restore the files** there that we backed up earlier.



Partition	File System		Used	Unused	Flags
unallocated	unallocated		---	---	
/dev/sda1	fat16		20.51 MiB	42.49 MiB	lba
unallocated	unallocated		---	---	
/dev/sda2	ext4		1.07 GiB	901.77 MiB	

```
Move /dev/sda2 to the right            1.95 GiB
```

1 operation pending



Partition	File System	Size	Used	Unused	Flags
unallocated	unallocated	5.26 GiB	---	---	
/dev/sda2	ext4	1.95 GiB	1.07 GiB	901.77 MiB	

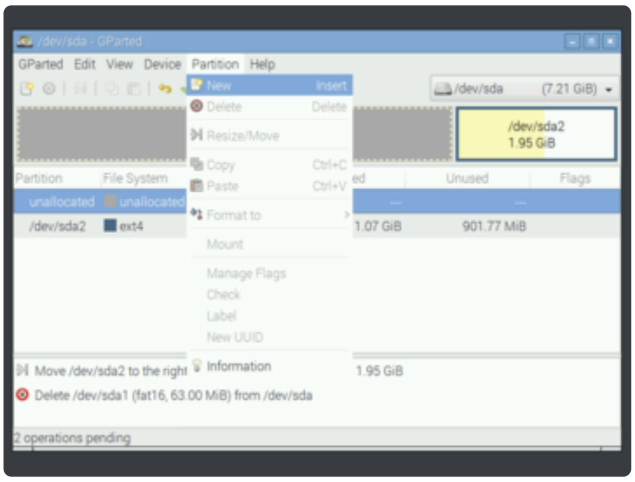
```
Move /dev/sda2 to the right and shrink it from 7.15 GiB to 1.95 GiB
Delete /dev/sda1 (fat16, 63.00 MiB) from /dev/sda
```

2 operations pending

Highlight the **boot** partition (**/dev/sda1**) and select **Partition→Delete** (or use the right-click shortcut).

You'll then have the root partition on the right, and a lot of unallocated space on the left.

Then select **Partition→New** to begin creating a replacement…

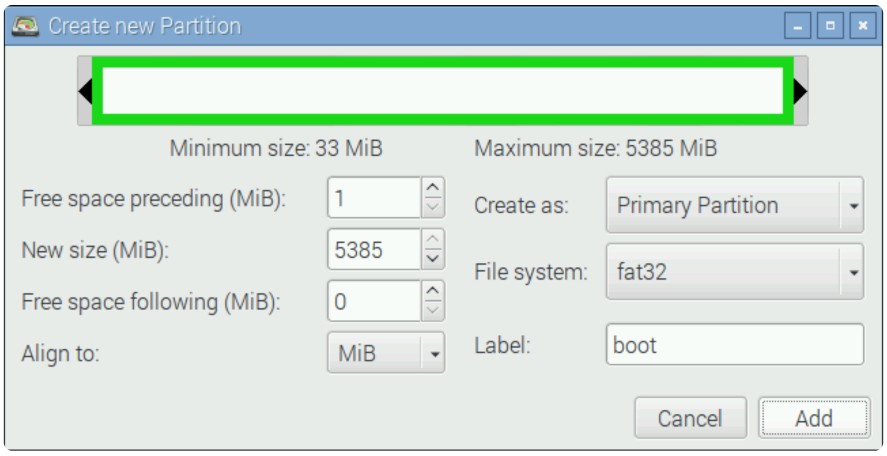


Partition	File System		Used	Unused	Flags
unallocated	unallocated		---	---	
/dev/sda2	ext4		1.07 GiB	901.77 MiB	

```
Move /dev/sda2 to the right            1.95 GiB
Delete /dev/sda1 (fat16, 63.00 MiB) from /dev/sda
```

2 operations pending

The Create new Partition dialog is similar to the Resize/Move dialog, but with some additional options…


Create new Partition

Minimum size: 33 MiB   Maximum size: 5385 MiB

Free space preceding (MiB):	1	Create as:	Primary Partition
New size (MiB):	5385	File system:	fat32
Free space following (MiB):	0	Label:	boot
Align to:	MiB		

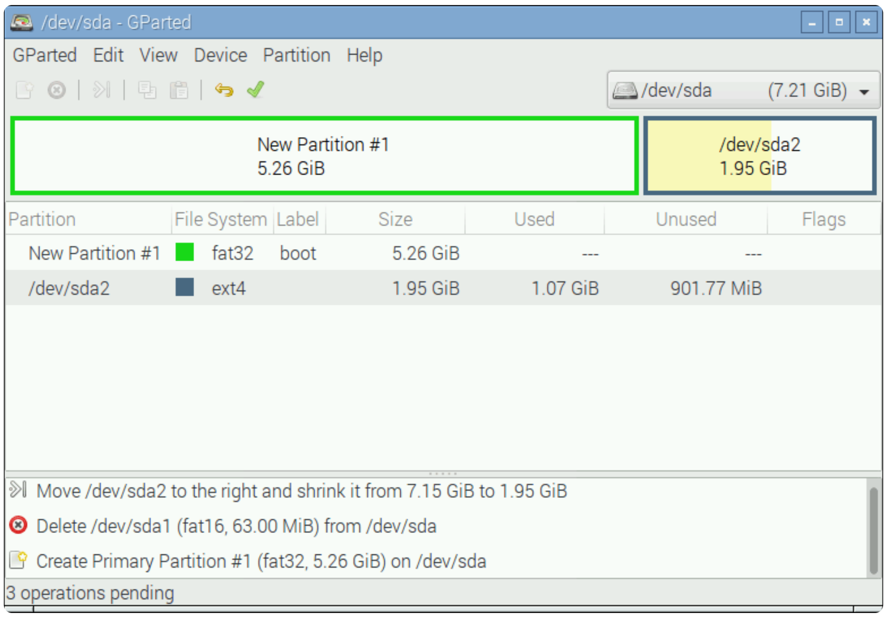
Adjust the size (using the mouse or text fields) to fill the available space, but note that there is an **upper limit** to the /boot partition size…it **cannot exceed 32GB**, even on a very large SD card. If there's room left over, you can edit the root partition again, enlarging it to claim this space, or create extra partition(s) there (these will appear as separate drives, not a contiguous space).

For the other options, select/enter the following exactly:

- Create as: **Primary Partition**
- File system: **fat32**
- Label: **boot**

Then click the "Add" button.

When you're done, the partition map should look something like the following. It may or may not include small "unallocated" blocks — either way is completely normal, it varies with partition sizes and SD card capacity.

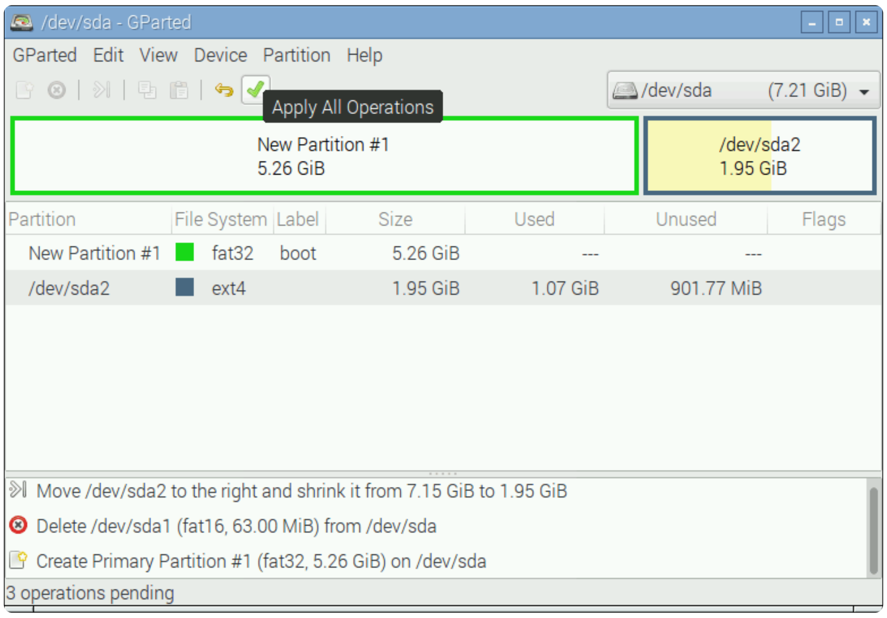


Partition	File System	Label	Size	Used	Unused	Flags
New Partition #1	🟩 fat32	boot	5.26 GiB	---	---	
/dev/sda2	■ ext4		1.95 GiB	1.07 GiB	901.77 MiB	

```
Move /dev/sda2 to the right and shrink it from 7.15 GiB to 1.95 GiB
Delete /dev/sda1 (fat16, 63.00 MiB) from /dev/sda
Create Primary Partition #1 (fat32, 5.26 GiB) on /dev/sda
3 operations pending
```

Apply Changes

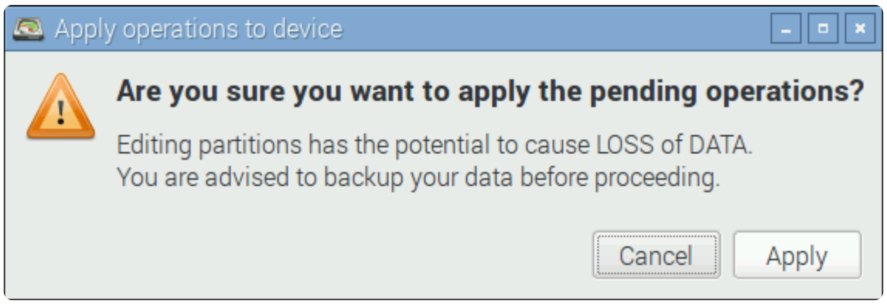
Nothing has actually changed yet; all operations are queued up. Clicking the “**Apply**” button (the green check mark) will commence reorganizing the data and partitions on the card…



Partition	File System	Label	Size	Used	Unused	Flags
New Partition #1	🟩 fat32	boot	5.26 GiB	---	---	
/dev/sda2	■ ext4		1.95 GiB	1.07 GiB	901.77 MiB	

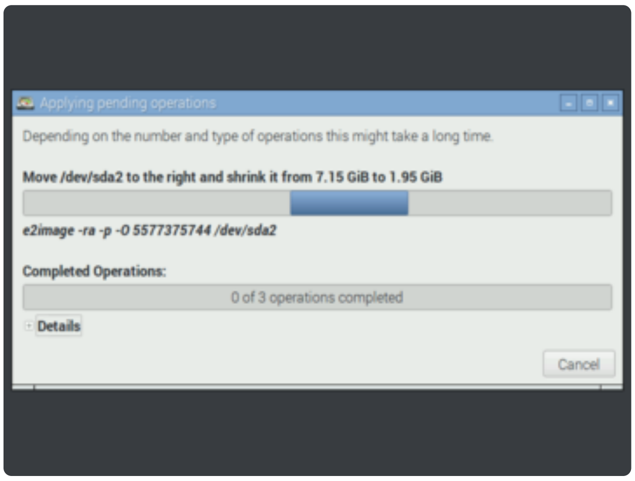
```
Move /dev/sda2 to the right and shrink it from 7.15 GiB to 1.95 GiB
Delete /dev/sda1 (fat16, 63.00 MiB) from /dev/sda
Create Primary Partition #1 (fat32, 5.26 GiB) on /dev/sda
3 operations pending
```

…but not without warning you first. Take heed, if something goes amiss you might end up with a scrambled card and have to start over from scratch. This is why it’s a good idea to make a backup before any serious disk editing.


Apply operations to device

Are you sure you want to apply the pending operations?

Editing partitions has the potential to cause LOSS of DATA.
You are advised to backup your data before proceeding.


Applying pending operations

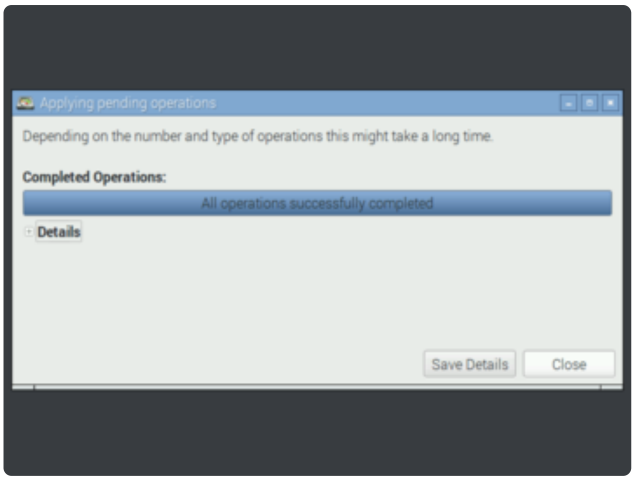
Depending on the number and type of operations this might take a long time.

Move /dev/sda2 to the right and shrink it from 7.15 GiB to 1.95 GiB

```
e2image -ra -p -O 5577375744 /dev/sda2
```

Completed Operations:

0 of 3 operations completed

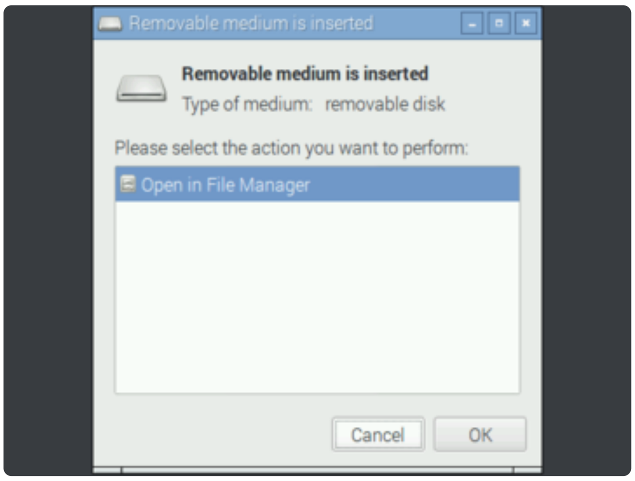

Applying pending operations

Depending on the number and type of operations this might take a long time.

Completed Operations:

All operations successfully completed

Once you click “Apply,” the system will be moving around a lot of data…these operations can take a long time, so this is a good opportunity to take a break or tidy up your desk or something.


Removable medium is inserted

Removable medium is inserted

Type of medium: removable disk

Please select the action you want to perform:

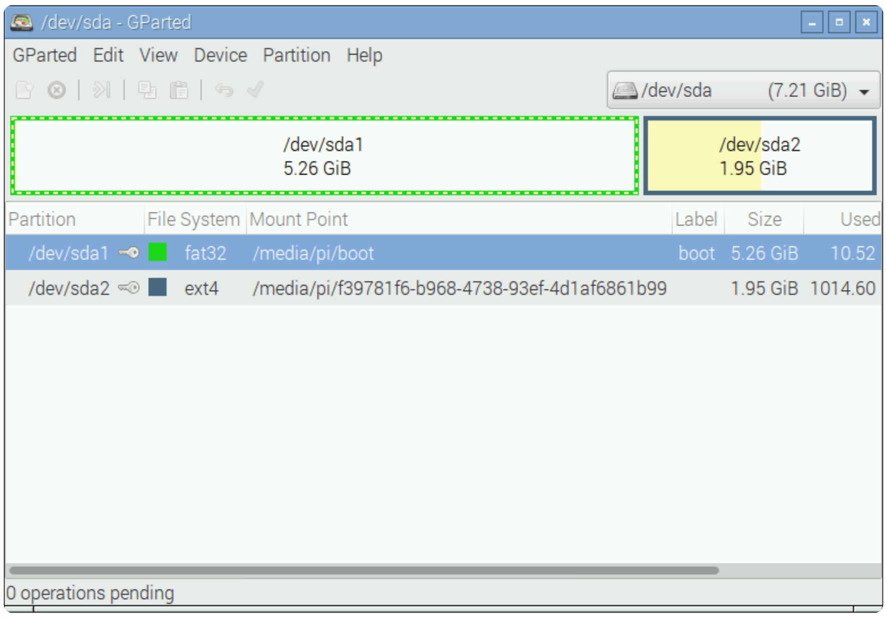
Open in File Manager

Once those operations are complete, you'll see the same partition map again, but without the "pending" messages at the bottom...everything is now "real" on the card.

To mount the new partitions, it's easiest just to **unplug and re-plug the USB card adapter**. You'll get a "Removable medium" dialog box when you do this (two, actually...one for the boot partition, one for root). Just click "**Cancel**" for each. We want the partitions mounted but don't need to browse the files.

Back in gparted now, select **GParted→Refresh Devices** and then select **/dev/sda** from the device list.

It should resemble the following, with a key next to each partition (indicating that it's mounted). This is with an 8 GB card, with Raspbian Jessie Lite in the root partition, the rest allocated to the /boot partition. There might be one or more small unallocated spaces depending on your card.



Partition	File System	Mount Point	Label	Size	Used
/dev/sda1	fat32	/media/pi/boot	boot	5.26 GiB	10.52
/dev/sda2	ext4	/media/pi/f39781f6-b968-4738-93ef-4d1af6861b99		1.95 GiB	1014.60

0 operations pending

Restore /boot Data

Last step now is to restore the data we saved from the original /boot partition. In a Terminal window, type the following commands.

(If you encounter an error, **do not** continue with the next command, since the last step **deletes** all the saved data! Check your typing and that the named directories exist, etc.)

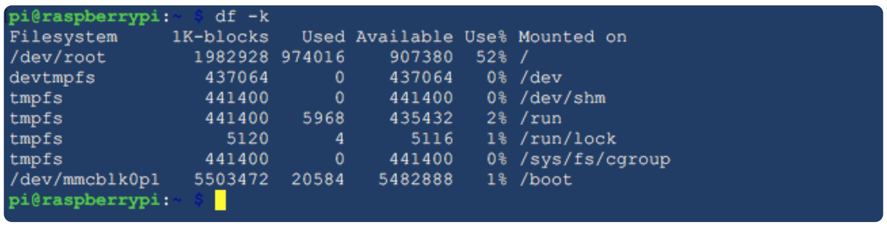
```
cd
sudo cp -r boot /media/pi
sudo umount /media/pi/*
sudo rm -r boot
```

Done!

You can quit from gparted now and unplug the SD card reader.

Perform an orderly shutdown of your setup system (**Pi Menu→Shutdown…**), then swap the SD cards and confirm the system can still boot from the newly-reconfigured card.

Since mine was a Jessie Lite system, this boots to a command line. I can log in and type “**df -k**” to see the partition sizes and free space, in kilobytes (the mount points “/” and “/boot” are of interest — the root and boot partitions, respectively).



```
pi@raspberrypi:~ $ df -k
Filesystem     1K-blocks   Used Available Use% Mounted on
/dev/root        1982928 974016    907380  52% /
devtmpfs          437064      0    437064   0% /dev
tmpfs             441400      0    441400   0% /dev/shm
tmpfs             441400   5968    435432   2% /run
tmpfs               5120      4      5116   1% /run/lock
tmpfs             441400      0    441400   0% /sys/fs/cgroup
/dev/mmcblk0p1   5503472  20584   5482888   1% /boot
pi@raspberrypi:~ $
```

Bonus! Shrinking Images

Once you create an image from an SD card, the resulting file is always the full capacity of the card, which may be way more than the actual data storage size. Copy the script below and paste it in a new file…suppose it’s called **script.sh**…which can then be executed as follows to shrink the image file to a more manageable size:

```
sudo bash ./script.sh pi.img
```

The script:

```
#!/bin/env bash

IMG="$1"

if [[ -e $IMG ]]; then
  P_START=$( fdisk -lu $IMG | grep Linux | awk '{print $2}' ) # Start of 2nd
partition in 512 byte sectors
  P_SIZE=$(( $( fdisk -lu $IMG | grep Linux | awk '{print $4}' ) * 512 )) #
```

```
Partition size in bytes
  losetup /dev/loop2 $IMG -o $(($P_START * 512)) --sizelimit $P_SIZE
  fsck -f /dev/loop2
  resize2fs -M /dev/loop2 # Make the filesystem as small as possible
  fsck -f /dev/loop2
  P_NEWSIZE=$( dumpe2fs /dev/loop2 2>/dev/null | grep '^Block count:' | awk '{print
$3}' ) # In 4k blocks
  P_NEWEND=$(( $P_START + ($P_NEWSIZE * 8) - 1 )) # in 512 byte sectors
  losetup -d /dev/loop2
  echo -e "p\nd\n2\nn\np\n2\n$P_START\n$P_NEWEND\np\nw\n" | fdisk $IMG
  I_SIZE=$((($P_NEWEND + 1) * 512)) # New image size in bytes
  truncate -s $I_SIZE $IMG
else
  echo "Usage: $0 filename"
fi
```

After writing this image to a card and booting from it, the filesystem can be re-expanded to the full card using:

```
sudo raspi-config --expand-rootfs
```