# Operating Systems

## Introduction

Ubuntu is one of the many distributions that use Linux as a kernel; which is the most commonly known as Linux distribution. In this document, I will be talking about how operating system works, what software is used, how the operating system is loaded, the process scheduler, the file manager, the memory manager, the device manager and what is a boot loader.

## What is an Operating System?

[[1]](#footnote-1)An Operating System, at a high level of abstraction, is a program. It combines software to form a user-friendly platform for them to do their computing. It does things such as scheduling when programs should be run; managing hardware resources such as the memory, CPU and I/O devices.

The dominant operating system is Microsoft Windows with a market share of around 82.74%. macOS by Apple in second place at 13.23%, and the distributions of Linux in third place at 1.57%.

[[2]](#footnote-2)Operating systems usually come in three different types, 64-bit, 32-bit and ARM. This defines how many much memory the processor can access. “A 32-bit system can access 232 memory addresses, i.e. 4 GB of RAM or physical memory. A 64-bit system can access 264 memory addresses, i.e. actually 18-Billion GB of RAM. In short, any amount of memory greater than 4 GB can be easily handled by it”. “A major difference between 32-bit processors and 64-bit processors is the number of calculations per second they can perform, which affects the speed at which they can complete tasks.” ARM stands for Advanced RISC Machines. ARM processors are designed to do fewer computations per second then a 32-bit or 64-bit processor, but also designed to have fewer instructions. By optimising instructions on the processor, ARM processor then have the advantage of being more power efficient.

## How does an Operating System run?

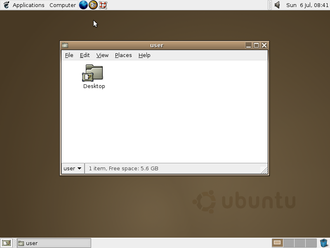
[[3]](#footnote-3)At boot, before the operating system can be loaded. The boot manager is loaded from disk and then from there the operating system is loaded. Ubuntu uses the boot manager GRUB (GRand Unified Bootloader). The process of the bootloader is:

Stage 1: (Located in the MBR) Instructions that point to load Stage 2.

Stage 2: Instructions that points to its configuration file (Which can be located anywhere on disk), which contains all the complex user interface; or if a configuration file is not found, it defaults to a command line where manual configuration is needed.

Note: The MBR (Master Boot Record) holds the information such as how partitions are configured, how file systems are configured, and how they are organized on the disk. The MBR also contains the code to launch the boot manager, which are what the instructions are in Stage 1.

## What is Ubuntu?

[[4]](#footnote-4)Ubuntu was made by Mark Shuttleworth. Ubuntu was founded when the company Canonical was made in October 20th in 2004. It is an operating system that runs on top of the Linux kernel as well as all other Linux distributions (Otherwise known as distros).

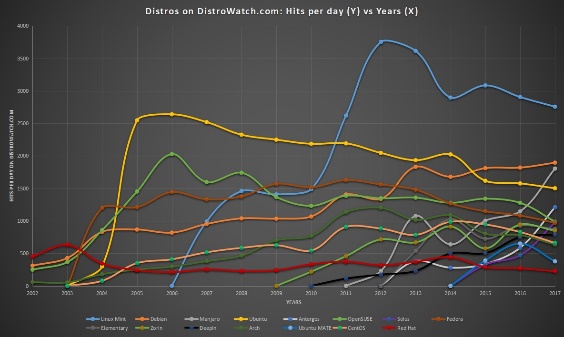
[[5]](#footnote-5)Ubuntu’s first iteration was called Warty Warthog (figure 1 shows what it looked like). It was based upon an existing distribution called Debian (This is common practice today) which had been out since August 16th, 1993, as documented in Debian’s documentation. In figure 2 we can see that Ubuntu’s (Yellow line) popularity skyrocketed when it was first introduced. This was mainly because canonical gave out free CDs with the Ubuntu Operating System burned onto it.

Figure 2

Figure 1

Ubuntu’s first proper desktop environment Unity came into play when Maverick Meerkat was released in April 2010. This decreased Ubuntu’s popularity. As shown in the picture at the same time Ubuntu Mate’s (Blue line) popularity skyrocketed instead. Ubuntu Mate was Ubuntu with a much nicer looking Desktop environment at the time, so everyone moved to it. There was no disadvantage.

## What is Linux?

[[6]](#footnote-6)Linux is a file-based operating system. This means that Linux does not have a registry like Windows. Every program on Linux has a configuration file, including all software involved with the operating system. This is what makes Linux special because it is so modular. Linux is not so dissimilar to windows though. It contains all the normal things like a bootloader and applications.

Linux runs on almost every server in the world. This is because it is so widely documented. As it is open source, when a security vulnerability is found in the code. Everyone knows about it and everyone has the ability to fix it.

## Who Created Linux?

Technically, Linus Torvalds developed Linux; but today the reason it is what it is today is because of Richard Stallman. Richard Stallman developed GNU. Which funnily enough stands for GNU not Unix, these funny recursive acronyms are a common occurrence through Linux module names. GNU is the desktop environment that runs on top of the kernel; this is what Ubuntu is.

Today every Linux distribution you download will include GNU. The Linux kernel and GNU go hand in hand as the Linux kernel by itself is basically useless, as put by Linus in his documentation in the Release notes when Linux was at version 0.01, he wrote [[7]](#footnote-7)“Sadly, a kernel by itself gets you nowhere. To get a working system you need a shell, compilers, a library etc.”

## How does Ubuntu handle Process Management?

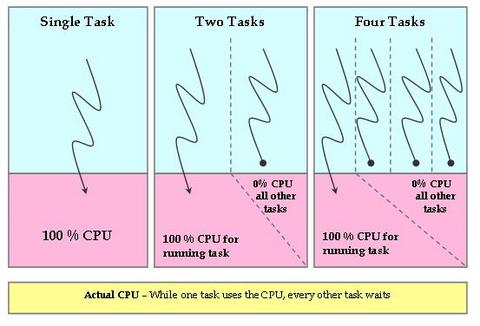
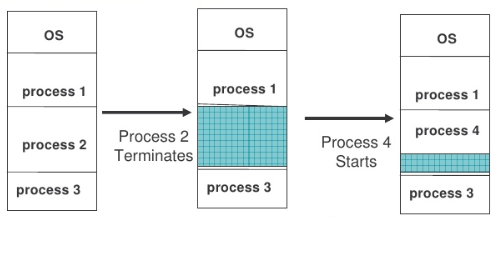
[[8]](#footnote-8)Process scheduling is handled by the Completely Fair Scheduler (CFS) in the Linux Kernel. The CFS is possibly the most important piece of software in the kernel as it is what makes our computers seem like they are running programs seamlessly. The CFS was added to the Linux kernel in kernel version 2.6. This is because the old Linux Scheduler was biased. The difference between the current scheduler (CFS) and the old Linux scheduler is that the current one realizes constantly on time. This is because the CFS calculates the maximum time a task should have to run for on an ‘ideal processor’. While the task is running it must keep track of how long the process has been running for and therefore it needs to have access to the time constantly.

Figure 3 shows exactly what the CPS tries to eliminate. This is because if one task takes up 100% of the CPU power, all other tasks must wait which ends up with programs freezing (When a program is allocated no processor time).

Figure 3

When the scheduler is invoked to run a new process, the operation of the scheduler is as follows:

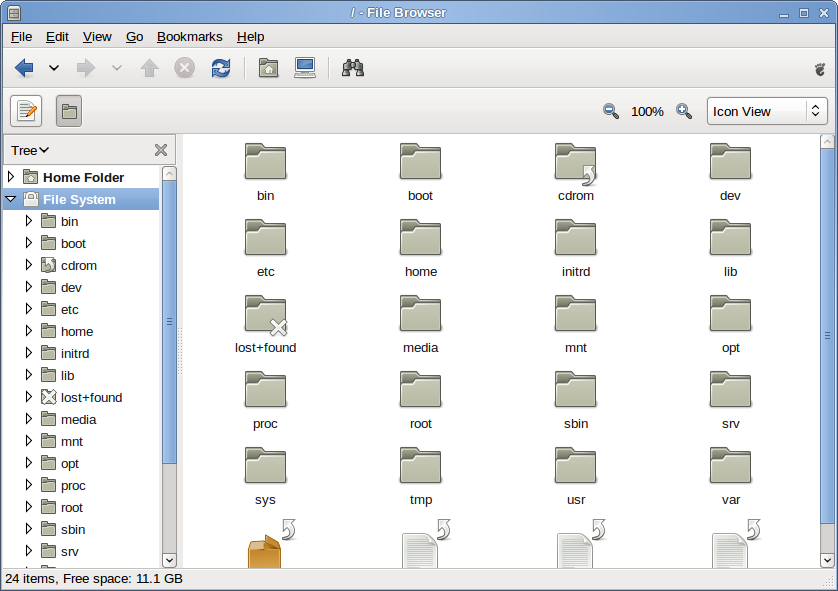
1. The task with the least amount of time needed to execute is sent for execution.
2. If the process simply completes execution, it is removed from the system and scheduling list.
3. If the process reaches its *maximum execution time* or is otherwise stopped (voluntarily or via an interrupt) it is reinserted into the scheduling list based on its new spent *execution time*.
4. The new task with the least amount of time to execute will then be selected from the list, repeating the iteration.

If the process spends a lot of its time sleeping, then its spent time value is low, and it automatically gets the priority boost when it finally needs it. Hence such tasks do not get less processor time than the tasks that are constantly running.

If a process terminates then the processor will try to fit another process in there the last process terminated so as not to lose processing time. As shown in figure 4.

Figure 4

## How does Ubuntu handle File Management?

[[9]](#footnote-9)GNOME Files, formerly and internally known as Nautilus is the file manager used by Ubuntu. Nautilus was originally released by Eazel in 2001 and incorporated into GNOME 1.4. GNOME files incorporated a lot of features Files manager that a lot of a lot of other file manager hadn’t incorporated at the time such as a fast-integrated search manager, a bookmarking system, a history tracking system, a zooming function and a view panel that allowed you to change the way you viewed files. Figure 5 shows what it looked like.

Ubuntu GNOME file manager came a long way from what it was. As because before this using the terminal to list off files in a directory was normal. Productive was increased dramatically with this software.

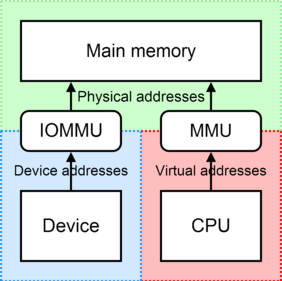
Figure 5

Everything you do within GNOME files; GNOME files must do in a terminal in the background. Such as the following shown in Figure 6.

Figure 6

## What is Memory Management?

[[10]](#footnote-10)Linux memory management subsystem is responsible, as the name implies, for managing the memory in the system. This includes implementation of virtual memory and demand paging, memory allocation both for kernel internal structures and userspace programs, mapping of files into processes address space.

[[11]](#footnote-11)Memory in the eyes of the Linux kernel can be seen in two different ways; as virtual memory and physical memory. Virtual memory is very complex but is very useful as it allows for the hard drive to store programs in ‘Huge Pages’. These huge pages are most commonly used for caching where data that might be used by the processor again is stored or where when programs are inactive, they can be stored on the hard drive instead of taking up memory. The reason Linux uses so much memory for disk cache is because the RAM is wasted if it isn't used.

What is important with memory management is to be careful where you allocate memory as you can easily overwrite another programs memory space. The OS’s memory allocation can easily be overwritten by a rogue program and cause the data on memory to be corrupted.

Differences between buffers and caches:

Figure 7

* The cache only contains the actual file data.
* Buffers are associated with metadata such as file permissions, what’s in directories and what memory has been edited and read from.

The memory manager is only concerned with:

* The computers available pool of memory.
* Allocating to applications and making sure they do not infringe on each other.

## How does Ubuntu handle Device Management?

Every modern operating system these days can detect when a device is connected to the system. The system is usually set up to run a program when such a device is plugged in; like when an iPad is plugged in, it will open a file directory or if a DVD is put in the DVD tray, it would open a video program.

[[12]](#footnote-12)The most common technical word in relation to device management is called hot-plugging. This is when a USB device can be plugged in and out of a system without causing any disruption to the system. This is achieved by using three pieces of software; Udev, HAL and Dbus.

Udev supplies the system with a directory to store nodes to reach the devices; such as /dev/keyboard. Devices get added and removed from the directory depending on if they are connected to the system or not. Dbus is in charge with communication with the device. Firstly, it picks from its vast amount of device drivers which is suited to the device and then attempt to connect via a virtual bus which connects to the device through the node in the dev directory. If the device has storage on it, HAL will create an XML file which has all the information that is needed for GNOME files to be able to mount the storage medium.

[[13]](#footnote-13)In the case of PCI devices, things get more complex. PCI devices cannot be hot-plugged as they must be recognised by the kernel at start-up. If a PCI device is added while the system is running, it can cause the system to crash. When the PCI device is added at boot it is given a unique string. In this instance which can be seen by Figure 8 the string is 1af4:1000 which is also known as the device address.

Figure 8

The device address is then passed onto IOMMU which is used to translate the device address to a physical address and vice versa. You can see it described in Figure 7.

## Bibliography

1 https://mirrors.edge.kernel.org/pub/linux/kernel/Historic/old-versions/RELNOTES-0.01.

2 https://www.geeksforgeeks.org/difference-32-bit-64-bit-operating-systems/

3 https://www.debian.org/doc/manuals/project-history/ch-intro.en.html.

4 https://www.techotopia.com/index.php/The\_History\_of\_Ubuntu\_Linux

5 https://www.debian.org/doc/manuals/project-history/ch-intro.en.html

6 https://www.linux.com/what-is-linux

7 https://mirrors.edge.kernel.org/pub/linux/kernel/Historic/old-versions/RELNOTES-0.01

8 https://en.wikipedia.org/wiki/Completely\_Fair\_Scheduler

9 https://en.wikipedia.org/wiki/GNOME\_Files

10 https://www.kernel.org/doc/html/latest/admin-guide/mm/index.html

11 https://www.youtube.com/watch?v=7aONIVSXiJ8

12 https://www.linux.com/news/udev-introduction-device-management-modern-linux-system

13 https://wiki.archlinux.org/index.php/PCI\_passthrough\_via\_OVMF.

1. https://en.wikipedia.org/wiki/Operating\_system [↑](#footnote-ref-1)
2. https://www.geeksforgeeks.org/difference-32-bit-64-bit-operating-systems/ [↑](#footnote-ref-2)
3. https://www.dedoimedo.com/computers/grub.html#mozTocId616834 [↑](#footnote-ref-3)
4. https://www.techotopia.com/index.php/The\_History\_of\_Ubuntu\_Linux [↑](#footnote-ref-4)
5. https://www.debian.org/doc/manuals/project-history/ch-intro.en.html [↑](#footnote-ref-5)
6. https://www.linux.com/what-is-linux [↑](#footnote-ref-6)
7. https://mirrors.edge.kernel.org/pub/linux/kernel/Historic/old-versions/RELNOTES-0.01 [↑](#footnote-ref-7)
8. https://en.wikipedia.org/wiki/Completely\_Fair\_Scheduler [↑](#footnote-ref-8)
9. https://en.wikipedia.org/wiki/GNOME\_Files [↑](#footnote-ref-9)
10. https://www.kernel.org/doc/html/latest/admin-guide/mm/index.html [↑](#footnote-ref-10)
11. https://www.youtube.com/watch?v=7aONIVSXiJ8 [↑](#footnote-ref-11)
12. https://www.linux.com/news/udev-introduction-device-management-modern-linux-system [↑](#footnote-ref-12)
13. https://wiki.archlinux.org/index.php/PCI\_passthrough\_via\_OVMF [↑](#footnote-ref-13)