OS Class notes

Followup for CSC420 Operating Systems Spring 2022

Table of Contents

README

Instead of bugging you with emails, I opt to summarize my course observations regarding content, process, in this file. These often contain additional links, articles, and musings.

I usually update it after each class - it also contains the **homework** (if any). The first point of call for any questions should be the FAQ. There are two FAQs - a general one (for all my courses), and a FAQ for CSC420.

You find the whiteboard photos here in GDrive.

The companion file to this file, with the agenda and much of the course content, is the agenda.org file.

Welcome to the course - w1s1 (01/11/22)

Homework (by Thursday 13-Jan)

- Register with GitHub (5 min)
- Complete the GitHub Hello World exercise (20 min)
- ~~Give me your GitHub user name (email - with course ID - or Thursday in class)~~ (no need - course is public)
- Submit an issue to `birkenkrahe/os420` which confirms that you did the exercise!

Stuff

- Agenda - we covered all of it (and more) - agenda is available in GitHub only
- There's been a long and still open debate in software engineering and computer science about the importance of learning languages over learning concepts.
- When people speak of Linux they usually mean the 'kernel' (the part of the OS that's always "on"). When they say GNU/Linux, they mean kernel + a bundle of functions. The word "kernel" contains the German word 'Kern' for nucleus, or core (which in turn, I think, comes from the Latin word for heart, 'cor').

- Showed a few books to demonstrate the relevance of Linux and command line skills to cybersecurity:

- More about the APE (Ital. for "bee") parallel computer from the German particle lab DESY, Hamburg, and University of Rome La Sapienzia, Rome. Incidentally, one of the creators of this machine was Giorgio Parisi, who received the Nobel Prize for physics 2021 for his work on complexity (Parisi, 2022). One of my graduate student jobs was to take care of the first APE.

- Showed Windows Cmd line, Windows PowerShell, and the `bash` shell, and summoned the wrath of Bill Gates by putting down Microsoft's attempts to provide half-hearted OS access. The GIF shows how the OS world might look like without Microsoft, Apple, Google: everybody's running Open Source (Linux)!

- Demonstrated remotely accessing a Linux from a Windows box using `ssh` both on the shell and inside GNU Emacs (using the Tramp package and the program `plink`, which is part of `PuTTY`).

GitHub, GNU Emacs installation - w1s2 (01/13/22)

GitHub assignment/DataCamp

- DataCamp: You should all be in your courses now.
  - Your assignments are on one page but you'll be notified via schoology as soon as an assignment is due

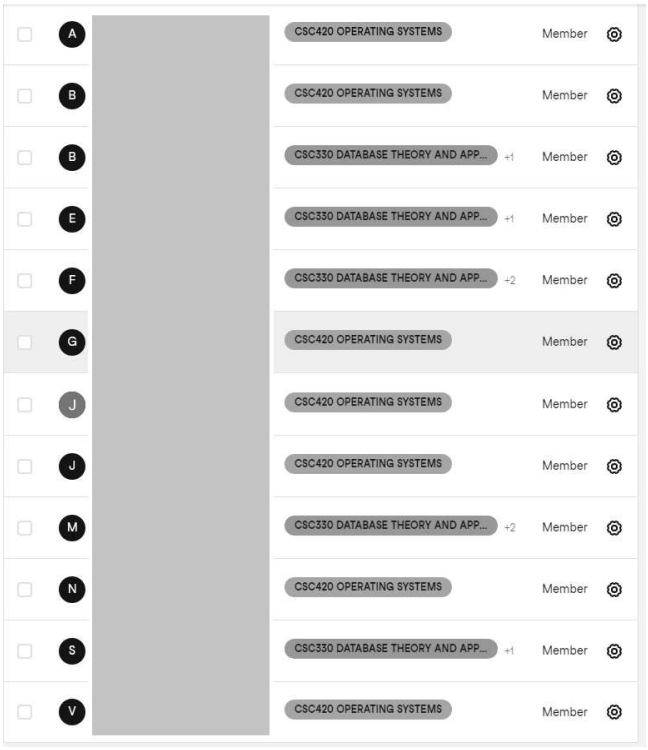

Figure 6: DataCamp OS course member list

- Importance of README files for packages that (super) users can build from source for their computer architecture (CPU) and OS
- Git is about version control, developed by Linus Torvalds, who also is the `main` branch maintainer for the Linux kernel. Here is his GitHub repo with the kernel.

- This is what I was thinking of - worth watching: The Computer Chronicles - Operating Systems (1984)

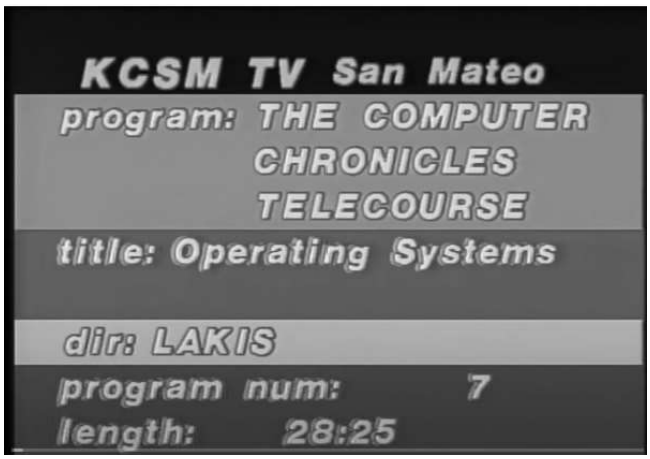

Figure 7: Computer chronicles-OS (1984)

  - However, my first thought, History of Databases (2019) is also really relevant for OS, because it shows how an important application (database management systems) is developed in concert with the operating systems on which it runs

Lecture on OS foundations

- Why different models? Each POV is a different model. Models change with time and represent focus and interests (e.g. IT careers)

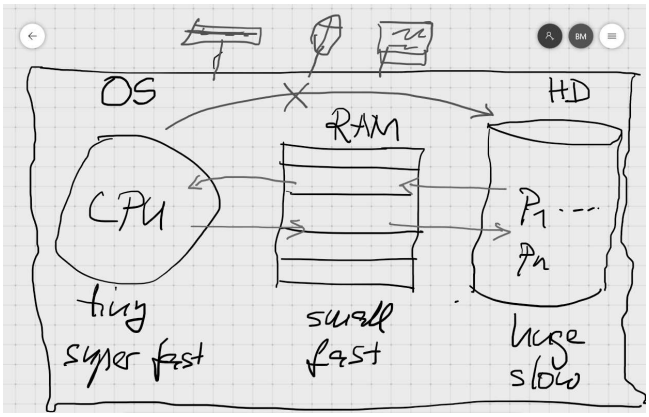

Figure 8: OS and computer hardware w/peripherals

- Diversity of architecture is like gaming success: a team consisting of 1 wizard + 1 dwarf + 1 elf has better chances to win than teams of three of each.

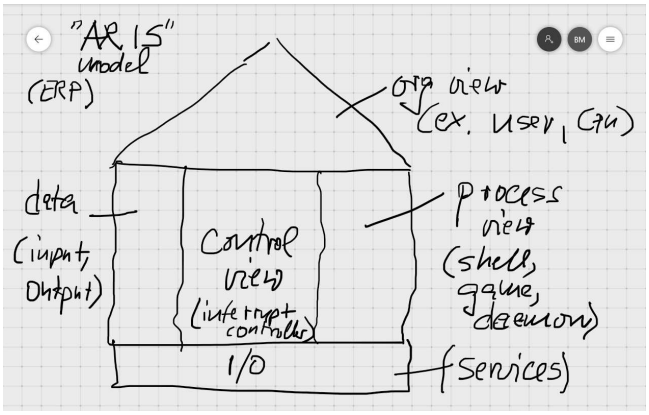

Figure 9: Architecture of Information Systems (ARIS) model

- Distinguish between PCs (solo), workstations (group, specific apps), and mainframe. They have different OS types.

GNU Emacs installation

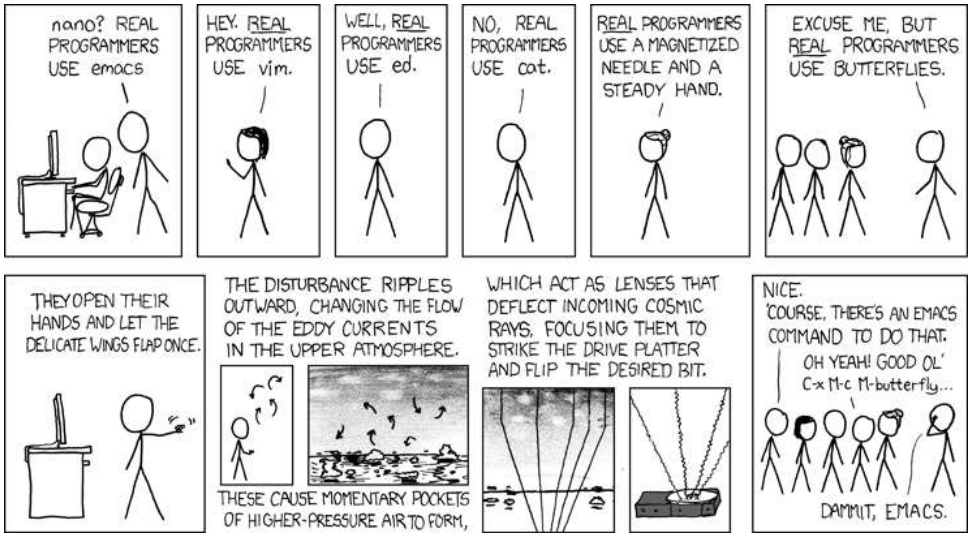

Figure 10: Real Programmers Use Emacs (Source: xkcd).

- xkcd (xkcd.com) is a cartoonist/nerd with a healthy appreciation for IT and computing (and also science)
- `nano` is a tiny GNU/Linux based editor
- `vim` is another (slightly larger) GNU/Linux based editor (available for Windows and Mac)
- `ed` is another editor, and `sed` on GNU/Linux is important for data science on the command line
- `cat` is a viewing program
- GNU Emacs v27 Windows installation: you need the `emacs-27.1-x86_64-installer.exe` from this page.

Interrupts, basic I/O - w2s3 (01/18/22)

Quiz 1

THE QUIZ IS ON … Schoology 9.30 - 9.45 AM

FOLLOWED by brief FEEDBACK:

- We'll do one of these per week
- Any content questions?
- Too much time? Too little?
- Questions too hard? Too easy?
- You should be able to see the correct solutions for rehearsal
- A subset of these questions will become the final exam

OS functions

(Source: the Computer Chronicles 1984)

1. Establish interface
2. Permit multiple users
3. Manage data files
4. Handle I/O
5. Error recovery
6. System accounting
7. Maintenance accounting

- CP/M: IBM PC operating system
- Apple IIe (BASIC) - early Apple PC
- FORTRAN (FORmula TRANslator - sci comp language)
- New kids: Go(lang), Rust,…but in OS, C programming rules

Review - OS position and purpose

ABSTRACTION[1]:

- The computer can be viewed from different angles (models):
  - Computer architecture (e.g. ARIS or CPU+RAM+NVM)
  - User + applications + OS + hardware

ARBITRATION[2]:

- The main (defining) job of the OS is to control and allocate computer resources
- Users want easy of use + performance + hidden from view
- Shared computers include mainframes and dedicated workstations

DEFINITION:

- The OS kernel is the program that runs on your computer at all times, and is always in (volatile) memory

COMPUTER PROGRAMS	EXAMPLES
Operating System (OS) kernel	`systemd` *daemon*
System (control) programs	`bash` *shell*

COMPUTER PROGRAMS	EXAMPLES
Application (user) programs	`emacs` editor

From bootstrapping to interrupts

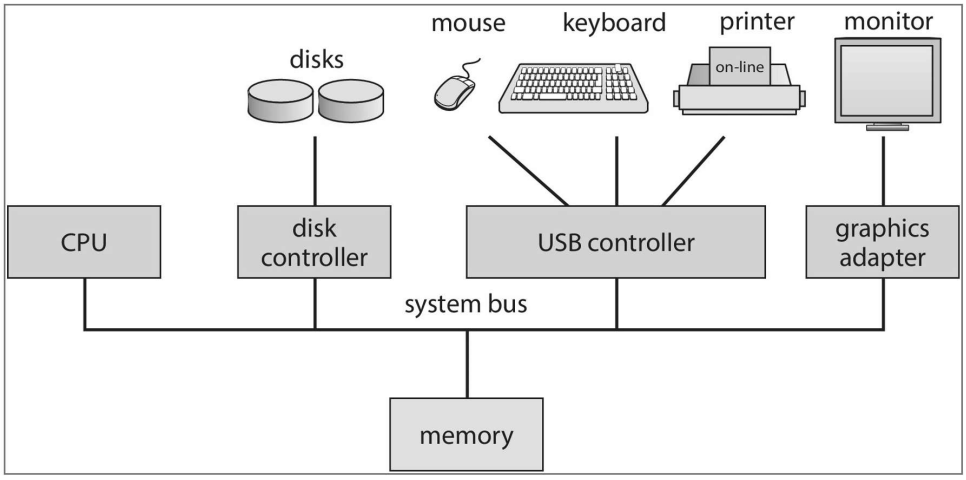

Figure 11: A computer system (Source: Silberschatz et al, 2018)

- Computer hardware = Motherboard (run) + Peripherals (I/O)
- In the past, application software had to explicitly be connected and configured to the operating system (MS DOS, Apple DOS, CP/M)
- Bootstrapping: the OS kernel program is loaded into (volatile) memory
- Booting = the BIOS is executed, establishing the basic I/O structure
- The system daemon (`systemd` in Linux) starts other daemons/services
- On Linux, firmware updates are not automatic, instead you apply them manually using `sudo apt update` and `sudo apt upgrade -y`
  - `sudo` give super-user privileges
  - `apt` is a package manager program for Debian Linux systems
  - `update` and `upgrade` are system programs
  - `-y` is a flag ("do it without asking questions")

Input/Output (I/O) Operation

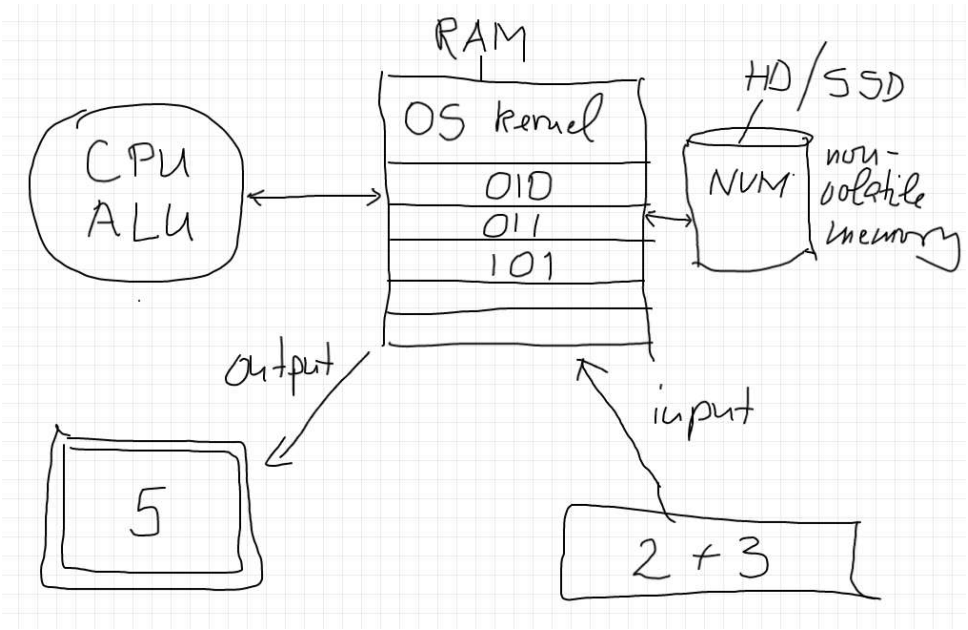

Figure 12: Input/Output (I/O) operation (schematic)

Organizational unit	Interrupt action
Device controller	Raises interrupt signal
CPU	Catches interrupt signal
	Dispatches interrupt to interrupt handler
Interrupt handler	Clears interrupt
CPU	Waits for next interrupt

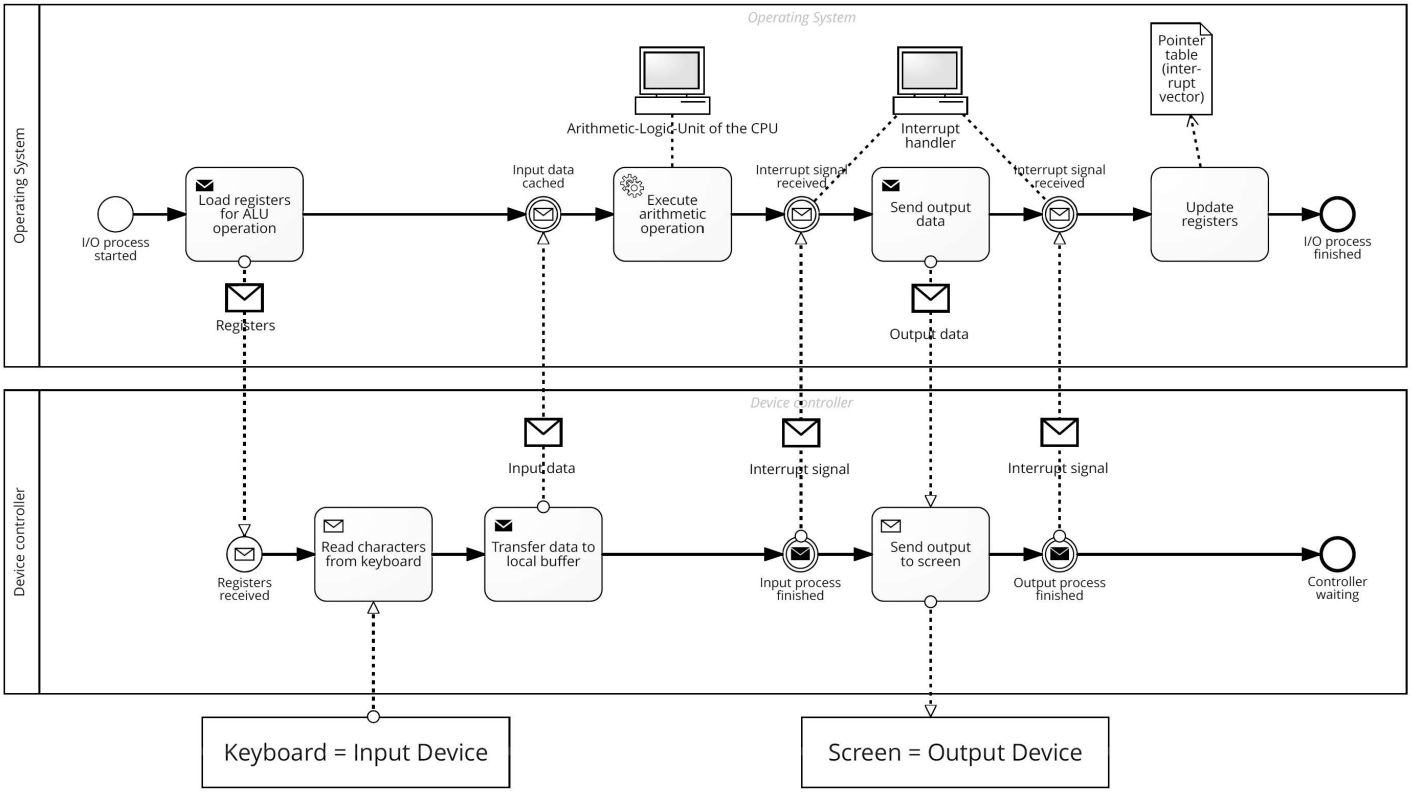

Figure 13: Input/Output (I/O) BPMN process diagram

- Here is a tutorial to develop a simple device driver is developed (Savin, 2022). It doesn't do anything else but read characters from input and generating output on a device. The driver has to be attached to a file using the `mknod` - then the device can be addressed/configured etc. using that file.

OS tasks, virtualization, GNU Emacs - w2s4 (01/20/22)

What does the OS manage? (With Linux examples)

- Task / process management (ps -a)
- Memory management (df -H)
- File-system management (ls -la)
- Network management (ifconfig) (ping)
- Mass-storage management (sudo fdisk -l)

Batch mode vs. interactive timesharing

- Multitasking is a smart scheduling illusion
- Virtual memory allows execution of large jobs
- Timer used to prevent infinite loop or resource hogging

User mode vs. kernel mode (with Linux examples)

- Mode-bit (hardware) indicates user vs. kernel mode (hostname -a)
- Kernel mode is privileged (sudo)
- Modern OS: Virtual Machine Manager mode for guest VMs (VirtualBox)

Captain's Log

- We looked at different virtual OS solutions under Windows
  - Dual boot (incl Linux on USB)
  - Ubuntu Linux as a Microsoft App from the Microsoft Store - there are pros and cons (Delony, 2022), but more pros!
  - Cygwin (not a complete OS but many Linux functions)
  - `ssh` to a Linux box (I showed this for the Pi) - something you can do from within Emacs with the (pre-installed) `Tramp` package (FAQ).
  - VirtualBox (works for many different OS)
- We compared open source and commercial systems ("bloatware")
- Linux has a package manager concept - you download/install only the software that you really need.
- Linux will run on very old computers ("bare metal")
- Three key issues in OS:
  - persistence (memory hardware)
  - concurrency (process mgmt)
  - virtualisation (memory mgmt)
- `sudo` is the Linux command for mode-bit = 0 ("kernel mode")

OS foundations, Eshell - w3s5 (01/25/22)

Captain's Log Star Date 99667.82

- Star Date Calculator
- Linux for iOS devices
- Termux: Linux for Android (get it via FDroid, not Google Play)
- What is CPU Clock Speed? (Intel)
- SQL injection (Computerphile, 2016)
- Online IDE: repl.it (replit.com) - REPL = Read-Eval-Print-Loop
- `bash(1)` shell program - Unix manual page

Shell scripts, Raspberry Pi setup - w3s6 (01/27/22)

Captain's Log Stardate 99672.08

- Assembly: last night, all the Pis were built with an improvised, family-based assembly line. Here is the complete process from start to finish in the hands of my daughter Lucia.

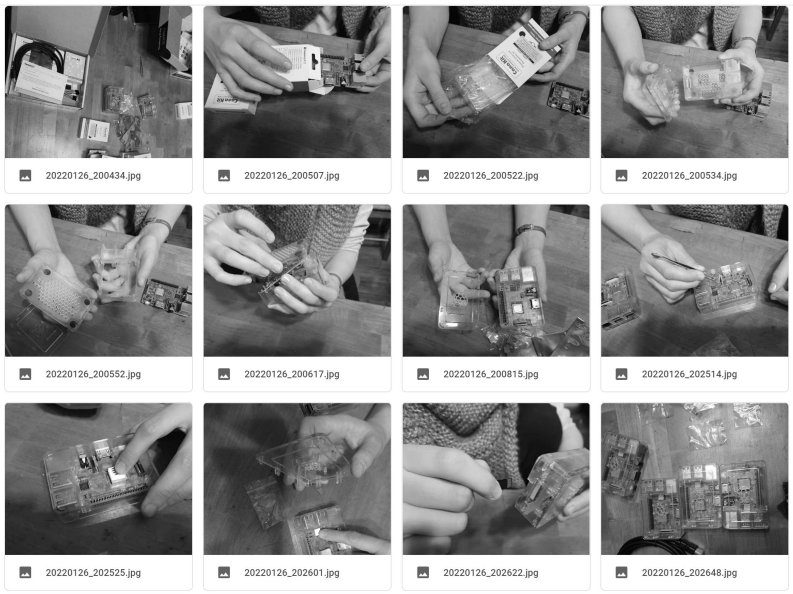

Figure 14: assembly of the CanaKit Raspberry Pi for Dummies

- NOOBS stands for "New Out Of Box Software" - an SD card-based installer for Pi. The Raspberry Pi Foundation now recommends to use an imager instead (Source).
- Windows Imager: https://rufus.ie/en/ - takes an OS image and unpacks (=flashes) it on a mass-storage device (e.g. a USB stick).
- Better (for all OS): Balena Etcher. Works like a charm. Just for fun, try to flash GNU Linux Trisquel on a USB Stick: "Run free!". I've got this on my Windows PC via VirtualBox.
- `Hello World` shell program problem under Windows: I realized that I had `bash` because I had installed the `cygwin` suite of tools. This is a really simply installation and you should do it (on your PC). Opens a world of (emulated) Linux commands.
- Cygwin installation: go to this page, download and run the file `setup-x86_64.exe` that you see linked to at the top of the page. This will give you the Cygwin App and also bash (I believe). Worth trying out - let me know if it works!

- Started the installation at 10.15 AM - and I personally finished the last installation at 3.30 pm of that day. Why did it take this long? I think mostly because the Pi's had a hard time getting all the stuff from the network - perhaps it's Lyon's fault. We'll never know because we won't have to do it again! Cp. "Installing Raspbian with NOOBS" (Source).

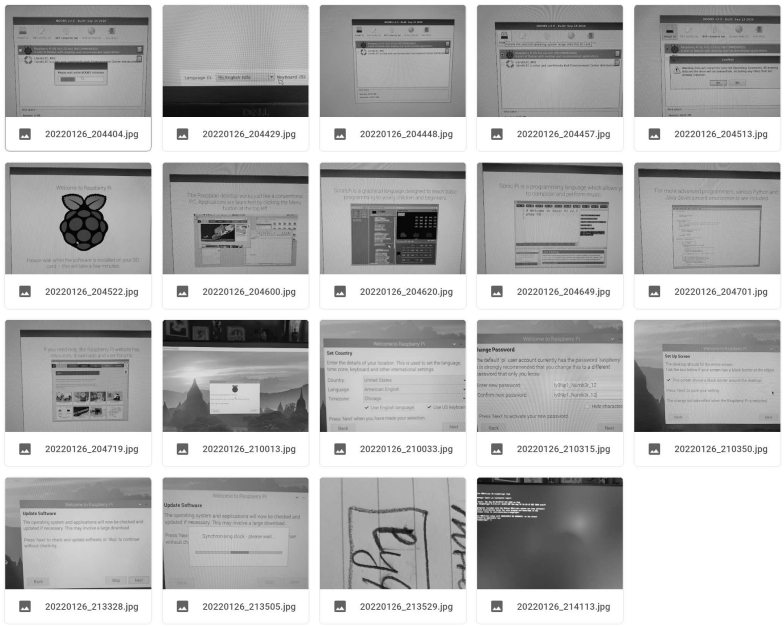

Figure 15: NOOBS-based installation of Raspbian Linux "Buster"

- Here are some views of the lab during installation. The last picture show Pi no. 12 attached to the front reception desktop PC so that I can present from the front.

Figure 16: Lab views during and after installation

Linux shell, UNIX man pages - w4s7 (02/01/22)

Captain's Log Stardate 99687.04

- I installed the OS (Raspbian OS "Buster" - which you can find out with the command `cat /etc/os-release`) on 10 of the 12 Pi's, as well as Emacs, SQLite, and R.

```
marcus@LCjvyz1b3:~$ cat /etc/os-release
NAME="Ubuntu"
VERSION="20.04.3 LTS (Focal Fossa)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 20.04.3 LTS"
VERSION_ID="20.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=focal
UBUNTU_CODENAME=focal
marcus@LCjvyz1b3:~$
```

Figure 17: /etc/os-release in the Windows Ubuntu App

```
pi@raspberrypi:~ $ 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"
pi@raspberrypi:~ $
```

Figure 18: /etc/os-release on Raspberry Pi

- UNIX man pages example: GNU manual page for `mv`
- To start the Desktop GUI on the Pi: enter `startx` in the terminal
- Popular Linux distros:
  - Kali (cybersecurity / hack functionality)
  - Trisquel9 (by GNU project, politically clean)
  - Pop!_OS (popular for gamers)
  - Ubuntu (extra popular, also App on Windows)
  - Arch (minimal Linux for budding experts)
  - Manjaro (available via Raspberry Pi Imager) based on Arch Linux
  - …and many more - see Best Linux Distros for 2022 (Tirthakar, 2021) for fanboys/fangirls.
- Then there are the great architectures - Debian, SUSE etc. - more about that when we talk about the history of Raspbian Linux for Pi.

- Install commands with the `apt` package manager program:

```
$ sudo apt install [package]
```

  E.g. for `tree` as `[package]`.

- `whoami` returns the current user
- `/` is the root directory, `$HOME` of the user `root` (super-user)

Shell commands, Linux file tree - w5s8 (02/08/22)

- should you upgrade my Operating System (Windows or Mac) when you're asked to do so? (Question for discussion - interested what y'all think, and what your experiences are).

- Feb 5, 2022: "Microsoft could be on the verge of forcibly upgrading Windows 10 on your computer"

  > "On May 10, in three short months, anyone who is still running this particular version of Windows 10 will no longer qualify for support from Microsoft. And this is why Microsoft is planning to forcibly update people to a supported edition

> of the operating system." (Wycislik-Wilson)

- Related: updates and upgrades on Linux[3]. Do this now:

```
sudo apt update -y
sudo apt upgrade -y
sudo apt autoremove -y
```

- What about Rasbian Linux releases? (Long, 2021).

> "Every two years, Debian Linux, on which Raspberry Pi OS is based, gets a major version upgrade. Debian 'buster' has been the basis of Raspberry Pi OS since its release in 2019, and Debian 'bullseye' was released in August. (As some of you may know, Debian name their versions after characters in Disney/Pixar's Toy Story films – Bullseye was Woody's horse in Toy Story 2.)"

- Raspberry Pi documentation - e.g. on terminal/Linux commands, or on the Raspbian Operating System.
- Inspiring video (11 min): Raspberry Pi Projects (TJ Free, 2020).
- Library: Raspberry Pi Ebooks in the Mabee-Simpson library

* Glossary

WHAT	DEFINITION	PURPOSE
Motherboard (h)	CPU + controllers + system bus + memory	Connect with I/O devices
Bootloader (p)	Computer startup sequence	Find and load OS
ROM/EPROM (h)	[Erasable Programmable] Read-Only-Memory	Permanent firmware
CMOS (h)	Complementary Metal-Oxide Semiconductor	BIOS memory
BIOS (s)	[Basic Input Output System]	Identify/configure hardware
BPMN	Business Process Model and Notation	Process diagram language
sudo	Linux shell command	Super user privileges
distro	Linux distribution (e.g. Ubuntu, Raspbian)	

Legend: h=hardware, p=process, s=software

GPIO pins - w6s11 (02/17/22)

- The `gpio` "Oops - unable to determine board type … model NN" issue is known and here is a fix: you need to get the latest from GitHub and build it.
- Here is information about a Java I/O library, P4J, to access the PIN numbering on the Raspberry Pi 400.
- I could not find anything about the `locate` command not working on any of your Pis, model 3 or 4. If someone finds something, let me know!

Wildcards and links - w7s12 (02/22/22)

- I was whining about not being able to display double square brackets in Emacs Org-mode because they are displayed as links (and the brackets disappear). The solution is extra simple: `M-x org-toggle-link-display`.
- I talked a little about SQL injection: here is the YouTube video "Running an SQL Injection Attack" (again - I had already mentioned it a few wweks back). It's been a while since I've seen it and by now it's become a classic, it seems (Computerphile, 2016)

Mid-term speech, REPLit, redirection revisited - w8s14

- explains it well (docker.com):

> "A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another."

The concept is similar to a Java VM runtime environment: but instead of just running Java, you can run pretty much anything in the container. As the figure shows, the Docker separates the app from the Operating System (Linux, Windows, MacOS). This is convenient, because now you don't need to bother with the OS. But it also stops you from learning anything about how apps interact with the system itself. It's super cool if all you are about is building apps, especially web apps, like replit.com. It's not so cool if you're

up against legacy systems (old software or hardware), or if you actually like interacting with the OS (via the shell), or if you want to create anything new, or if your mojo is performance improvement (e.g. making algorithms or data pipelines faster), because that depends on deeper knowledge. The good news: everyone can install a container, and they safe (actually, that's another problem…more layers, more potential attack points). Here is a list of 6 issues (Brandon, 2021).

Long story short: important concept and technology, you should try it out and explore it a little, perhaps you fall in love, and in the least you get another marketable skill.

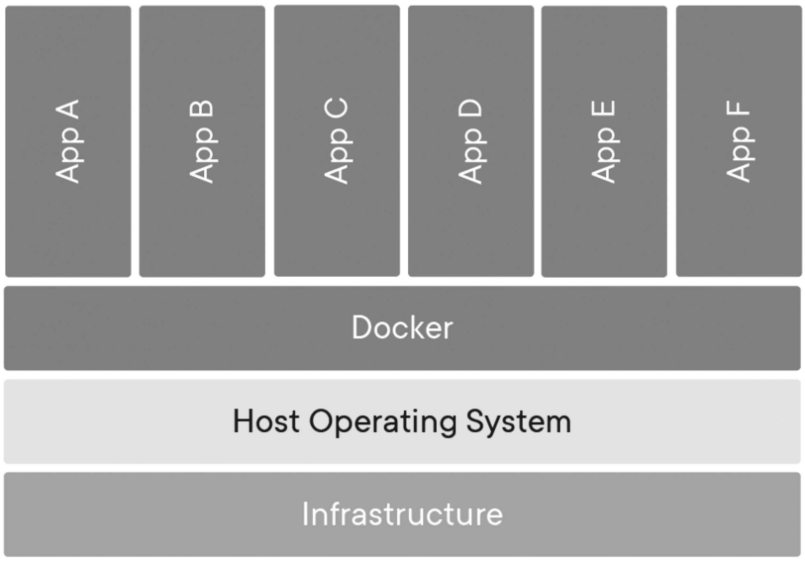

Figure 19: Containerized applications (docker.com)

References

- Brandon (Apr 10, 2021). 6 Problems with Container Technology [blog]. URL: ondat.io.
- Computer History Museum (Sep 5, 2019). History of Databases [video]. URL: youtu.be/KG-mqHoXOXY.
- Computerphile (Jun 15, 2016). Running an SQL Injection Attack - Computerphile [video]. URL: youtu.be/ciNHn38EyRc.
- Grubb (2021). How Cybersecurity Really Works. NoStarch Press.
- Intel (n.d.). What is Clock Speed? [website]. URL: intel.com.
- Long (8th Nov 2021). Bullseye - the new version of Raspberry Pi OS [blog]. URL: raspberrypi.com.
- Raspberry Pi Foundation (n.d.). Installing Raspbian with NOOBS [website]. URL: projects.raspberrypi.org.
- OccupyTheWeb (2018). Linux Basics for Hackers. NoStarch Press.
- Parisi (14 May 2002). Complex Systems: a Physicist's Viewpoint [preprint]. URL: arxiv.org.
- PuTTY (n.d.). SSH and telnet client program for Windows. URL: www.putty.org.
- Savin (2022). Linux Device Drivers: Tutorial for Linux Driver Development [website]. URL: www.apriorit.com.
- Simma (2004). Parallel Computing on APE Systems [website]. URL: www-zeuthen.desy.de.
- Wycislik-Wilson (5 Feb 2022). Microsoft could be on the verge of forcibly upgrading Windows 10 on your computer [blog]. URL: techradar.com.
- The Computer Chronicles (Nov 8, 2012). Operating Systems (1984). URL: youtu.be/V5S8kFvXpo4.
- Tirthakar (2021). Best Linux Distros for 2022 [blog]. URL: linuxhint.com.
- TJ Free (Mar 13, 2020). Raspberry Pi Projects [video]. URL: youtu.be/ZDfhcA0SCiM.
- Ubuntu (2022). The Ubuntu lifecycle and release cadence [website]. URL: ubuntu.com.
- xkcd (n.d.). A webcomic [website]. URL: xkcd.com.

Footnotes:

[1] Word origin: *abs-trahere* (Latin) = to withdraw [from details] - abstraction is the opposite of specialization (add detail)

[2] Word origin: *arbitrari* (Latin) = to resolve a dispute [because on a computer many different parties compete for available resources]

[3] Many Linux distros have LTS (Long Term Support) versions, e.g. Ubuntu.

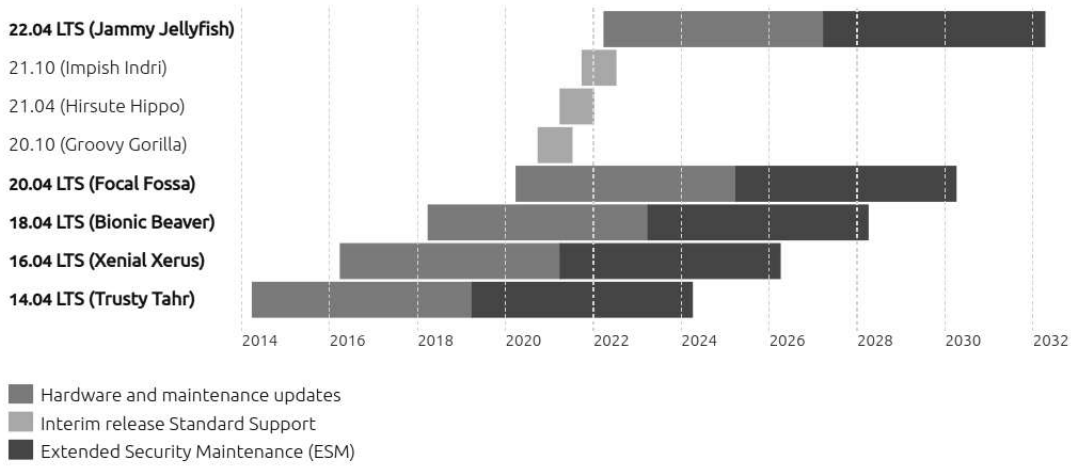

Figure 20: Ubuntu Long Term support and interim releases (Source: Ubuntu)

Author: Marcus Birkenkrahe
Created: 2022-03-11 Fri 22:39
Validate