# Part 55
# -
# Troubleshooting Your Pi

*Version: 2020-03-27*

**Cables**
Make sure that everything is properly plugged in to your Raspberry Pi – this includes power, HDMI, USB, Ethernet, audio, etc. If in doubt, give it a gentle jiggle in the slot and firmly push it in. Double-check the other end of the cables as well.

**microSD card**
Make sure a microSD card is inserted properly into the underside of your Raspberry Pi. Again, push it firmly in – if needed, take your Raspberry Pi out of its case before doing this.

**Monitor**
Connect to a monitor or display and turn it on before booting your Raspberry Pi – sometimes a Raspberry Pi won't output a video signal if there's nowhere for it to output to.

**Lightning bolt / not enough power**
A telltale sign of a Raspberry Pi not having enough power is a lightning bolt symbol appearing in the corner of the screen, or a peripheral/accessory not working properly. You need to make sure you have a powerful enough supply for your Raspberry Pi – this means a rating of 5 V 2.5 A for a Raspberry Pi 3B+ and below, or 5 V 3 A for a Raspberry Pi 4.

**Camera not detected**
Using a Raspberry Pi Camera Module is usually straightforward. However, if it won't respond to commands then it may likely be a couple of things.
First of all, check it's plugged in the correct way. Only one side of the ribbon cable has silver connectors and they need to line up with the connectors in the Camera slot.
Secondly, you may need to enable the Camera in the Raspberry Pi Configuration tool

**Reusing a microSD card**
Once upon a time, you were pretty easily able to transfer a working microSD card from one Raspberry Pi to another. Unfortunately, that doesn't always work any more, and you'll be forced to reinstall Raspbian (or your preferred OS) onto the card. Make sure to back up any important files to a USB stick or the cloud before doing so, though – it will completely wipe the card.

**No picture on Raspberry Pi 4**
We've all stumbled into this problem and spent far too long trying to fix it: long story short, Raspberry Pi 4 likes to boot from the 'HDMI 0' output, which is the one closest to the power port. You may just need to switch over the cable from the other HDMI port.

**Code won't run**
You've written out your code or copied it from the magazine and it won't run. What to do? You should get an error message when this happens, and it will list the problem and where it happened. If you don't understand the error message, make sure to check the line it mentions. If that doesn't help you spot an error, put the error message into your search engine of choice.

**Booting quick fix**
Having problems booting? Before you head over the page, try formatting and then reinstalling Raspbian (or whatever OS you prefer) and try again. While uncommon, installation issues can occur. This is especially common if you've just installed an OS.

**Blinking ACT light**
On the opposite end of the USB and Ethernet ports on a Raspberry Pi, you'll notice there are two little LEDs. The red one shows if there's power, while the green one is the ACT light. If the ACT light is blinking erratically during bootup, it means it's reading the microSD card fine. If it's not blinking at all, it means it can't read the microSD card and so you may need to use a new one or reinstall your OS

**No power LED**
Very simply, this means not enough or no power is getting to your Raspberry Pi. If the power supply drops below 4.65 V, this red LED will not light up. You may need to check your power supply for faults, including a broken connector, and replace it.
If you've checked your power supply and it's all working (on another device), if the problem Raspberry Pi isn't turning on then it's likely the board is broken.
Note: the first-generation Raspberry Pi model has a polyfuse – if this gets blown, it can reset/repair itself, but you may have to wait a few days for this to happen.

**Composite video**
Raspberry Pi can output a composite video signal via the 3.5 mm headphone jack. If you're using an older pre-installed NOOBS card, or still have some lying around, it's good to know that holding the number 3 key on a connected keyboard will force it to output via the composite output if that's what you're using. If you're in the US or another NTSC country, hold the 4 key instead.

**Corrupt microSD card**
While uncommon, microSD cards can sometimes become corrupted while being used in a Raspberry Pi. Unfortunately, there's no solution to this other than getting a new microSD card, so make sure to back up your card regularly. You can do this by saving any important files, or even by making an image from your current card. We have a video on how to do this here: magpi.cc/backupvid

**Power problems**
Raspberry Pi Zero only has an ACT LED, so you will need to check your power supply if it won't boot up. We suggest trying the power and microSD card on another Raspberry Pi when in doubt.

**Network speed differences**
Different Raspberry Pi models have different connection speeds, especially on Ethernet connections. Raspberry Pi 3 and below max out at about 100Mbps as they use a 100MB adapter over USB. Raspberry Pi 3B+ is roughly 280Mbps, as it has a Gigabit Ethernet connection on a USB 2 line. Raspberry Pi 4, however, is a full gigabit. If you're having speed issues, make sure your router or switch supports up to the speeds you're expecting, and test with different Ethernet cables.

**Networking booting and PoE**
Raspberry Pi 3/3B+ and 4 support network booting, and Raspberry Pi 3B+ and 4 even support Power over Ethernet with the PoE HAT. These are advanced methods and have some issues you might come across, such as DHCP timing out, ARP check failing, and others. For full details on how to set it up and troubleshoot, we suggest heading to the network booting documentation here: magpi.cc/netboot

**Wireless testing**
There are a lot of things that can interfere with your wireless reception in your home. Distance from your router, interference from neighbours, walls, and even just too many people on the same channel. We like to use apps like Wifi Analyzer to test locations around our home to find out the best spots to place wireless devices such as a Raspberry Pi.

**SSH connection**
Many security precautions have been implemented in the last couple of years to make sure Raspbian and Raspberry Pi are more secure – this includes making sure people change the password on their Raspberry Pi, and keeping SSH off. You can turn it on from the Interfaces tab in Raspberry Pi Configuration – or, if you're using a headless Raspberry Pi, simple place an empty file called ssh in the boot folder of the microSD card.

**Wired Raspberry Pi Zero**
Raspberry Pi Zero does not include an Ethernet port. However, if you need to use a wired connection, you can always use a USB Ethernet dongle. It may not be as speedy as other Raspberry Pi models, but it will do the job.

**Raspberry Pi Zero and A gadget issues**
A workaround for a lot of people wanting a portable, network connected computer is to put it into 'gadget' or USB boot mode. If you're having trouble and want to deactivate it, either flash the microSD card (see page 6) or head to the usbboot documentation to fix your setup: magpi.cc/usbboot

**System information**

***CPU information***

The command `cat /proc/cpuinfo` can be used to display information about the CPU on your Raspberry Pi. As you can see from the display, the processor is the ARMv7 Processor rev 3 (v71) .

```
# cat /proc/cpuinfo
processor       : 0
model name      : ARMv7 Processor rev 3 (v7l)
BogoMIPS        : 108.00
Features        : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt
vfpd32 lpae evtstrm crc32
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xd08
CPU revision    : 3

processor       : 1
model name      : ARMv7 Processor rev 3 (v7l)
BogoMIPS        : 108.00
Features        : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt
vfpd32 lpae evtstrm crc32
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xd08
CPU revision    : 3

processor       : 2
model name      : ARMv7 Processor rev 3 (v7l)
BogoMIPS        : 108.00
Features        : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt
vfpd32 lpae evtstrm crc32
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xd08
CPU revision    : 3

processor       : 3
model name      : ARMv7 Processor rev 3 (v7l)
BogoMIPS        : 108.00
Features        : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt
vfpd32 lpae evtstrm crc32
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xd08
CPU revision    : 3

Hardware        : BCM2711
Revision        : b03111
Serial          : 10000000374944bc
Model           : Raspberry Pi 4 Model B Rev 1.1
```

***CPU architecture information***
Information about the CPU architecture can be displayed by entering the command `lscpu`

```
# lscpu
Architecture:        armv7l
Byte Order:          Little Endian
CPU(s):              4
On-line CPU(s) list: 0-3
Thread(s) per core:  1
Core(s) per socket:  4
Socket(s):           1
Vendor ID:           ARM
Model:               3
Model name:          Cortex-A72
Stepping:            r0p3
CPU max MHz:         1500.0000
CPU min MHz:         600.0000
BogoMIPS:            108.00
Flags:               half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva
idivt vfpd32 lpae evtstrm crc32
```

***Operating system information***
Some useful operating system related information can be obtained by entering the following commands:

```
$ cat /proc/sys/kernel/hostname
raspberrypi

$ cat /proc/sys/kernel/ostype
Linux

$ cat /proc/sys/kernel/osrelease
4.19.57-v71+

$ cat /etc/os-release
PRETTY_NAME="Raspbian GNU/Linux 10 (buster)"
NAME="Raspbian GNU/Linux"
VERSION_ID="10"
VERSION="10 (buster)"
VERSION_CODENAME=buster
ID=raspbian
ID_LIKE=debian
HOME_URL="http://www.raspbian.org/"
SUPPORT_URL="http://www.raspbian.org/RaspbianForums"
BUG_REPORT_URL="http://www.raspbian.org/RaspbianBugs"
```

***Memory use***

The command `cat /proc/meminfo` can be used to display information about memory usage

```
# cat /proc/meminfo
MemTotal:        1962580 kB
MemFree:         1536800 kB
MemAvailable:    1824020 kB
Buffers:          265776 kB
Cached:           100680 kB
SwapCached:            0 kB
Active:            94596 kB
Inactive:         287904 kB
Active(anon):      18496 kB
Inactive(anon):     8332 kB
Active(file):      76100 kB
Inactive(file):   279572 kB
Unevictable:          16 kB
Mlocked:              16 kB
HighTotal:       1284096 kB
HighFree:        1159104 kB
LowTotal:         678484 kB
LowFree:          377696 kB
SwapTotal:        102396 kB
SwapFree:         102396 kB
Dirty:                 0 kB
Writeback:             0 kB
AnonPages:         16100 kB
Mapped:            29044 kB
Shmem:             10788 kB
KReclaimable:      12052 kB
Slab:              23484 kB
SReclaimable:      12052 kB
SUnreclaim:        11432 kB
KernelStack:         888 kB
PageTables:          936 kB
NFS_Unstable:          0 kB
Bounce:                0 kB
WritebackTmp:          0 kB
CommitLimit:     1083684 kB
Committed_AS:      85948 kB
VmallocTotal:     245760 kB
VmallocUsed:        3812 kB
VmallocChunk:          0 kB
Percpu:              416 kB
CmaTotal:         262144 kB
CmaFree:          256544 kB
```

***USB devices***

A list of the connected USB devices can be displayed by using the command `lsusb -t`

```
# lsusb -t
/:  Bus 02.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/4p, 5000M
/:  Bus 01.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/1p, 480M
    |__ Port 1: Dev 2, If 0, Class=Hub, Driver=hub/4p, 480M
```

***SD card information***

The SD card where the operating system is installed on has the device name `/dev/mmc-blk0`. We can display information about this SD card using the command `fdisk -l /dev/mmcblk0`.

```
# sudo fdisk -l /dev/mmcblk0
Disk /dev/mmcblk0: 14.9 GiB, 15962472448 bytes, 31176704 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: 0x96ea1f4a

Device         Boot  Start      End  Sectors  Size Id Type
/dev/mmcblk0p1        8192   532479   524288  256M  c W95 FAT32 (LBA)
/dev/mmcblk0p2      532480 31176703 30644224 14.6G 83 Linux
```

**Resource monitoring**

System resource monitoring is an important topic, especially in real-time multiprocessing applications where it may be required to monitor and control the processes running in the system.

Perhaps one of the most important system monitoring commands is `top`, which displays current system processes, how much memory they use, CPU time consumed, process priorities, and much more.

```
$ top

top - 16:17:46 up 15:52,  1 user,  load average: 0.21, 0.21, 0.18
Tasks: 102 total,   1 running, 101 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.6 sy,  0.0 ni, 99.3 id,  0.1 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   1916.6 total,   1498.7 free,     46.6 used,    371.3 buff/cache
MiB Swap:    100.0 total,    100.0 free,      0.0 used.   1780.8 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  113 root      20   0       0      0      0 D   2.0   0.0  72:21.26 w1_bus_master1
 4268 root      20   0   10340   2924   2464 R   0.3   0.1   0:00.09 top
    1 root      20   0   15260   7796   6316 S   0.0   0.4   0:04.19 systemd
    2 root      20   0       0      0      0 S   0.0   0.0   0:00.06 kthreadd
    3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
    8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
    9 root      20   0       0      0      0 S   0.0   0.0   0:00.39 ksoftirqd/0
   10 root      20   0       0      0      0 I   0.0   0.0   0:08.37 rcu_sched
   11 root      rt   0       0      0      0 S   0.0   0.0   0:00.01 migration/0
   12 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
   13 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
   14 root      rt   0       0      0      0 S   0.0   0.0   0:00.01 migration/1
   15 root      20   0       0      0      0 S   0.0   0.0   0:00.10 ksoftirqd/1
   17 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/1:0H-kblockd
   18 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/2
   19 root      rt   0       0      0      0 S   0.0   0.0   0:00.01 migration/2
   ...
```

The display is dynamic and is updated continuously. It consists of the following sections.

- <u>Uptime and load average</u>
  The following information is given in the first line:

  - Current time (16:17:46)
  - System uptime (15:52 hours)
  - Number of users (1)
  - Average Load measured over 1-, 5-, 15-minutes (0.21, 0.21, 0.18)

  The load average is the average number of processes in the run queue, i.e. processes that are using the CPU or waiting for their turn.

- <u>Tasks</u>
  The second line lists the tasks the system is currently performing. The following information is given:

  - The total number of processes in the system (102)
  - The number of actively running processes (1)
  - The number of processes that are sleeping in the background (101)
  - The number of stopped processes (0)
  - The number of zombie processes (0)

  Zombie processes are the ones that have finished executing but are still resident in system memory. Such processes can cause problems.

- <u>CPU utilization</u>
  The third line is the CPU utilization percentage values. The following information is given:

  - us - for user applications (0.0%)
  - sy - for system applications (0.6%)
  - ni - for processes that have been niced (0.0%)
  - id - for idle processes (99.3%)
  - wa - for processes waiting for I/O completion (0.1%)
  - hi - for processes waiting for hardware interrupts (0.0%)
  - si - for processes waiting for software interrupts (0.0%)
  - st - for hypervisor processes (0.0%)

  It is important to notice high values of wa, hi, si, or st. They are not a good sign as high values indicate the system is waiting on hardware or software to finish processing.

- <u>Memory usage</u>
  Lines 4 and 5 represent memory usage of the system, where line 4 displays actual memory usage. Line 5 displays the swap space usage.

  The following information is given in line 4:

  - Total memory (1916.6MB)
  - Free memory (1498.7MB)
  - Used memory (46.6MB)
  - Memory allocated to buffers/cache (371.3MB)

  The following information is given in line 5:

  - Total swap space (100MB)
  - Free swap space (100MB)
  - Used swap space (0MB)
  - Available swap space (1780.8MB)

- <u>Process table</u>
  The process table gives the following information for all the processes running in the system. The following information is given for each process:

  - PID : the process ID number
  - USER : the owner of the process
  - PR : priority of the process
  - NI : the nice value of the process
  - VIRT : the amount of virtual memory used by the process
  - RES : the size of the resident memory
  - SHR : shared memory the process is using
  - S : process status (sleeping, running, zombie)
  - %CPU : the percentage of CPU consumed
  - %MEM : percentage of RAM used
  - TIME+ : total CPU time the task used
  - COMMAND : The actual name of the command

Notice that the `htop` command gives a display similar to the top command but it is more user friendly with colours and dynamic load bars.

The process ID number is a unique number identifying a process. The PR and NI fields represent the priority of a process. NI displays the nice level which is a static priority of a process when it is initialized. By default, when a process is initialised it has a nice value of 0, but can be changed using the nice command. NI has 40 values from -20 to +19. -20 is the highest priority, and +19 the lowest priority. PR shows the process dynamic priority which is calculated based on the nice value and when a process starts, the PR value equals NI value plus 20. Usually, processes start with NI value of 0 and PR value of 20. During the lifetime of a process, PR can change depending upon the system.
The priority of a running process can be changed using the `renice` command. As an example, assume that process `rcu_sched` has `ID = 10` and NI priority of 0. The priority of this process can be changed to 1 as follows:

```
$ sudo renice 1 –p 10
10 (process ID) old priority 0, new priority 1
```

**The ps command**
The nice value of a process can be displayed using the `top` or `ps` command. For example, the PID and nice numbers of process 11 (`rcu_sched`) can be displayed as follows:

```
$ ps –o pid,nice –p 10
PID NI
10 1
```

The `ps` command can also be used to list all processes used by the current user.

```
$ ps
PID TYY TIME CMD
982 pts/0 00:00:00 bash
1172 pts/0 00:00:00 ps
```

The command `ps -ef` gives more information about the processes running on the system,

```
# ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 00:25 ?        00:00:04 /sbin/init
root         2     0  0 00:25 ?        00:00:00 [kthreadd]
root         3     2  0 00:25 ?        00:00:00 [rcu_gp]
root         4     2  0 00:25 ?        00:00:00 [rcu_par_gp]
root         8     2  0 00:25 ?        00:00:00 [mm_percpu_wq]
root         9     2  0 00:25 ?        00:00:00 [ksoftirqd/0]
root        10     2  0 00:25 ?        00:00:08 [rcu_sched]
root        11     2  0 00:25 ?        00:00:00 [migration/0]
root        12     2  0 00:25 ?        00:00:00 [cpuhp/0]
root        13     2  0 00:25 ?        00:00:00 [cpuhp/1]
root        14     2  0 00:25 ?        00:00:00 [migration/1]
root        15     2  0 00:25 ?        00:00:00 [ksoftirqd/1]
root        17     2  0 00:25 ?        00:00:00 [kworker/1:0H-kblockd]
root        18     2  0 00:25 ?        00:00:00 [cpuhp/2]
root        19     2  0 00:25 ?        00:00:00 [migration/2]
root        20     2  0 00:25 ?        00:00:00 [ksoftirqd/2]
root        22     2  0 00:25 ?        00:00:00 [kworker/2:0H-kblockd]
root        23     2  0 00:25 ?        00:00:00 [cpuhp/3]
root        24     2  0 00:25 ?        00:00:00 [migration/3]
root        25     2  0 00:25 ?        00:00:00 [ksoftirqd/3]
root        28     2  0 00:25 ?        00:00:00 [kdevtmpfs]
root        29     2  0 00:25 ?        00:00:00 [netns]
...
```

The PID of a process can be displayed using command pidof, followed by the name of the process. An example is shown below where the PID of process rt is displayed:

```
$ pidof  rcu_sched
10
```

To display the name of a process given its PID, use the ps command with option –q as shown in the following example:

```
$ ps –q 10
PID TTY TIME CMD
10 ? 00:00:00 rcu_bh
```

Sometimes we may want to know if a process is running in the system. This can be checked using the ps command with option –C, followed by the process name. For example, to check if process rcu_bh is running, enter the following command:

```
$ ps –C  rcu_sched
PID TTY TIME CMD
10 ? 00:00:00  rcu_sched
```

To display the processes started by a specific user (e.g. pi), enter the following command:

```
$ ps –u pi
```

**Killing a process**

There are many options for killing (or stopping) a process. A process can easily be killed by specifying its PID and using the following command:

```
$ kill <PID>
```

Sometime you may need to use the `-9` option to force all related commands to stop. i.e.

```
$ kill -9 <PID>
```

You could also use the `killall` command to stop all occurrences of a process. This command is not a clean stop as it will immediately stop the specified process:

```
$ killall <process name>
```

You may need to specify `sudo` before using `kill` commands.

**Disk usage**
The disk free command `df` with option `-h` can be used to display disk usage statistics.

```
# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        15G  1.5G   13G  11% /
devtmpfs        830M     0  830M   0% /dev
tmpfs           959M     0  959M   0% /dev/shm
tmpfs           100M  8.4M   92M   9% /run
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           959M     0  959M   0% /sys/fs/cgroup
tmpfs           100M     0  100M   0% /tmp
tmpfs            10M     0   10M   0% /var/tmp
tmpfs           100M  2.3M   98M   3% /var/log
/dev/mmcblk0p1  253M   54M  199M  22% /boot
tmpfs           192M     0  192M   0% /run/user/0
```

**CPU temperature**
A Raspberry Pi can be overclocked and is generally passively cooled which can result in high temperatures under load. When temperatures get too high the firmware should throttle back to prevent damage.To view and monitor the CPU temperature, in human readable format, run the command

```
# vcgencmd measure_temp
temp=43.0'C
```

To view the clock frequency at that moment

```
$ vcgencmd measure_clock core
frequency(1)=250000000
```

**Power**
If your Pi does not get the juice it needs, it can run into all sorts of problems. We also have a sub-command which reports throttling.

```
$ vcgencmd get_throttled
throttled=0x50000
```

The bits in this number represent

```
0x50000 =
01010000000000000000
||||            ||||_ Under-voltage detected now
||||            |||_ Arm frequency capped now
||||            ||_ Currently throttled
||||            |_ Soft temperature limit active
||||_ Under-voltage has occurred since last reboot
|||_ Arm frequency capped has occurred since last reboot
||_ Throttling has occurred since last reboot
|_ Soft temperature limit has occurred since last reboot

=> Under-voltage has occurred since last reboot & Throttling has occurred since
last reboot

0x50005 =
01010000000000000101
||||            ||||_ Under-voltage detected now
||||            |||_ Arm frequency capped now
||||            ||_ Currently throttled
||||            |_ Soft temperature limit active
||||_ Under-voltage has occurred since last reboot
|||_ Arm frequency capped has occurred since last reboot
||_ Throttling has occurred since last reboot
|_ Soft temperature limit has occurred since last reboot

=> Currently Under-voltage detected & throttled as well as Under-voltage has
occurred since last reboot & Throttling has occurred since last reboot
```

**Monitoring 24/7**

All of above gives you ad-hoc information and you have to keep watching what is going on. If you use a monitoring tool you can captures that information on a time line and analyse it afterwards. I refer to my documents over SNMP, MRTG and RRD as well as Telegraf, InfluxDB and Grafana.