SINGAPORE POLYTECHNIC

SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING

ET0104 Embedded Computer Systems Laboratory

Laboratory 10 – Universal Serial Bus Application

1. Objectives

- To understand the application of USB by mounting a USB drive on a RPi3B+.
- To format the USB drive to a FAT File System.

In this lab, we illustrate the procedure to mount and format a USB drive on a linux system in a Raspberry Pi 3B+. We will be using Linux commands to achieve our objectives.

2. Introduction

USB drives are typically small, portable data storage devices typically used for data transfer. USB drives are block storage devices which implies that they will be formatted to a particular file system. This lab utilizes one of the so-called FAT file systems, chosen because of its compatibility with virtually all modern operating systems. Note however that while the FAT file system is ubiquitous, it does not compare particularly well performance-wise against other, more modern file systems.

The drive-mounting process for many Unix-based systems (including RPi OS) is more complicated, and therefore time- & labour-intensive than it is for a Mac (or PC). There are now Linux "desktop systems" that mount external drives when plugged into the system - same as with the Mac. But many, including RPi OS 'Lite', still require a "manual" mount process.

In this lab, we will be using the exFAT, or exfat file system on USB thumb drives connected to the RPi. exFAT was chosen for the simple reasons that: a) it's supported by Linux, MacOS and Windows, and b) it doesn't have the limits on file size that FAT & FAT32 do.

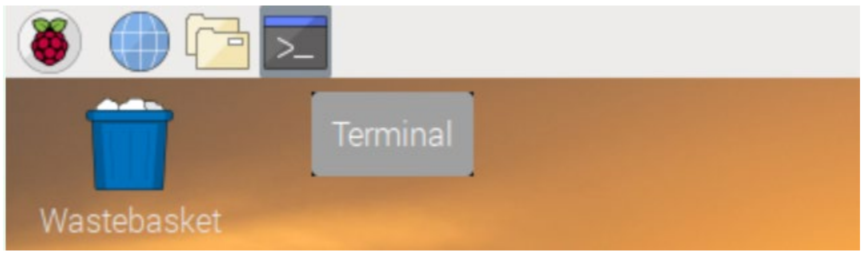
Warning: you will lose all the data in your USB Thumb drive!

Please back up your data before starting this experiment!

3. Experiment

1. Determine what drives are currently connected to the RPi

On the Raspberry Pi 3B+ desktop, **open the terminal program** from the taskbar.



Before we plug our external drive into the RPi, let's check to learn what drives are already connected by typing:

```
$ sudo fdisk --list
```

The RPi has an SD card, the listing (using the command above) will include one device named /dev/mmcblk0.Make sure you can see the listing.

A device name refers to the entire disk; in this case /dev/mmcblk0 is the entire SD card.

Device names are usually cryptic abbreviations such as: **/dev/sda, /dev/sdb**, or in this case **/dev/mmcblk0**. The /dev identifies it as a device and is followed by a name. The "mmc" part of the device name refers to "multimedia card".

As we shall see shortly, another common type of device is named "sd", which refers to "SCSI driver" - not Secure Digital. sd device names are also used for USB drives.

We've now seen the output fdisk --list produces. We shall not use it again here as fdisk is primarily a tool for formatting and partitioning block devices. As we've seen, fdisk produces a lot of output that we don't need now, but it's instructive to see what it does.

Next, compare the output of fdisk to that of the lsblk tool; lsblk gives us what we need for the task of mounting an external drive for the RPi.

Type in the following command:

```
pi@raspberrypi3b:~ $ lsblk --fs
```

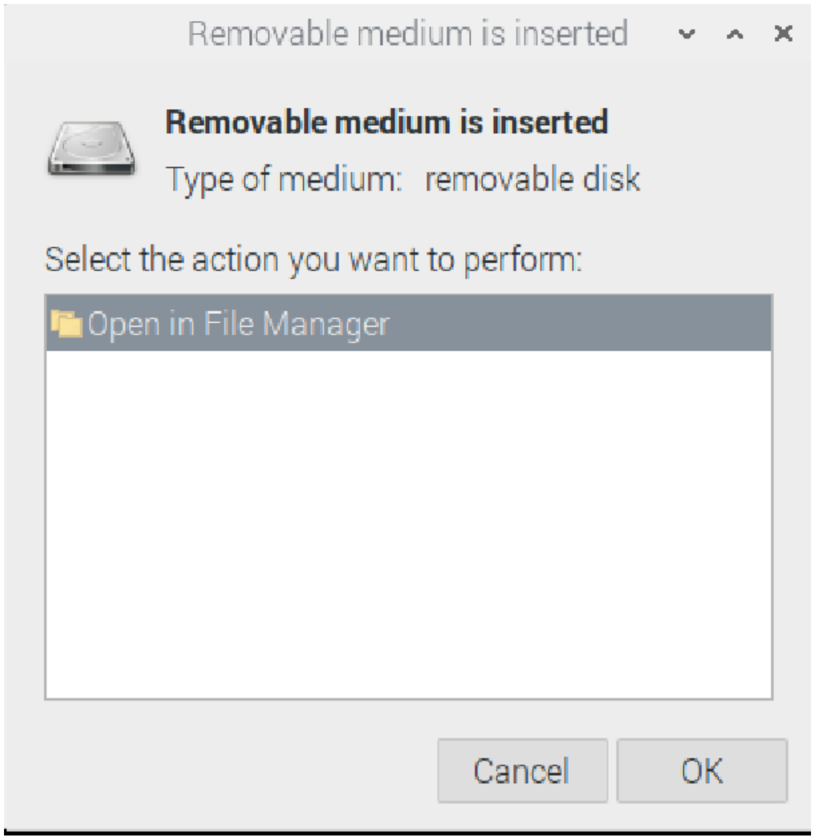
which yields the following:

```
NAME        FSTYPE FSVER LABEL      UUID                                 FSAVAIL FSUSE%
MOUNTPOINT
mmcblk0
├─mmcblk0p1 vfat   FAT32 boot       19E2-67CF                             200.9M    20%
/boot
└─mmcblk0p2 ext4   1.0   rootfs     97ca6ca8-5cb1-413f-84d0-569efd4e2c0f   25.8G     7% /
```

Here we see again the SD card device (mmcblk0), and its two partitions: p2 (root, /) and p1 (/boot). We also note that /boot is reported as formatted in vfat (a variant on FAT) under the heading FSTYPE, whereas root (/) is formatted in ext4. Having established our baseline, we'll move on to the next step.

2. Plug the USB drive in the RPi, partition and format it

Plug a USB drive in to the RPi, if you see below pop-up window, just tick "**Cancel**".


Removable medium is inserted

Removable medium is inserted

Type of medium: removable disk

Select the action you want to perform:

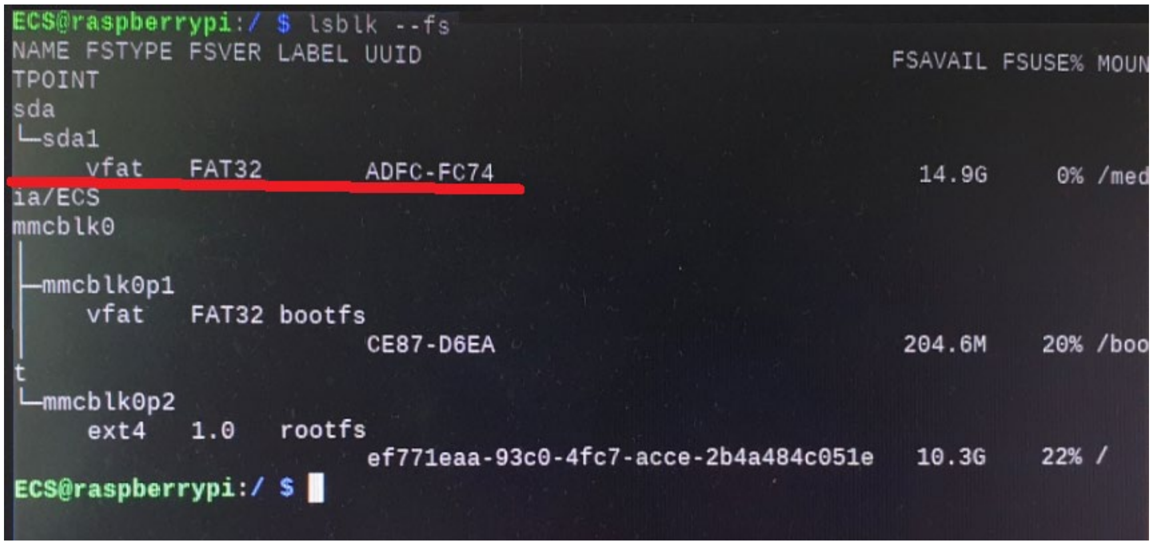
Then go back to the Terminal window and run *lsblk --fs* again at the RPi command line:

```
$ lsblk -fs
```

You may get a similar (not same) output as below:

```
NAME        FSTYPE FSVER LABEL  UUID                                 FSAVAIL FSUSE%
MOUNTPOINT
sda
mmcblk0
├─mmcblk0p1 vfat   FAT32 boot   19E2-67CF                             200.9M    20% /boot
└─mmcblk0p2 ext4   1.0   rootfs 97ca6ca8-5cb1-413f-84d0-569efd4e2c0f   25.8G     7% /
```

Note in the lsblk output above the MOUNTPOINT column is empty for sda, and that sda does not have a partition, e.g., sda1
The absence of a MOUNTPOINT for sda simply reflects the fact that this drive is not yet mounted.
However, when you plug in a USB thumb drive that you ever used in Windows system for your Laptop PC, you may get a similar output as below:



```
ECS@raspberrypi:/ $ lsblk --fs
NAME FSTYPE FSVER LABEL UUID                                 FSAVAIL FSUSE% MOUN
TPOINT
sda
└─sda1
     vfat   FAT32       ADFC-FC74                              14.9G     0% /med
ia/ECS
mmcblk0
│
├─mmcblk0p1
│    vfat   FAT32 bootfs
│                       CE87-D6EA                             204.6M    20% /boo
t
└─mmcblk0p2
     ext4   1.0   rootfs
                        ef771eaa-93c0-4fc7-acce-2b4a484c051e   10.3G    22% /
ECS@raspberrypi:/ $ 
```

It has been partitioned into “Microsoft basic data” and “FAT32”. It could be “sda” or “sdb”, or “sdc”, depending on the system allocation of the port.

Let’s digress briefly to review file systems first. File system formats and file system partitions:

NOTE 1: File systems and formats

A filesystem is about the structure for organizing the data stored on a non-volatile memory storage device. This structure is also called the format, and the process of *formatting* a drive is just applying the chosen filesystem’s data structure to the drive/device/partition.
As mentioned earlier, we will be using a FAT filesystem for thumb drives due to the fact that FAT can be read and written on all the major OSs: Mac, Linux/Unix and Windows. FAT is a simple filesystem, but this is complicated somewhat by the different ***flavors*** of FAT; e.g. FAT16, FAT32, exFAT. Here, we used the exFAT *flavor* because it will accommodate larger partition sizes than the other *FAT flavors*.

NOTE 2: Partitions and their uses

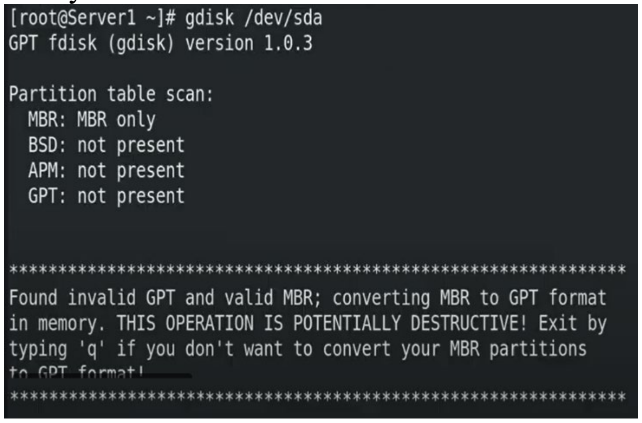
The thumb drive itself is a device. It contains a mass of unallocated memory storage, but that unallocated memory is virtually useless to our operating system until it is: 1) partitioned, and 2) formatted. The partitioning process is simply dividing the memory storage on the device into “blocks” of data that can be formatted - or structured - by writing a filesystem onto the partition. A partition can [may] encompass the entire device (i.e., all of its memory), or it can cover only a very small slice of the device's memory. The size of the partition should reflect its intended usage.

Partition:

Type the following command:

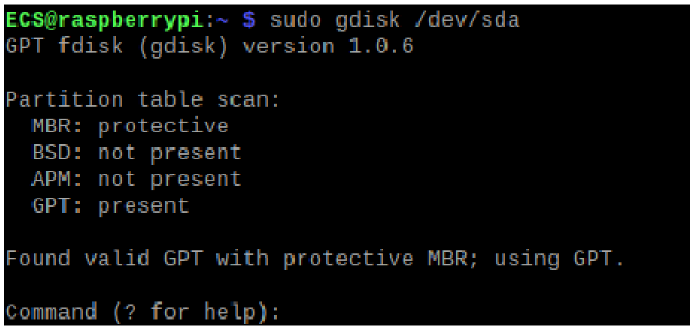
```
$ sudo gdisk /dev/sda
```

If your thumb drive was ever formatted with FAT32 (used for Windows), you may see below similar output on your screen:



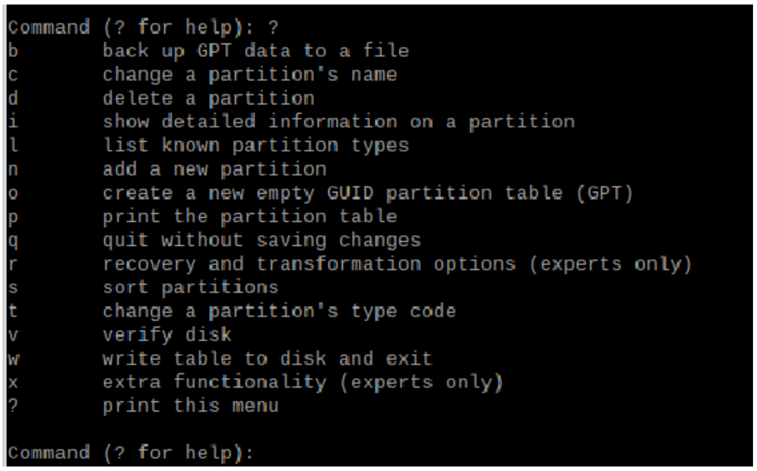
```
[root@Server1 ~]# gdisk /dev/sda
GPT fdisk (gdisk) version 1.0.3

Partition table scan:
  MBR: MBR only
  BSD: not present
  APM: not present
  GPT: not present


****************************************************************
Found invalid GPT and valid MBR; converting MBR to GPT format
in memory. THIS OPERATION IS POTENTIALLY DESTRUCTIVE! Exit by
typing 'q' if you don't want to convert your MBR partitions
to GPT format!
****************************************************************
```

Or,



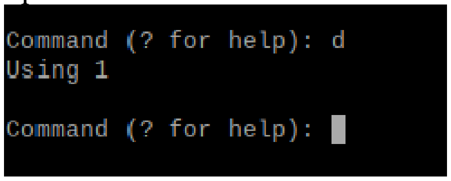
```
ECS@raspberrypi:~ $ sudo gdisk /dev/sda
GPT fdisk (gdisk) version 1.0.6

Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: present

Found valid GPT with protective MBR; using GPT.

Command (? for help):
```

Type "?" to list down all the commands in gdisk:



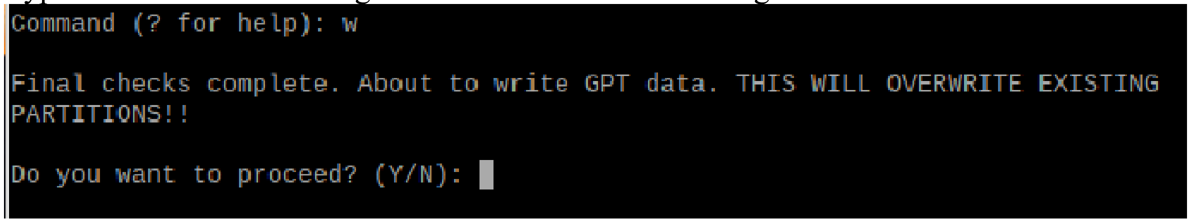
```
Command (? for help): ?
b       back up GPT data to a file
c       change a partition's name
d       delete a partition
i       show detailed information on a partition
l       list known partition types
n       add a new partition
o       create a new empty GUID partition table (GPT)
p       print the partition table
q       quit without saving changes
r       recovery and transformation options (experts only)
s       sort partitions
t       change a partition's type code
v       verify disk
w       write table to disk and exit
x       extra functionality (experts only)
?       print this menu

Command (? for help):
```

Type “d” to delete the current partition on the thumb drive:



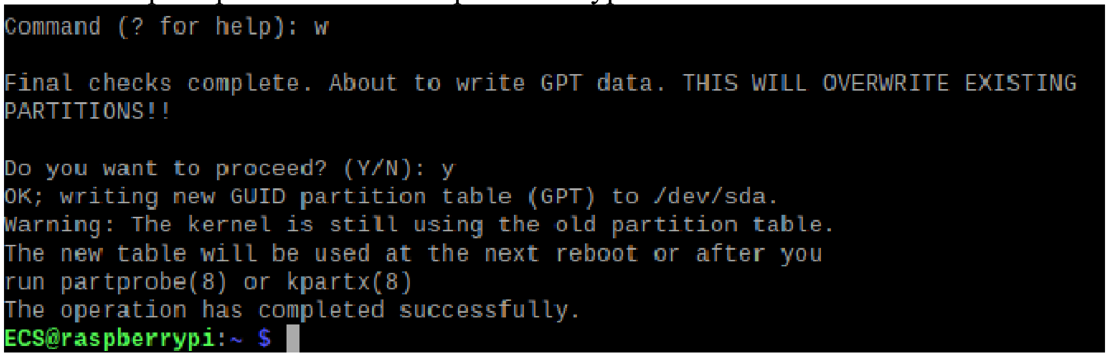
```
Command (? for help): d
Using 1

Command (? for help):
```

All the data in the thumb drive will be lost after this command!
Type “w” to make the change on the thumb drive and exit gdisk:



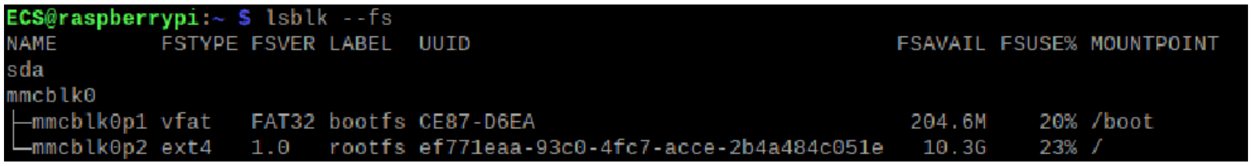
```
Command (? for help): w

Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING
PARTITIONS!!

Do you want to proceed? (Y/N):
```

You will be prompted to confirm to proceed. Type ‘Y’ to continue.



```
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
OK; writing new GUID partition table (GPT) to /dev/sda.
Warning: The kernel is still using the old partition table.
The new table will be used at the next reboot or after you
run partprobe(8) or kpartx(8)
The operation has completed successfully.
ECS@raspberrypi:~ $
```

Now, plug out the thumb drive, wait for 3 – 5 seconds, then plug in the thumb drive again. run *lsblk --fs* again at the RPi command line:

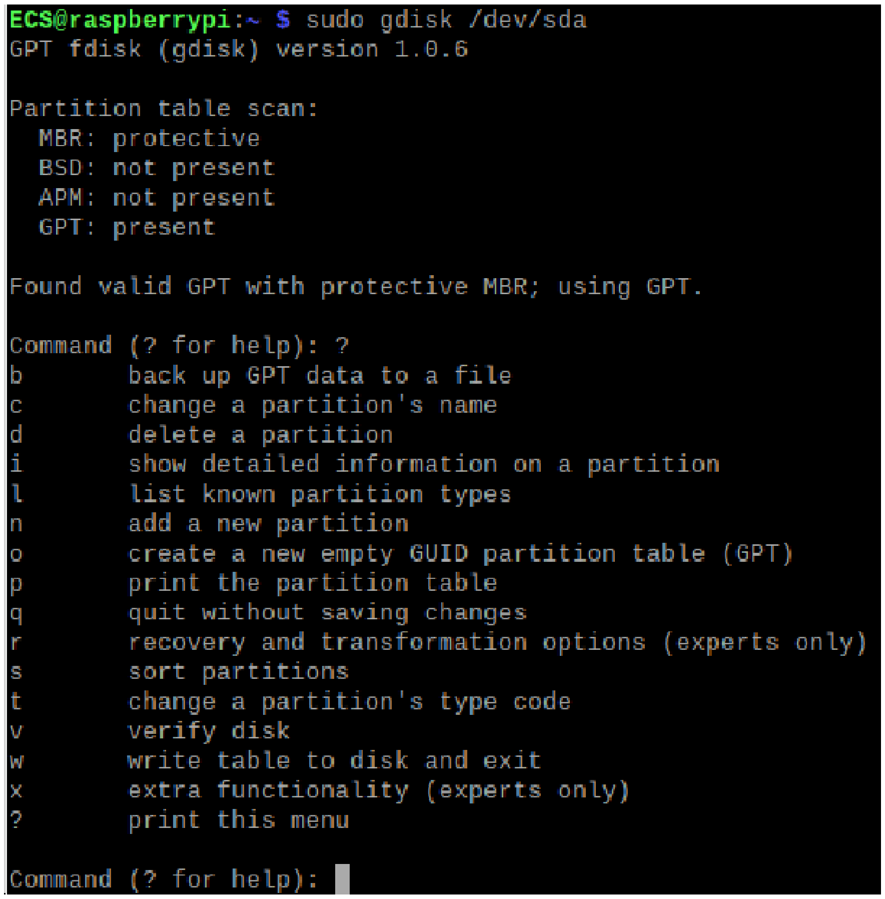


```
ECS@raspberrypi:~ $ lsblk --fs
NAME        FSTYPE FSVER LABEL  UUID                                 FSAVAIL FSUSE% MOUNTPOINT
sda
mmcblk0
├─mmcblk0p1 vfat   FAT32 bootfs CE87-D6EA                             204.6M    20% /boot
└─mmcblk0p2 ext4   1.0   rootfs ef771eaa-93c0-4fc7-acce-2b4a484c051e   10.3G    23% /
```

There is no partition under sda now.
Type the following command again:

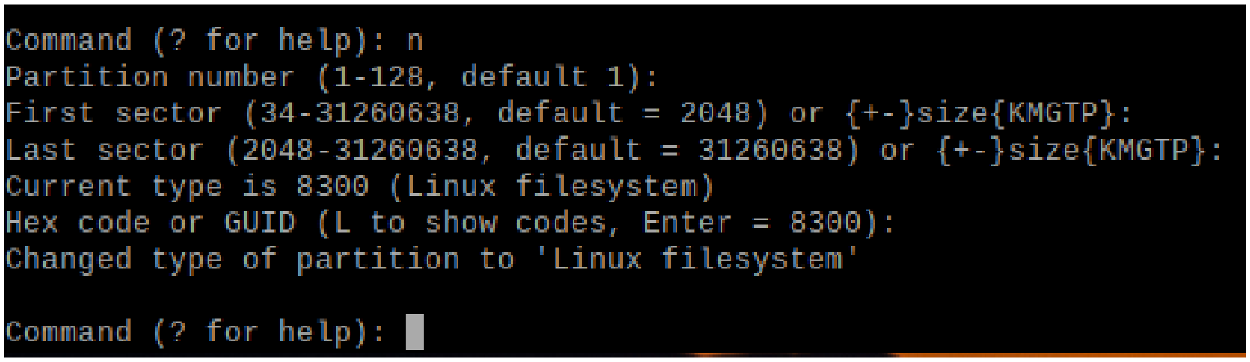
```
$ sudo gdisk /dev/sda
```

After seeing the line of “Command (? For help):”, type “?” to get the full list of gdisk commands.



```
ECS@raspberrypi:~ $ sudo gdisk /dev/sda
GPT fdisk (gdisk) version 1.0.6

Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: present

Found valid GPT with protective MBR; using GPT.

Command (? for help): ?
b       back up GPT data to a file
c       change a partition's name
d       delete a partition
i       show detailed information on a partition
l       list known partition types
n       add a new partition
o       create a new empty GUID partition table (GPT)
p       print the partition table
q       quit without saving changes
r       recovery and transformation options (experts only)
s       sort partitions
t       change a partition's type code
v       verify disk
w       write table to disk and exit
x       extra functionality (experts only)
?       print this menu

Command (? for help):
```

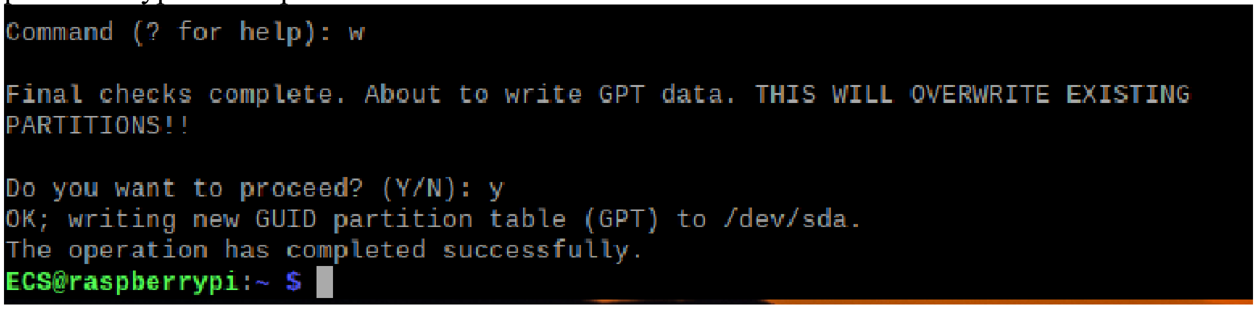
Now, type “n” to add in a new partition.
For all the options, just use the default setting by click “Enter” key on your keyboard.



```
Command (? for help): n
Partition number (1-128, default 1):
First sector (34-31260638, default = 2048) or {+-}size{KMGTP}:
Last sector (2048-31260638, default = 31260638) or {+-}size{KMGTP}:
Current type is 8300 (Linux filesystem)
Hex code or GUID (L to show codes, Enter = 8300):
Changed type of partition to 'Linux filesystem'

Command (? for help):
```

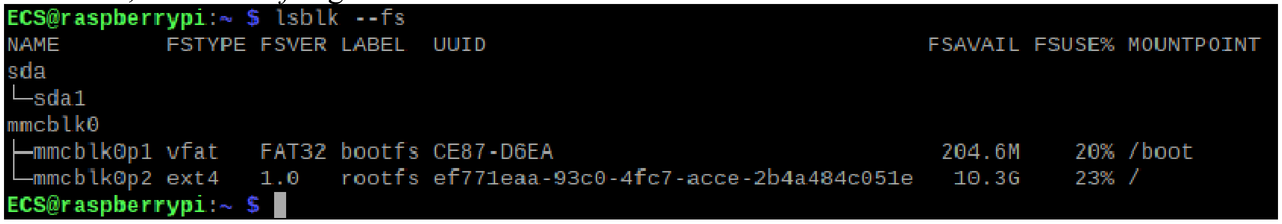
You can see now the type of partition is changed to “Linux filesystem”.

Now, type “w” to confirm the changes and exit gdisk. Again, you will be asked “Y/N” to proceed. Type ‘Y’ to proceed.



```
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
OK; writing new GUID partition table (GPT) to /dev/sda.
The operation has completed successfully.
ECS@raspberrypi:~ $
```

After that, run *lsblk --fs* again at the RPi command line:



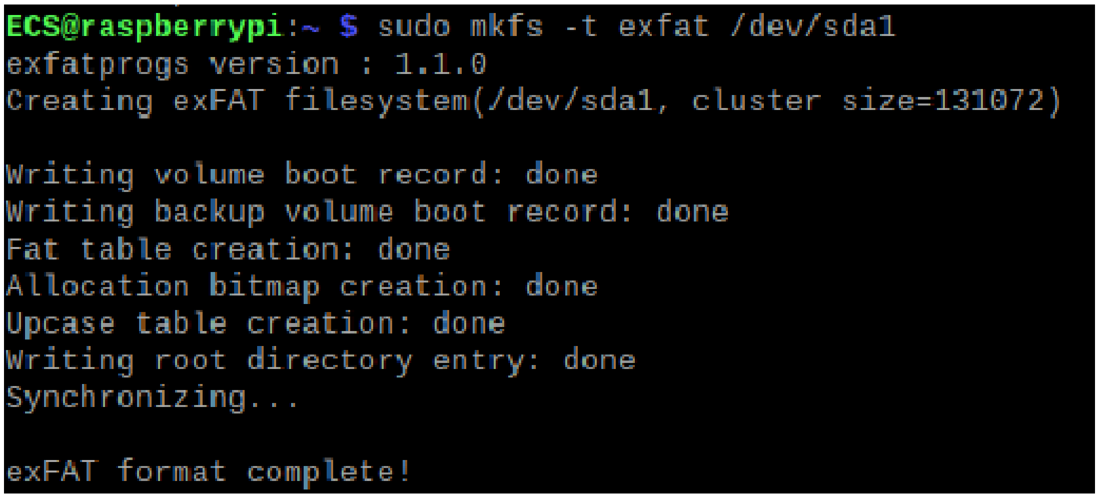
```
ECS@raspberrypi:~ $ lsblk --fs
NAME        FSTYPE FSVER LABEL  UUID                                 FSAVAIL FSUSE% MOUNTPOINT
sda
└─sda1
mmcblk0
├─mmcblk0p1 vfat   FAT32 bootfs CE87-D6EA                              204.6M    20% /boot
└─mmcblk0p2 ext4   1.0   rootfs ef771eaa-93c0-4fc7-acce-2b4a484c051e   10.3G    23% /
ECS@raspberrypi:~ $
```

This shows that we have successfully created a new partition on /dev/sda, namely /dev/sda1.

Format:
Type the command below:

```
$ sudo mkfs -t exfat /dev/sda1
```

You should get a similar output as shown below:

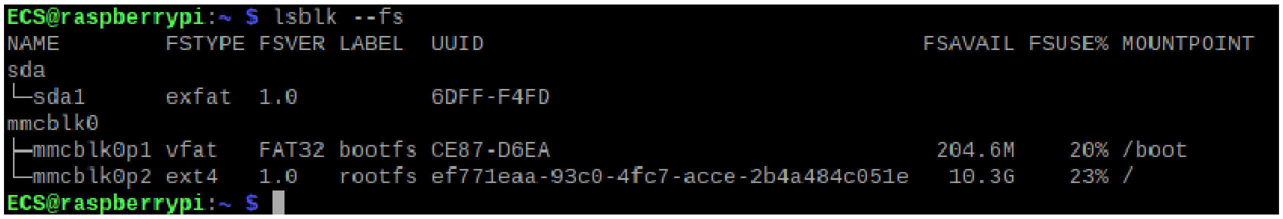


```
ECS@raspberrypi:~ $ sudo mkfs -t exfat /dev/sda1
exfatprogs version : 1.1.0
Creating exFAT filesystem(/dev/sda1, cluster size=131072)

Writing volume boot record: done
Writing backup volume boot record: done
Fat table creation: done
Allocation bitmap creation: done
Upcase table creation: done
Writing root directory entry: done
Synchronizing...

exFAT format complete!
```

Type the command below:

```
$ lsblk --fs
```

You will get similar output as below:

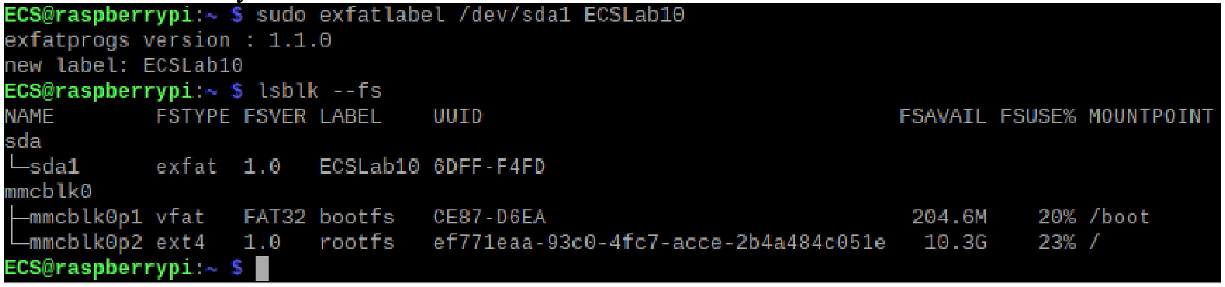


```
ECS@raspberrypi:~ $ lsblk --fs
NAME        FSTYPE FSVER LABEL  UUID                                 FSAVAIL FSUSE% MOUNTPOINT
sda
└─sda1      exfat  1.0          6DFF-F4FD
mmcblk0
├─mmcblk0p1 vfat   FAT32 bootfs CE87-D6EA                             204.6M    20% /boot
└─mmcblk0p2 ext4   1.0   rootfs ef771eaa-93c0-4fc7-acce-2b4a484c051e   10.3G    23% /
ECS@raspberrypi:~ $ 
```

Now we change the label name for this thumb drive by typing below command:

```
$ sudo exfatlabel /dev/sda1 ECSLab10
```

Then we can verify with “lsblk –fs”.



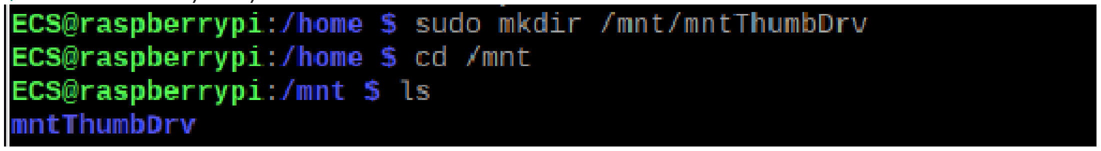
```
ECS@raspberrypi:~ $ sudo exfatlabel /dev/sda1 ECSLab10
exfatprogs version : 1.1.0
new label: ECSLab10
ECS@raspberrypi:~ $ lsblk --fs
NAME        FSTYPE FSVER LABEL    UUID                                 FSAVAIL FSUSE% MOUNTPOINT
sda
└─sda1      exfat  1.0   ECSLab10 6DFF-F4FD
mmcblk0
├─mmcblk0p1 vfat   FAT32 bootfs   CE87-D6EA                             204.6M    20% /boot
└─mmcblk0p2 ext4   1.0   rootfs   ef771eaa-93c0-4fc7-acce-2b4a484c051e   10.3G    23% /
ECS@raspberrypi:~ $ 
```

We now have a partitioned and formatted exfat USB drive.

3. Mount the USB drive

Before a drive can be mounted, a mount point is needed in the RPi's file system; let's do that.
We can have USB mount point under /media or /mnt.
Here's the command to create a mount point under /mnt:

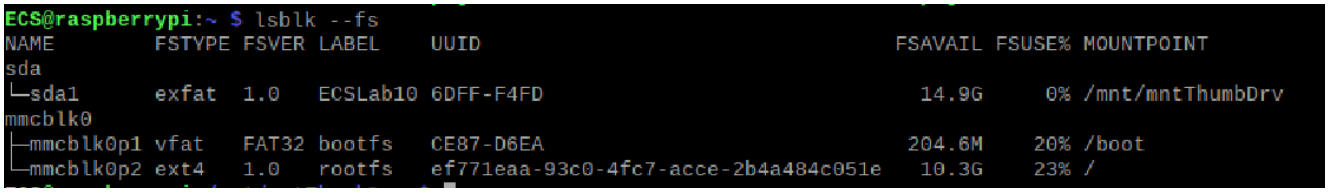
```
$ sudo mkdir /mnt/mntThumbDrv
```



```
ECS@raspberrypi:/home $ sudo mkdir /mnt/mntThumbDrv
ECS@raspberrypi:/home $ cd /mnt
ECS@raspberrypi:/mnt $ ls
mntThumbDrv
```

A mount point is just a directory in the computer's file system! Furthermore, that directory is (typically) empty until the mount is completed. Only then does the directory contain any data.
We need for a mount and confirm it with "lsblk –fs":

```
$ sudo mount -t exfat /dev/sda1 /mnt/mntThumbDrv

$ lsblk --fs
```



```
ECS@raspberrypi:~ $ lsblk --fs
NAME         FSTYPE FSVER LABEL    UUID                                 FSAVAIL FSUSE% MOUNTPOINT
sda
└─sda1       exfat  1.0   ECSLab10 6DFF-F4FD                              14.9G     0% /mnt/mntThumbDrv
mmcblk0
├─mmcblk0p1 vfat   FAT32 bootfs   CE87-D6EA                             204.6M    20% /boot
└─mmcblk0p2 ext4   1.0   rootfs   ef771eaa-93c0-4fc7-acce-2b4a484c051e   10.3G    23% /
```

Now you can copy files to the folder: /mnt/mntThumbDrv. All files there are actually written to the USB thumb drive.
Notice that it is using "Linux filesystem". This thumb drive may not be recognized in your Windows PC. If so, please use the following steps to recover it back to the "Microsoft basic data".

1. Type the following command again:

   ```
   $ sudo gdisk /dev/sda
   ```

2. Type '?' to list down all the gdisk commands
3. Type 'd' to delete the current partition on the thumb drive. Then Type 'w' to write the changes to the drive and exit gdisk.
4. Take out the thumb drive, wait for 3 – 5 seconds, plug in the thumb drive again.
5. Run *lsblk --fs* again at the RPi command line and confirm there is no partition under sda.
6. Type the following command again:

   ```
   $ sudo gdisk /dev/sda
   ```

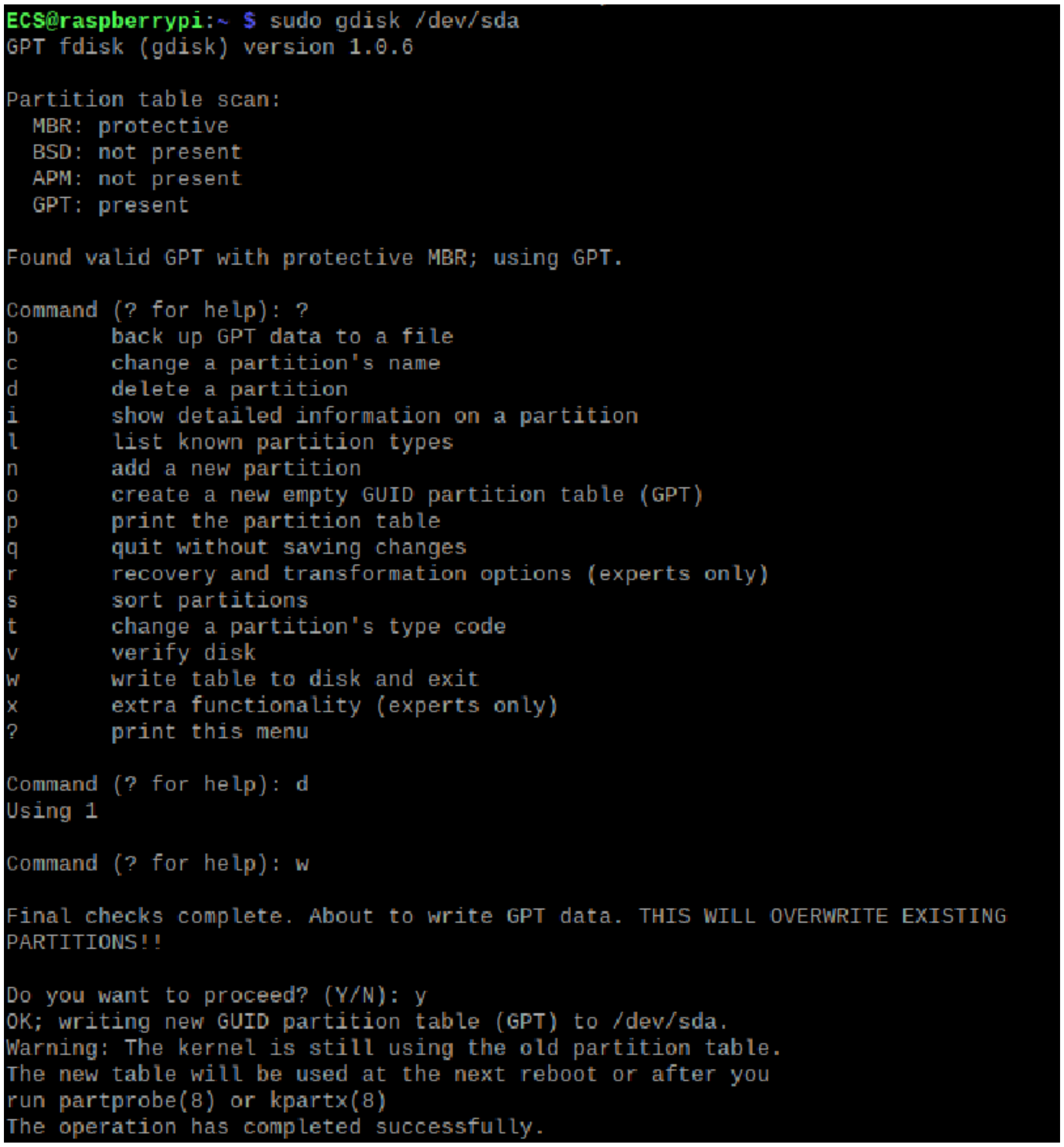
7. Type '?' to list down all the gdisk commands
8. Type 'n' to add in a new partition. For all the options, just use the default setting by click

   "Enter" key on your keyboard, **except for the last option: "changed type of partition to: ". For this option, type "0700" which is the code for "Microsoft basic data".**

   Then Type 'w' to write the changes to the drive and exit gdisk.
9. Type the following commands to format the thumb drive into FAT32:

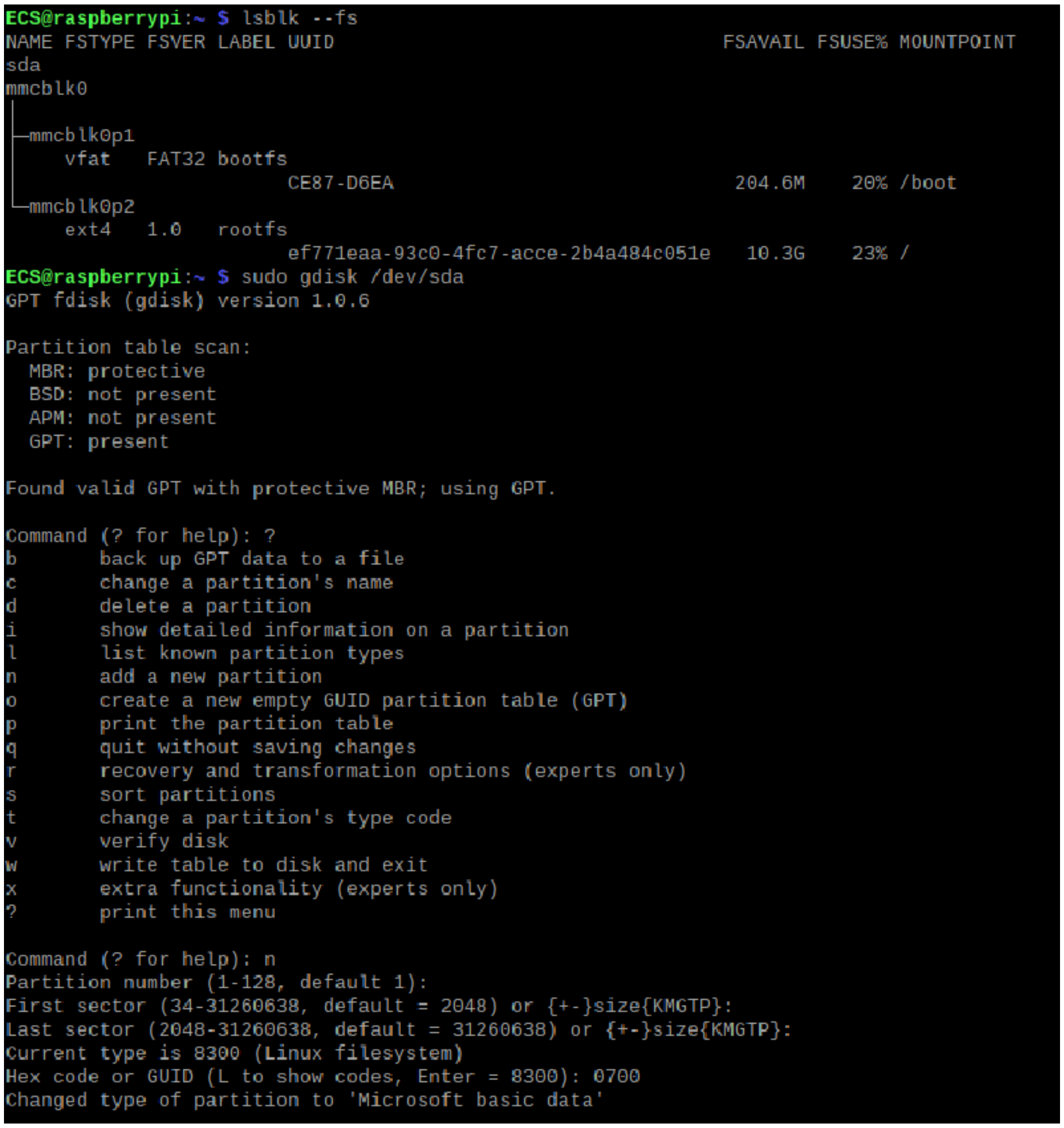
```
$ sudo mkfs.vfat /dev/sda1
$ sudo fsck /dev/sda1
```

You may refer to the screenshots #1, #2, and #3 shown below for the above procedures .



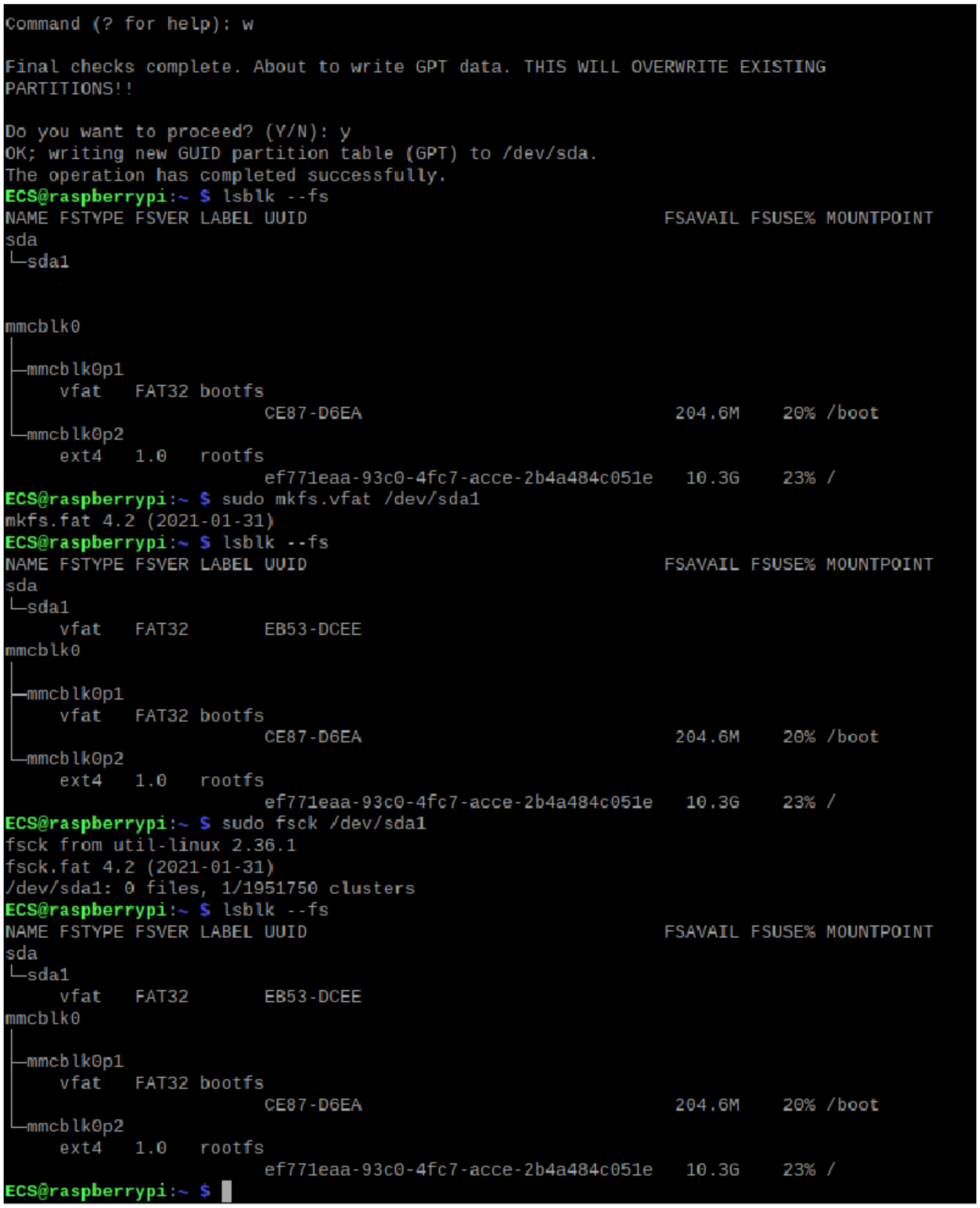
```
ECS@raspberrypi:~ $ sudo gdisk /dev/sda
GPT fdisk (gdisk) version 1.0.6

Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: present

Found valid GPT with protective MBR; using GPT.

Command (? for help): ?
b       back up GPT data to a file
c       change a partition's name
d       delete a partition
i       show detailed information on a partition
l       list known partition types
n       add a new partition
o       create a new empty GUID partition table (GPT)
p       print the partition table
q       quit without saving changes
r       recovery and transformation options (experts only)
s       sort partitions
t       change a partition's type code
v       verify disk
w       write table to disk and exit
x       extra functionality (experts only)
?       print this menu

Command (? for help): d
Using 1

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
OK; writing new GUID partition table (GPT) to /dev/sda.
Warning: The kernel is still using the old partition table.
The new table will be used at the next reboot or after you
run partprobe(8) or kpartx(8)
The operation has completed successfully.
```

Screenshot #1



```
ECS@raspberrypi:~ $ lsblk --fs
NAME FSTYPE FSVER LABEL UUID                                 FSAVAIL FSUSE% MOUNTPOINT
sda
mmcblk0
│
├─mmcblk0p1
│    vfat   FAT32 bootfs
│                       CE87-D6EA                               204.6M    20% /boot
└─mmcblk0p2
     ext4   1.0   rootfs
                        ef771eaa-93c0-4fc7-acce-2b4a484c051e   10.3G    23% /
ECS@raspberrypi:~ $ sudo gdisk /dev/sda
GPT fdisk (gdisk) version 1.0.6

Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: present

Found valid GPT with protective MBR; using GPT.

Command (? for help): ?
b       back up GPT data to a file
c       change a partition's name
d       delete a partition
i       show detailed information on a partition
l       list known partition types
n       add a new partition
o       create a new empty GUID partition table (GPT)
p       print the partition table
q       quit without saving changes
r       recovery and transformation options (experts only)
s       sort partitions
t       change a partition's type code
v       verify disk
w       write table to disk and exit
x       extra functionality (experts only)
?       print this menu

Command (? for help): n
Partition number (1-128, default 1):
First sector (34-31260638, default = 2048) or {+-}size{KMGTP}:
Last sector (2048-31260638, default = 31260638) or {+-}size{KMGTP}:
Current type is 8300 (Linux filesystem)
Hex code or GUID (L to show codes, Enter = 8300): 0700
Changed type of partition to 'Microsoft basic data'
```

Screenshot #2



```
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
OK; writing new GUID partition table (GPT) to /dev/sda.
The operation has completed successfully.
ECS@raspberrypi:~ $ lsblk --fs
NAME FSTYPE FSVER LABEL UUID                                 FSAVAIL FSUSE% MOUNTPOINT
sda
└─sda1


mmcblk0
│
├─mmcblk0p1
     vfat   FAT32 bootfs
                        CE87-D6EA                              204.6M    20% /boot
└─mmcblk0p2
     ext4   1.0   rootfs
                        ef771eaa-93c0-4fc7-acce-2b4a484c051e   10.3G    23% /
ECS@raspberrypi:~ $ sudo mkfs.vfat /dev/sda1
mkfs.fat 4.2 (2021-01-31)
ECS@raspberrypi:~ $ lsblk --fs
NAME FSTYPE FSVER LABEL UUID                                 FSAVAIL FSUSE% MOUNTPOINT
sda
└─sda1
     vfat   FAT32       EB53-DCEE
mmcblk0
│
├─mmcblk0p1
     vfat   FAT32 bootfs
                        CE87-D6EA                              204.6M    20% /boot
└─mmcblk0p2
     ext4   1.0   rootfs
                        ef771eaa-93c0-4fc7-acce-2b4a484c051e   10.3G    23% /
ECS@raspberrypi:~ $ sudo fsck /dev/sda1
fsck from util-linux 2.36.1
fsck.fat 4.2 (2021-01-31)
/dev/sda1: 0 files, 1/1951750 clusters
ECS@raspberrypi:~ $ lsblk --fs
NAME FSTYPE FSVER LABEL UUID                                 FSAVAIL FSUSE% MOUNTPOINT
sda
└─sda1
     vfat   FAT32       EB53-DCEE
mmcblk0
│
├─mmcblk0p1
     vfat   FAT32 bootfs
                        CE87-D6EA                              204.6M    20% /boot
└─mmcblk0p2
     ext4   1.0   rootfs
                        ef771eaa-93c0-4fc7-acce-2b4a484c051e   10.3G    23% /
ECS@raspberrypi:~ $
```

Screenshot #3