Welcome Lab Session 0

Getting started

Contents

These notes are available online at
`studentnet.cs.manchester.ac.uk/ugt/COMP10120/labscripts/intro0.pdf`
You may find it useful to use the online version so that you can follow web links.

0.1 Welcome

Hello and welcome. In this first, introductory, lab we're going to cover some of the basic things you'll need to know about the IT infrastructure here in the School of Computer Science, and get your Raspberry Pi set up so that you can get going quickly in the next lab. Some of what we tell you may well seem very obvious to you, and if that's the case we ask you to be patient. Some other things might not be so obvious...

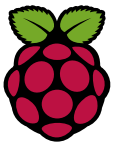
In this and every lab there will be staff, consisting of academic staff and postgraduate students, around to help you. If you're stuck or find something that you really can't understand, then *please ask for help*; that's what the lab staff are here for, don't just sit there getting frustrated. The postgraduate students are known as **Teaching Assistants**, or **TAs** for short, although up to 2014 they were referred to as **Demonstrators**. We will try to use the new name, but inevitably both will be used for a while.

0.2 About these notes

0.2.1 Breakout boxes

Scattered throughout the main text of these exercises there are information boxes of various kinds:

Danger! The bomb icon explains problems and pitfalls, and how to avoid them. It's really important that you read these sections, or you may regret it later.



Raspberry Pi Facts and Factoids. These sections explain useful but non-essential things about the Raspberry Pi. If you're pushed for time, you can safely skip these boxes and perhaps come back to them another time.

We digress... Boxes marked with this icon contain digressions and facts that are hopefully interesting but are probably tangential to the main flow of the exercise.

Stop here... Boxes marked with this icon contain checkpoint activities that you should complete before proceeding further.

0.2.2 Styles and conventions

We'll be using various typographic styles to indicate different things throughout these notes. When we introduce new concepts, or use terms that have a different meaning in computer science to their everyday use, they will appear **like this**, and will be explained in the nearby text. Things that are either the input or output to commands will appear `in this font`. And many words or phrases will have a little 'w' after them, meaning that if you click on the term (if you're reading the notes online) or type the phrase into **Wikipedia**[w] (if you're reading this on paper), you should be taken to a sensible definition in case you want to learn more about something that we've touched on here.

Where you see text in square brackets [LIKE THIS] it will mean that when you're typing things you should replace the text with something that the text describes; for example [NAME OF THIS CITY], would become Manchester.

0.3 The CS labs setup

All the desktop PCs in the labs in the Kilburn building are 'dual-boot': they can be started up running either Linux or Windows. This is for flexibility – the labs for most course units use Linux, but some use Windows, and of course, outside the labs, you're free to use whichever you prefer for any aspect of your studies that does not require use of a particular operating system. If you're not familiar with Linux, don't worry. We'll be telling you a bit about it in this lab, and you'll be looking at it in much more detail in subsequent labs.

Figure 1
The Windows7 Welcome screen.

0.4 Using Windows

We'll start with Windows, so the first thing we need to do is to make sure your PC is running Windows. Have a look. If it is, and the screen looks like Figure 1, then you will not need to reboot, but please read the next bit anyway, because at some point you may need to reboot from Linux to Windows.

If your PC is currently in Linux, showing a black screen with a login prompt, you need to reboot the PC. To do this, press `ctrl`, `alt` and `delete`.

This will probably cause strange messages to appear on the screen, disappear, and be replaced by yet more strange messages. Now look at Figure 2: when you see something like that on the screen press `space` you will need to respond. Meanwhile, be patient, watch it all happen, and don't worry what it all means.

After a while, everything should settle down and the screen will look like Figure 2; press `space` now!

This is where you decide whether to start up Windows or Linux. *If you do nothing when this screen appears, the computer automatically boots into the default operating system after a fixed timeout period. By pressing the space bar (any other key would have done) you have stopped this timeout process.*

Now use the up/down arrow keys on the keyboard to highlight the line reading `Windows 7`, and press the `enter` key. After a short while, the Windows 7 welcome screen will appear, as shown in Figure 1. Now use `ctrl-alt-del` again as instructed and you should see the Windows 7 login screen, as shown in Figure 3.

To log in to your personal account, type in your username (this is an 8-character name usually starting with an 'm' that you were given at registration). Your password will be the one you

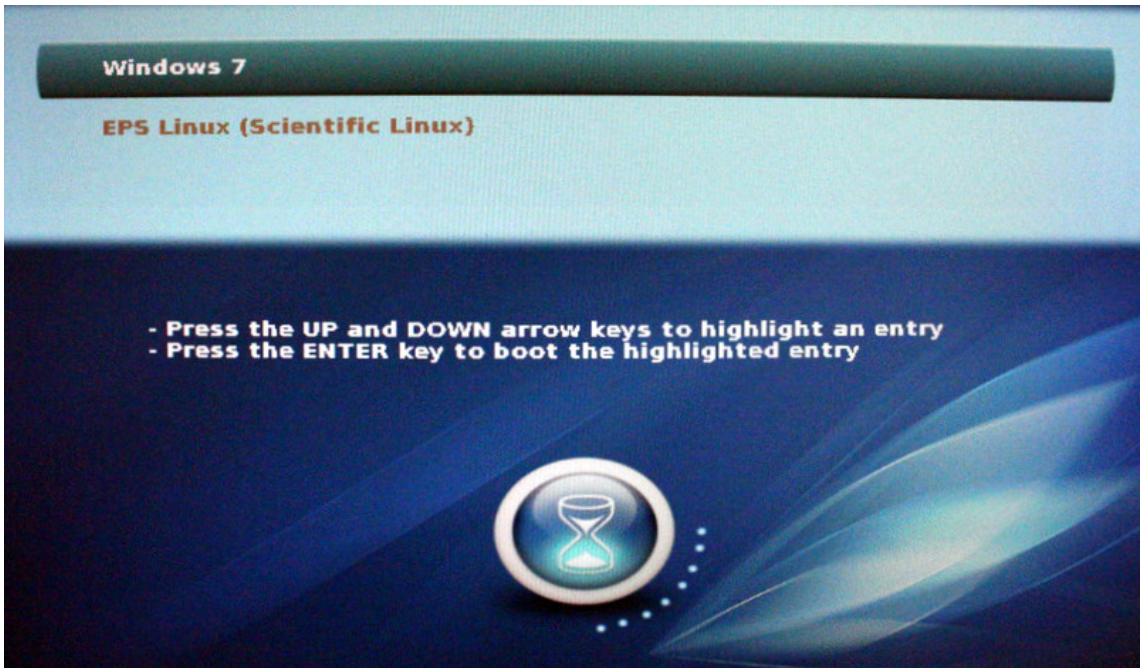


Figure 2
The boot selection menu screen.

Figure 3
The Windows 7 login screen.

set at registration. If your password doesn't work, or if you've forgotten it, you need to fix this urgently. You can login on a machine running Windows, with the username `register` and password `register`, then follow the instructions.

If you forget your password later on, you can also use one of the following methods:

- If you have access to a web browser, use the password recovery page at `https://iam.manchester.ac.uk/recover`
- You can also contact the University IT helpdesk (opening hours are Monday to Friday 9am to 5pm) by phone on 0161 306 5544, or dial 65544 from a University internal phone; or you can visit a helpdesk either on the ground floor of the Kilburn Building, in the Main Library (Building 55 on the Campus Map), or in the Alan Gilbert Learning Commons (building 63).
- Remember, if you need help, ask!

Once you're logged in, go to your My Manchester page in a web browser, at

`https://my.manchester.ac.uk`

You will have to authenticate again to access this page, which should look something like Figure 4.

0.5 Reading your email

We use email extensively in the School, so it's vitally important that you read your mail regularly—we work on the assumption that you'll be checking your School email at least once a day (and probably much more often). On the My Manchester page, follow the mail link (indicated by the hand drawn arrow in Figure 4) to access your email on the University's Office365 system. This is a system that gives you 25GB of email storage space and an integrated calendar. You should see a page looking something like Figure 5.

Have a look around for a few minutes and check what mail you've got. In particular, look for one with 'The Monday Mail' in the Subject line. This is an important mail you'll receive every Monday (hence the name) throughout your 3 or 4 years with us in the School which tells you what's going on each week. Take a moment to read it now. You can always read the Monday Mail, by the way, at the archive located at
`studentnet.cs.manchester.ac.uk/ugt/mondaymail/`

Office365, like most modern email systems, supports the IMAP protocol – which means that you can access your mail from any device that can run an IMAP mail client. Some examples of such clients are: Mozilla Thunderbird, Mac Mail, Windows Live Mail and mail apps on mobile devices. No matter what client you use, you need to tell it the appropriate settings. The mechanism for finding these setting can be found in our CS student IT wiki, which is located on the web at `wiki.cs.manchester.ac.uk/index.php/StudentFAQ/IT`. Look for the answer to the question 'How do I configure my favourite IMAP mail client?'. This wiki provides help about the IT infrastructure within the School; it is one of several School FAQ lists, which can be found at `wiki.cs.manchester.ac.uk/index.php/StudentFAQ`. Please make use of these valuable sources of information.

Once you've found the settings, use Office365 to email this information to yourself, to your University account, which is usually of the form:

`firstname.lastname@student.manchester.ac.uk`

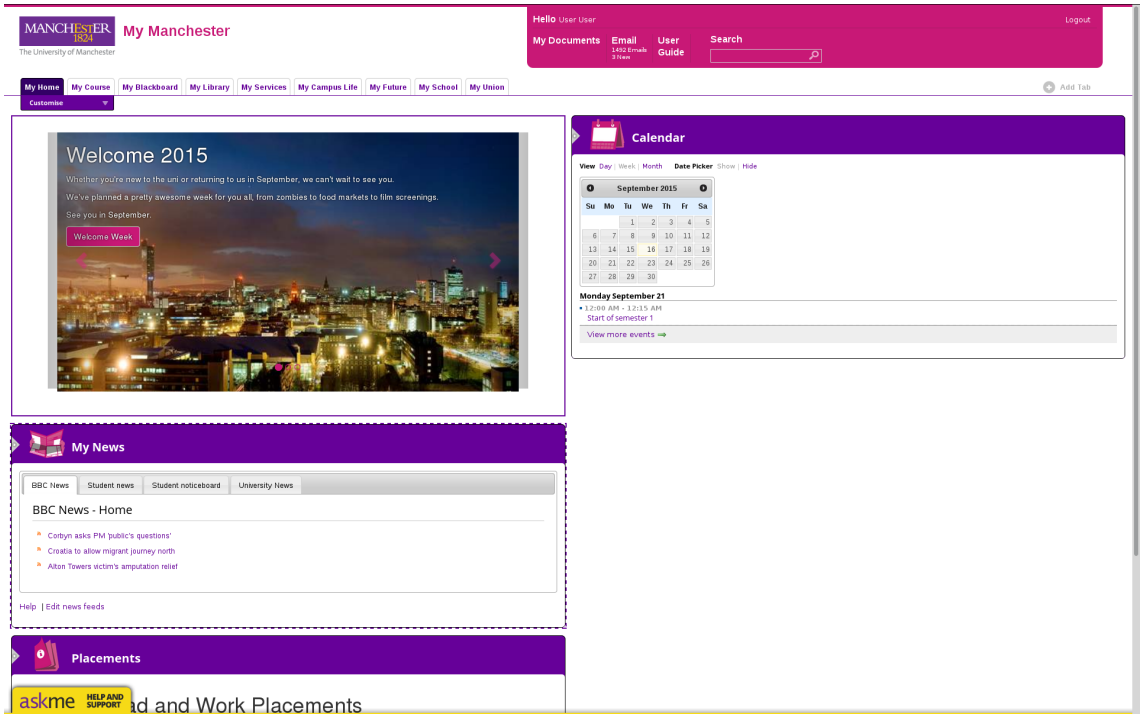


Figure 4
Your MyManchester page.

You should also write this information down somewhere you can find it, as you will need it in a later lab when you may not have access to your email.

You'll need this information later so please don't delete this email after you've read it.

Finally, if you use an IMAP mail client on your phone or mobile device, you can configure it to use the settings you've just found, and check that you can read and send email successfully. It's probably best to try this later as the mobile signal in the lab is not good.

That's all on Windows for now, but of course you're free to boot an available PC into Windows and use it at any time unless you are in a lab that requires the use of Linux.

0.6 Rebooting into Linux

So let's reboot the PC, and start it up in Linux. First, log out of Windows by selecting the Windows icon in the bottom left of the screen and then `Log off`. Get back to the login screen, click the small icon in the lower right of the screen (see Figure 6) and select `Restart`.

The system will shut down and after a while we'll be back to the boot selection menu screen we saw earlier in Figure 2. This time, press space immediately, then use the up/down arrow keys to select `EPS Linux (Scientific Linux)`. Linux will now start, and after a short while you should see a black screen, with a white login prompt.

We won't login to Linux at this stage, that can wait until a later lab, next week. We would, however, like you to read the rest of this document, which gives you some useful and interesting background information about Linux. If you don't have time to finish reading it in the lab, please make sure you do so before the next lab session.

Before we move on to this reading we would like you to make a start in setting up your very

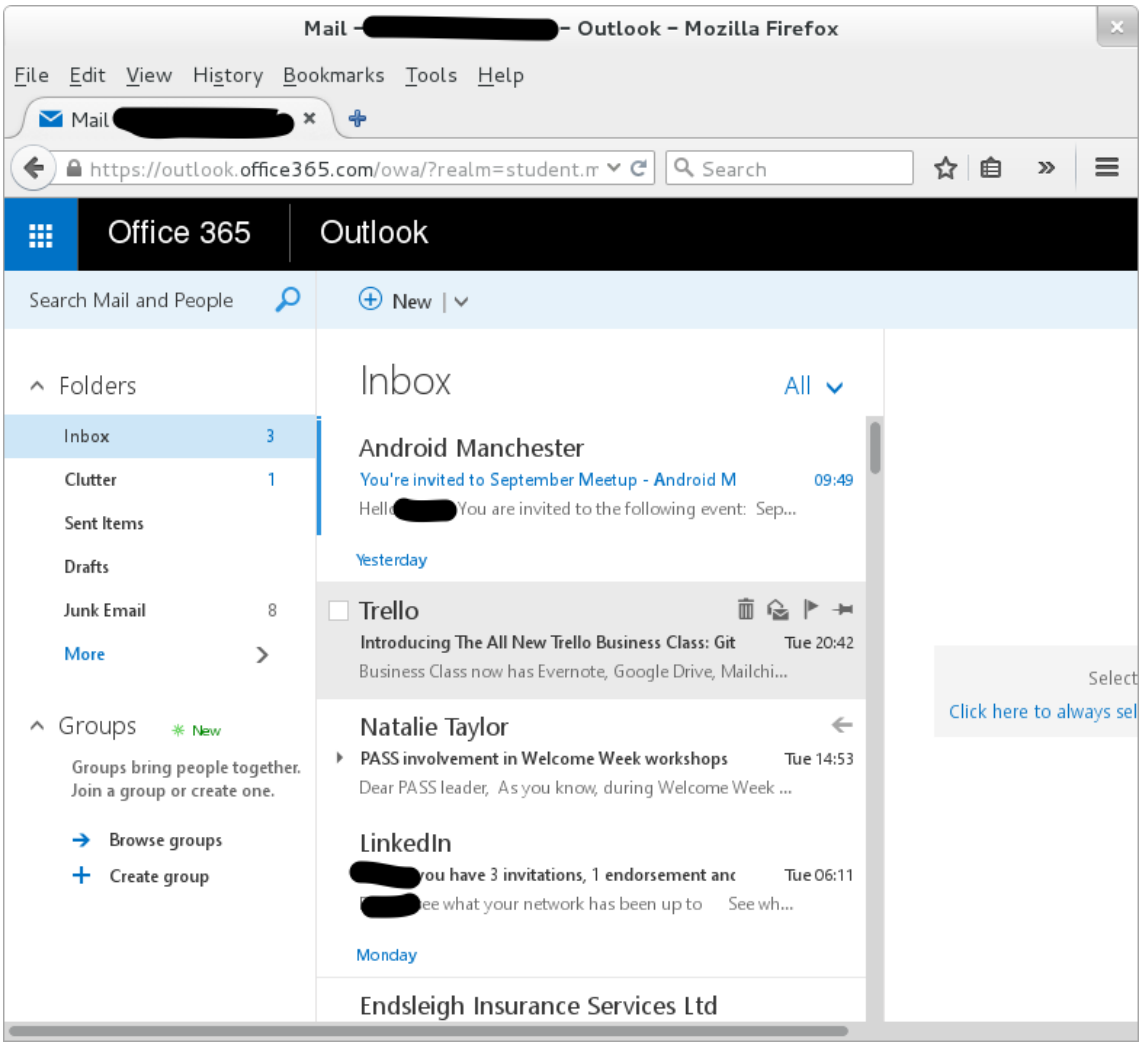


Figure 5
Your Office365 email.

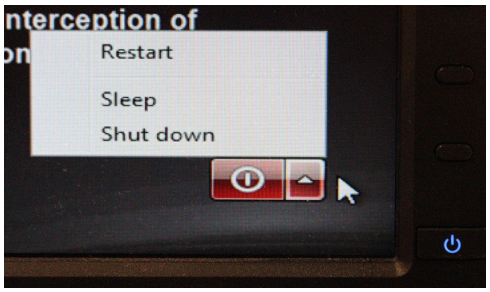


Figure 6
Restarting the PC.

own Raspberry Pi computer. During this process there will be a period where you have to sit and wait for something to happen; this will give you a good opportunity to read the background information about Linux.

0.7 Getting started with Raspberry Pi

You should by now have received a bag containing all the bits you'll need to assemble your Raspberry Pi. The Raspberry Pi is an astonishing piece of hardware. Not because it is super-fast or super-powerful—it's actually quite a slow computer by today's standards—but because it is small, cheap and needs very little energy to run. Its cheapness means you can experiment with it safe in the knowledge that if you mess it up, lose it, or drop it into the canal, getting hold of a replacement isn't going to cost you much more than a text-book or a night out at the cinema. Its small size and fairly modest power requirements mean that it can be put to use in lots of applications where a regular-sized PC would be impractical. We hope this will encourage you to experiment and explore, and to take risks playing with both its hardware and software that you might be reluctant to do on your own computer or tablet, or simply can't do with the School's lab machines.

In this lab session we start the process of setting up the Raspberry Pi and continue in the next one to use it and get you familiar with the Pi, and with some of the basics of the Linux operating system it uses. We're going to be covering a lot of ground quite quickly, and it's important that you read these notes carefully and follow the instructions precisely for now. As the lab sessions progress, the instructions will become much less prescriptive, and we'll be encouraging you to experiment and explore much more, and to find out things for yourself. But for now, just follow our lead.

The most remarkable thing about the Pi is that, although it's not the most powerful of computers, it is capable of running the same full **Linux**[w] operating system as the machines that you'll be using in the labs for the duration of your studies, and which you'll undoubtedly encounter in your future careers. We're actually going to be using slightly different flavours of Linux on the Pi and the lab machines, but the differences are fairly minor—more on that later. Let's get started.

The Raspberry Pi itself is just the circuit board shown from the top in Figure 7. Your kit contains a Pi together with a case, SD card and power supply. If you want to replace the case with one in a different style you're welcome to do so (there are plenty available to buy online, and lots of people have made their own unique ones just for the fun of it). If you do put it in your own case, please remember to transfer the 'sequence number' sticker we've placed on the original box to the new box so that if the Pi is left behind in the lab it can be returned to you (this number is also labelled on the ethernet port). The Pi is reasonably robust, and you can use it without a case, but obviously it's a bit more vulnerable if it's not in a box of some sort. Figure 8 shows the Pi's circuit board from beneath, indicating where the micro SD card gets inserted.

0.7.1 Connecting the Pi

It's important that you connect the Pi up in exactly the order specified here—so even if you are familiar with using a Pi, please don't jump ahead and plug everything in at once (no harm will come to the Pi if you do, but this exercise relies on your following our instructions closely). Refer to Figure 9, and then connect your Pi up like this:

- The monitors in the LF31 Lab are all fitted with an extra video lead for connecting up the

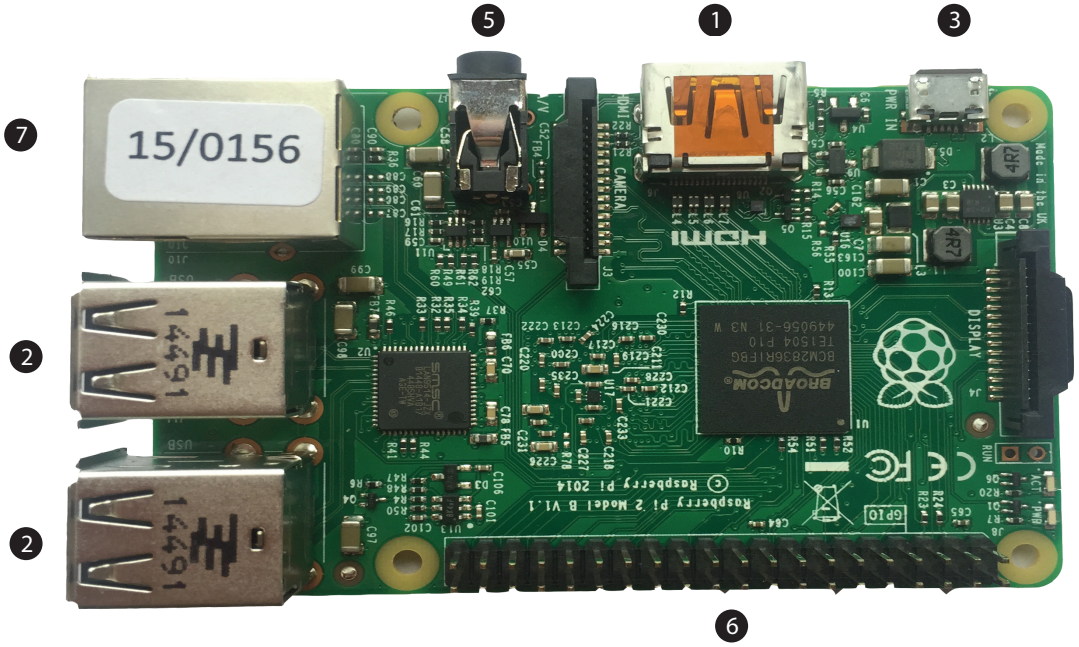


Figure 7
An uncased Raspberry Pi 2. The numbered connectors are ❶ HDMI output, ❷ four USB ports, ❹ micro-USB power input, ❺ a red power LED and a green 'activity' LED, ❻ stereo audio out, ❼ General Purpose Input/Output (GPIO), ❽ Ethernet.

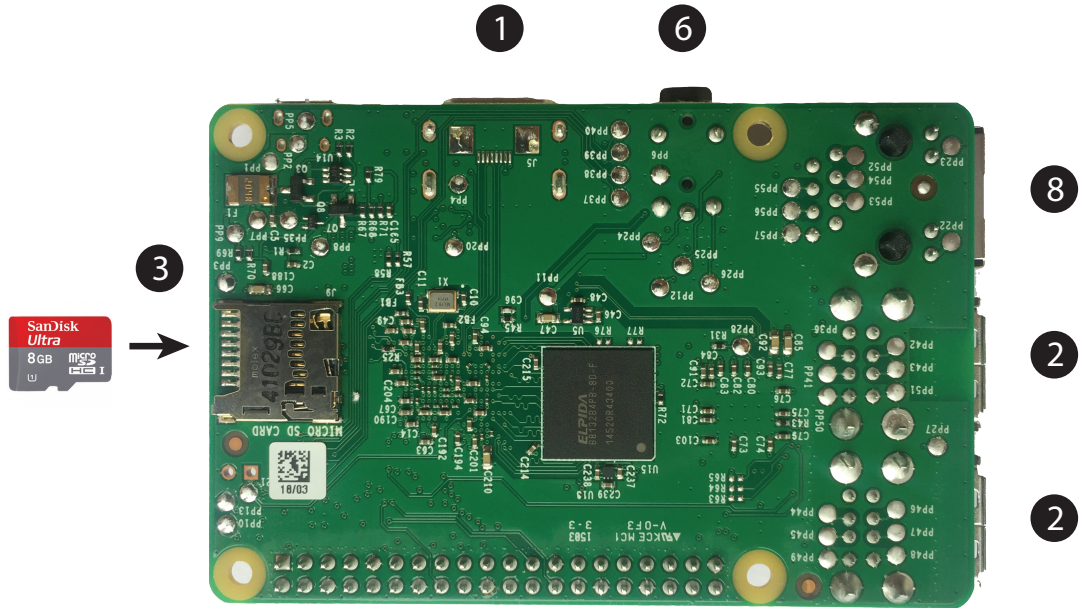
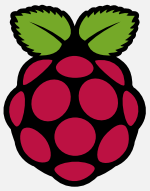


Figure 8
A naked Raspberry Pi from below. The ports are numbered as in Figure 7, and in addition ❸ shows the location of the micro SD (Secure Digital) memory card and the correct orientation of the card to insert into the Pi.

Breakout 1: Why is it called a Pi?



The Raspberry Pi apparently got its name because (a) lots of other computer systems have been named after fruit (you'll know of Apple and Blackberry, but in the past there has also been at least Apricot and Tangerine), and (b) the **Python language**[W] was one of the first things ported to run on it. The logo was created by Paul Beech, and is based on **Buckminsterfullerene**[W], a spherical fullerene molecule more commonly called a Buckyball. Its designer pointed out that a full buckyball has 32 faces, but that only 11 are visible in the 2D logo; and that the Raspberry Pi has a 32-bit processor and an ARM11 on board.

The ARM processor, on which the Pi and the vast majority of the world's other mobile devices are based, was originally designed by a team led by **Steve Furber**[W], who is a Professor in this School.

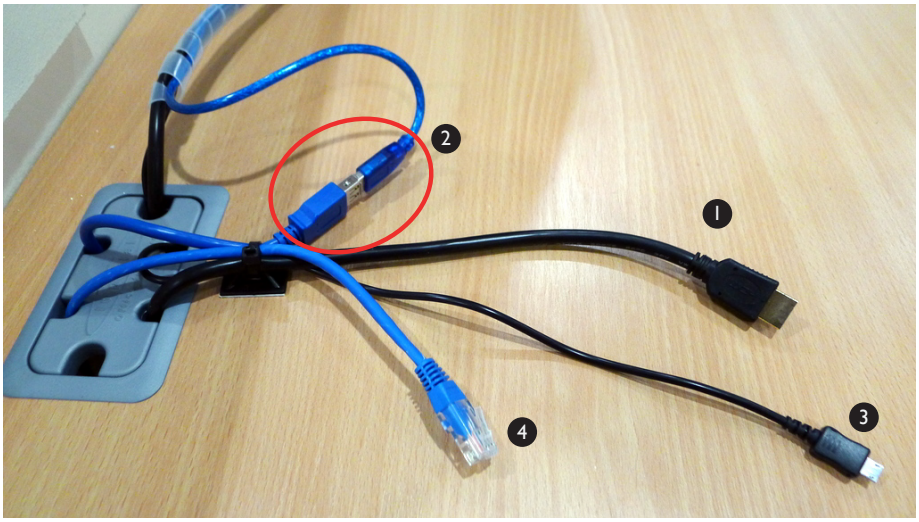


Figure 9
The following cables are available to connect up your Pi. ❶ HDMI, ❷ Keyboard and Mouse, ❸ 5 Volt Micro USB power supply, and ❹ network.

Pis. We try to standardise on blue cables for connecting to a Pi. Locate the HDMI lead (labelled ❶ in Figure 9), and plug this into the socket marked ❶ on Figure 7.

- Next we'll need to connect a keyboard and mouse. Separate the male and female USB connectors marked ❷ on Figure 9, and plug the male end into one of the two USB sockets on your Pi (marked with ❷ on the Figure 7). It doesn't matter which USB socket you choose, but please make sure to reconnect this to the PC when you're done with these experiments as a courtesy to the next user.
- **Do not connect the power or network connectors at this stage!**

Next, we need to insert the memory card that contains the Pi's operating system, and on which you'll be storing your own files (note although this is a 'memory card', for the Raspberry Pi it's behaving like the 'hard disk' of a regular computer, not like the *memory* of a regular computer).

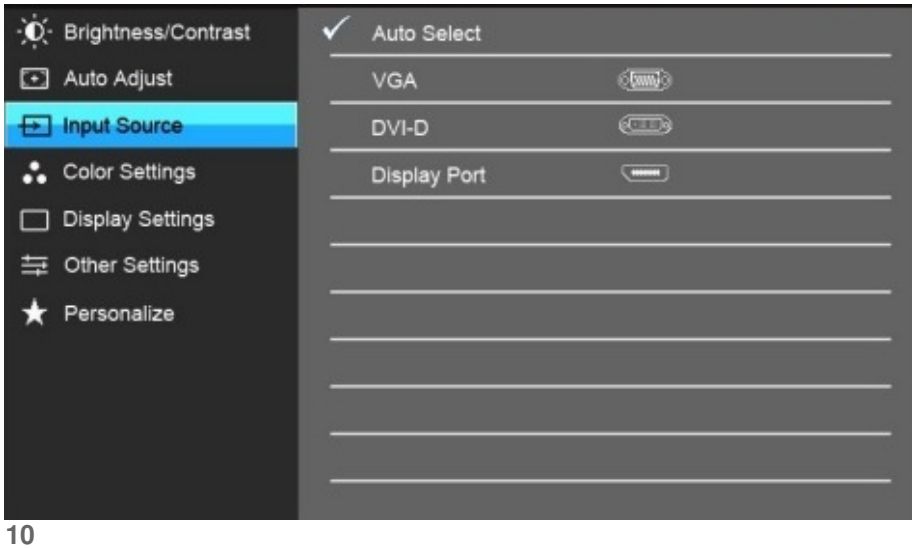


Figure 10
The Input Selection menu on the monitor. This image of the Dell U2412M Monitor has been reproduced with the kind permission of Dell. All rights reserved.

The card you've been given comes in a little caddy that allows it to be used as a full-sized Secure Digital card, but for the Raspberry Pi Model 2 we just need the little mico SD card it contains. Carefully pop that out of the caddy and then refer to Figure 8 to see where it goes on the Pi's board. It should slide both in and out smoothly. Don't force the card in or you might damage it or the Pi – if you're not sure, get some help.

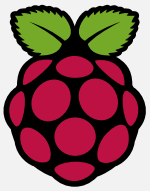
Now you're ready to power up the Pi. In our lab setup in LF31 the power to the Pi is supplied via the monitor, so it is important that the monitor is not in standby mode. Standby is indicated by an orange light on the on/off button at the bottom right of the screen, rather than blue if it is on. If the light is orange, press it to turn the monitor off, then again to switch it back on and turn it blue. Now locate the 5v Micro USB cable (the same connector that you'll find on many modern smartphones and tablet devices); this is labelled ❸ on Figure 9. This goes into the socket marked ❸ on Figure 7. There is no power switch on the Pi, so as soon as the power cable goes in, it will start to boot (this strange term is explained in breakout box 3): the red PWR LED should light up and stay on, and you'll also notice that the OK LED (which indicates SD-card activity) to its left also flickers. If any of the other three LEDs marked FDX, LNK, or 10M are lit, then that means you've already plugged a network cable into your Pi, in which case please unplug that now!

Now we need to switch the input on the monitor from DisplayPort (which is the input used by the PC under the desk) to DVI (the cable you've connected your Pi to is a HDMI to DVI cable). On the lower-right side of the monitor you will see 4 buttons – see Figure 10. Press any of them to bring up the monitor's menu. Now press the 3rd button down and navigate to the input selection menu. Switch the Input to DVI and all being well you should see a black screen containing the Raspberry Pi logo at the top left, and white text that scrolls up the screen as the Pi boots.

The latest generation of the Pi automatically starts a graphical user interface; as we need to be more flexible, we will issue some commands to disable this behaviour. Once this is done your Pi will subsequently start from a simple command line prompt.

On the screen you will see a panel of tool icons on the top left. Click on the icon that looks like a monitor screen. This runs a command line terminal that you can enter commands in. Don't worry at this stage exactly what the commands mean. Type:

Breakout 2: Raspbian



Raspbian is a version of the Debian release of Linux, tuned for the Pi. If you manage to corrupt the operating system, or just want to start from scratch, then re-writing the SD card with a fresh image is reasonably straightforward: simply reboot your Pi and press the Shift key when the initial splash screen loads (you will need to be quick, as it only appears for a few seconds). This will present you with a list of options. Ensure that Raspbian is selected and then click the Install icon to re-write the SD card with a fresh image. This will erase all of the data on the Pi, so it's crucial that you make sure that you have a backup of anything important.

If you buy a new SD card, instructions on how to get hold of the files you'll need, and how to write them to the SD card on various operating systems are available at `www.raspberrypi.org/downloads`. You might want to think about getting a larger SD card in any case; the one we've given you is fine for the labwork we've set, but probably a bit on the small side for anything else. SD cards are widely available in shops and online, and aren't expensive. But you should check whether the specific card you're going to buy is compatible with the Pi before parting with any money—differences in the read/write speeds of some cards mean they don't play nicely with the Pi.

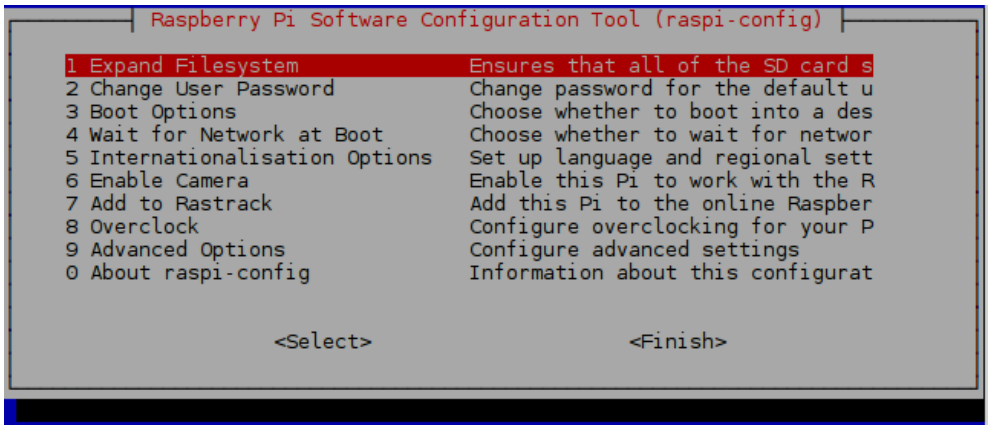


Figure 11
The Raspberry Pi config tool. The highlight bar can be moved between the different controls using the Tab key, or up and down in the menu using the arrow keys.

```
sudo raspi-config
```

followed by the Return key.

You should see a configuration screen as shown in Figure 11. Use the arrow keys to select 'Boot Options', then press Tab until `<Select>` is highlighted, then press Return. This will bring up another configuration screen. This time use the arrow keys to select the option 'Text console, requiring user to login'. Again use Tab until `<OK>` is highlighted, then press Return. You should now be back at the previous configuration screen where you should Tab to select `<Finish>` then press Return. You should now see a screen asking 'Would you like to reboot?'. Tab to `<No>` then press return.

You should now be back in the command line Terminal, where you can stop the Pi by typing

```
sudo halt
```

Breakout 3: Booting

The phrase 'booting' to refer to the process of starting up some computer system has become quite commonplace, but its origins are rather strange. It's thought to have first been used in early 19th Century America as a way of describing an obviously impossible action such as to "pull oneself over a fence by one's bootstraps". These days it is used to refer to any self-sustaining process that is able to happen without external help.

So why is starting a computer a bootstrapping process? In order for you as a user to be able to run a program, the computer needs an operating system. But in order to load its operating system, it needs some instructions that tell it how to understand the file system. And in order to load the instructions that tell it how to understand the file system it needs to…well, you get the idea. In reality, most computers have a very small set of instructions hardwired into them that begin the process of loading a slightly more complex 'bootloader', which then begins the process of loading the OS kernel and any modules needed to interact with the hardware, and then starts loading services and features of increasing sophistication that rely on the simpler ones loaded previously to function.

As an aside, you may want to consider this: if you need a text editor to write programs, how did the first text editor (which is itself a program) get written?

followed by the Return key.

We'll explain a bit more about what's going on here in the next lab session, but for now please just follow the instructions.

The Pi will now power itself down; the ACT LED will go off, the screen should go black, and only the red PWD LED will remain lit on the circuit board. At this point it's safe to pull the Micro-USB cable out of the Pi. Please make sure you reconnect the USB connector for use with the PC as a courtesy to the next user, *taking care to connect the USB cable the correct way round.* Make sure that you leave the desktop PC booted into Linux.

That's all we want you to do with the Raspberry Pi for this session, we will start using it in earnest in the next session. When you have completed the work for this session, please tell the lab supervisor so we know how you're getting on. Remember, if you don't finish it during the lab session, please try to do so before the next one.

Please make sure you bring your Pi and this set of notes to all the introductory lab sessions.

0.8 Unix and Linux

Over the next couple of weeks you will be undertaking a number of introductory labs to familiarise yourself with the School's computing infrastructure. Much of this is based on devices and machines running Linux, a variant of the Unix family of operating systems; this document provides some background on Unix and explains why we think it is important. It would very useful if you could read this before you attend the first introductory labs, where the emphasis will be on leading you through a series of tasks to explore our setup.

0.8.1 Operating Systems

An **operating system**[w] (OS) is a suite of software that makes computer hardware usable; it makes the 'raw computing power' of the hardware available to the user. You're probably most familiar with the **Microsoft Windows**[w] and Apple **OS X**[w] families of operating systems for 'desktop' computers, and **iOS**[w] (Apple, again) and Google's **Android**[w] for mobile devices; but many other more specialist operating systems exist, and you'll be studying some of these and the principles that underpin OS design in COMP25111 in your second year. In the meantime, a potted history of OS development will tide us over...

0.8.2 Unix Origins

In the late 1950s, an American company called **Bell Laboratories**[w] decided that they needed a system to improve the way they worked with their computer hardware (it's probably quite hard to imagine what interacting with a computer *without* an operating system might be; but it wasn't pretty and involved manually loading and running programs one by one). Together with the **General Electric Company**[w] and the **Massachusetts Institute of Technology**[w], they set about the design of an operating system they called **Multics**: the 'Multiplexed Information and Computing Service'. Multics was hugely innovative, and introduced many concepts that we now take for granted in modern operating systems such as the ability for more than one program to run 'at once'; but it did rather suffer from 'design by committee', and the final system was seen at the time as being overly complex and rather bloated ('bloated' is all a matter of perspective of course: it's sobering to realise though that the entire Multics operating system was only around 135Kb. Today's operating systems are something like 30,000 times this size...). In the late 1960s, a group of programmers at Bell Labs created a cut-down, leaner and cleaner version of Multics that would work on more modest hardware. Legend has it that this was to allow them to play their favourite (only!) computer game, **Space Travel**[w]. In an early example of the trend of giving things 'punny' names to contrast with the more clumsy Multics, they called this new system Unix! The so-called **Jargon File**[w] is a good source of explanations of various bits of computer slang and their obscure origins, and is well worth a read: in part to give some background history, but mostly as an insight into the minds of the computing pioneers of the past!

Even though Unix is now quite old, most Computer Scientists recognise that the designers of Unix got most of the fundamental concepts and architecture right. Given how much computing has changed since the 1960s, this was an astonishing intellectual achievement. Although Microsoft's **Windows**[w] is by far the most common operating system on *desktop* machines, the majority of the Internet, much of the world's corporate infrastructure, virtually all supercomputers, and even some mobile devices are powered by Unix-like operating systems. So, while the polished graphical user interfaces of Windows and **OS X**[w] appear to dominate the world of computing, most of the real hard-core and leading-edge computation relies on an elegant

operating system designed nearly 50 years ago (by a team of scientists who wanted to play a game).

0.8.3 Modern Unix Variants

The history of Unix is complex and convoluted, with the system being updated, re-implemented, and mimicked repeatedly over the years, primarily by commercial companies who guarded their versions jealously. Figure 12 shows a tiny fragment of the Unix's 'family tree' (the full diagram, which you can find at `www.levenez.com/unix/unix.pdf`, is *many* times the size of the portion you can see here).

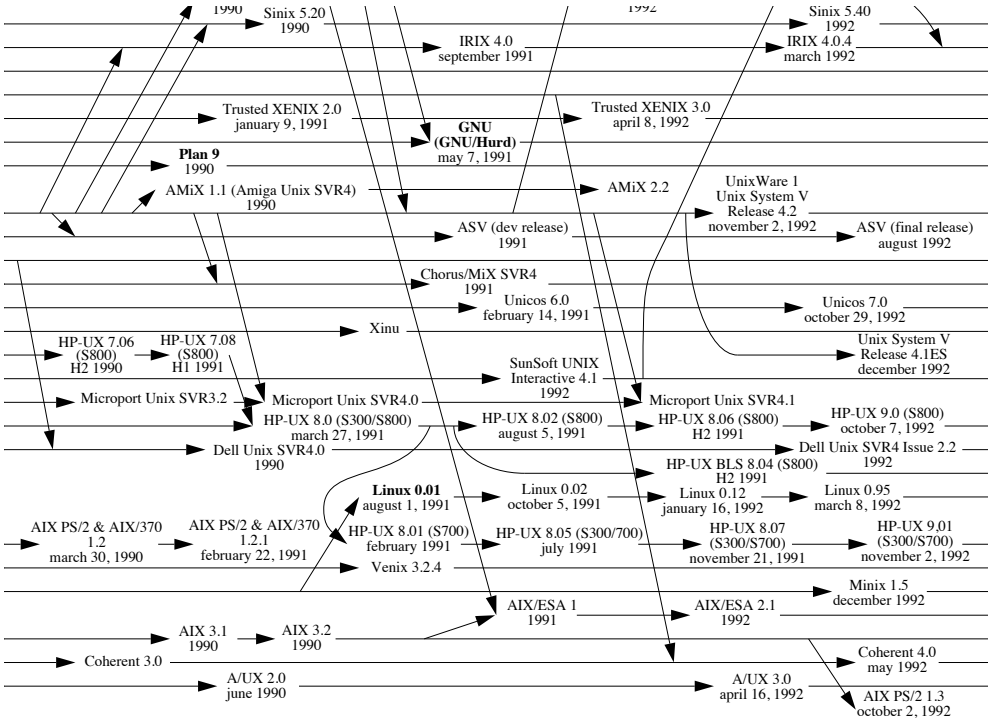


Figure 12
A fragment of Éric Lévénez's Unix History chart, reproduced with permission and showing the beginnings of Linux in amongst other versions of Unix.

Although many of the branches represent interesting innovations of one kind or another, there are perhaps two that deserve particular attention. The first of these was the decision by Apple some time around the turn of the millennium to drop their own—highly popular, but ageing—bespoke operating system (unimaginatively called **Mac OS**[W]) in favour of a Unix-based system (now the more familiar 'OS X', where 'X' is both the Roman numeral '10' and a nod in the direction of the uniX nature of the OS). Although the majority of Mac users are blissfully unaware of the fact, behind the slick front-end of OS X, sits a variant of Unix. The second, and perhaps more profound of these events was the creation in 1991 by Swedish programmer **Linus Torvalds**[W] of a Unix-like system, the source code to which *he gave away for free*[1]; this became known as the **Linux Kernel**[W]. Combined with other free software created by the **Free Software Foundation**[W], a non-commercial version of Unix called **GNU/Linux**[W] was born (GNU here is a recursive acronym for "GNU's not Unix", a swipe at other commercial non-Free versions; much to the annoyance of the Free Software Foundation, GNU/Linux is almost

[1] 'free' here in the sense both of 'freedom to reuse or adapt', and also in the sense of 'without charge'.

always called just 'Linux'[2].)

Linux has been, and continues to be, developed cooperatively by thousands of programmers across the world contributing their effort largely free of charge (although many are now paid to work on Linux as part of their job). It is amazing to think that such a project could ever happen—and it is surely a testament to the better side of Human Nature. But what is interesting is the observation that these programmers are not motivated by commercial concerns, but by the desire to make good reliable software and have it used by lots of people. Thus, Linux is a good choice of Unix: it's Free, it's efficient, and it's reliable, and it is now used by large corporations, governments, research labs and individuals around the world. Even Google's **Android**[W] platform is a Linux-based mobile OS, and the **Amazon Kindle**[W] is also a Linux box behind the electronic ink of its user interface (Figure 13).

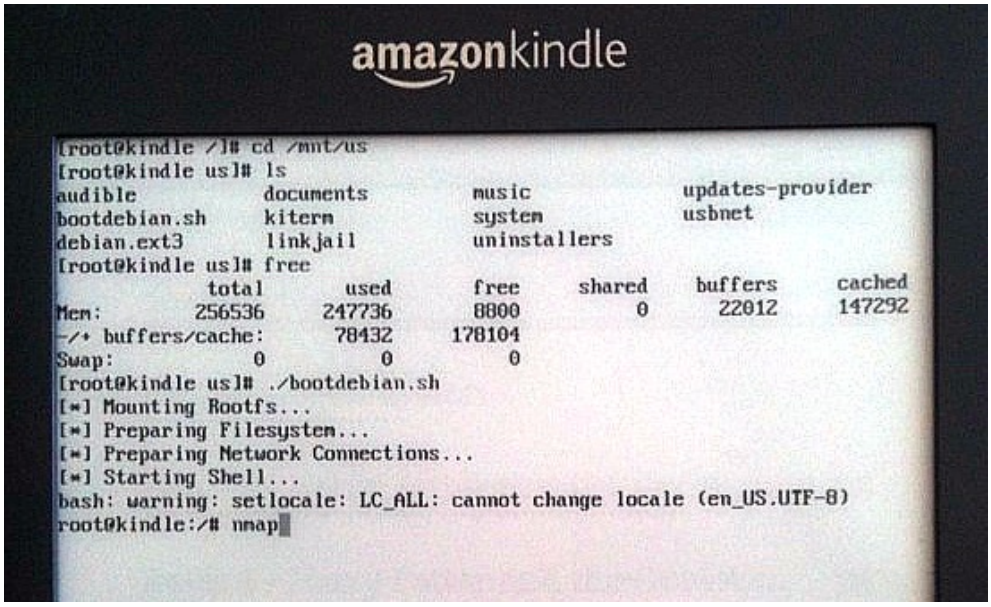


Figure 13
A photograph of Liraz Siri's 'rooted' kindle, showing the Linux command prompt. Reproduced with the author's kind permission from `www.turnkeylinux.org/blog/kindle-root`

One of the results of the fact that Linux is Free is that several organisations and companies have created their own distributions of it; these vary a bit (in fact, anybody is free to make any change they like to Linux, and pass it on to whoever wants it). The distribution we use in this School is **Scientific Linux**, which is based on a distribution by a US company called **Red Hat**.

So, if you are to become an expert computer professional, it is important that you understand the theory and practice of Unix based systems. Learning Unix is not only a crucial skill for any serious computer scientist, it is a very rewarding experience; the labs over the next couple of weeks are designed to help you become familiar with what will be your daily working environment.

When you have completed the work for this session, please tell the lab supervisor, so we know how you're getting on.

Please make sure you bring your Pi and this set of notes to all the introductory lab sessions.

[2]Linux is pronounced "Linn-ucks", despite the fact the name was coined by its creator, and his name 'Linus' is pronounced "Leen-uss"!