Diploma In Computer Science

Contents

Lesson outcomes

By the end of this lesson, you should be able to:

- Identify the most popular operating systems just by looking at their features
- Understand how operating systems work, and why they are so important to computers
- know the technologies used in handling information

Introduction

In our previous lesson, we looked at the roots of computer science and the road that got us where we are today. In this lesson we are now going to dissect the process that information takes, from acquisition, organisation, storage, processing, and retrieval. We will look at the structure of one of the most important parts of a computer- the operating system. We will explore the structure of the operating system and look at some of the popular operating systems and where they are used.

Information processes

Example – baking a cake

To make this easier to understand, we'll look at an example of baking a cake, which we will refer back to a few times:

- First, you gather all the requirements, that is, your flour, sugar, milk, eggs, and all
- You follow a recipe and mix those ingredients, before putting the mixture in an oven
- Sometime later, you have a cake

It doesn't truly end here though, as systems are constantly improved and perfected. Think of how your phone or computer has received several software updates since you started using it.

Collection of information

Data collection is the first step in information processing. Data is collected from various sources, such as the environment, input files, previously processed information or generated by the computer. Sensors are used to gather information from the environment - we will take a closer look at this in lesson 4.

The data must be in a specific format that is set out by the program that is going to process it.

- The flour has to be in a specific "format"- you can't go to the field, come back with a sheaf of wheat and pop it into the bowl as it is
- The wheat has to be "formatted" correctly so that we can use it as flour
- The same applies to all the other ingredients that we put in

Data

- Input is generally stored in a file or a database
- Data is converted into a machine-readable format by input devices
- A keyboard is a very common input device that nearly everyone who has use a computer has used
- Nowadays, computers can collect their own data from the environment with little or no human intervention

Organising data

Data is processed using software programs. The Central processing unit uses the input data and transforms it using instructions that are known as programs. From module 2, you will learn to make your very own programs.

Back to our baking example:

- The program would be the recipe - it tells you exactly which ingredients (inputs) to use, the sequence in which you put them, how long you mix the dough, how long you bake, and the oven temperature
- In this process, you are the central processing unit (CPU) that processes everything. Much like our raw ingredients, data isn't much use if it is not processed
- Processing produces information, which is the useful product of processed data, like our freshly baked cake

The speed of data processing is dependent on a number of factors, such as the processor that the computer has, the bus size, the cache size among other things.

After data is processed, it needs to be stored somewhere so we can use it when we are ready. Computers use two methods to store output; it is either stored temporarily, or permanently.

- Temporary storage is called memory and is referred to as storage when permanent. Temporary storage is called RAM, short for Random Access Memory. As soon as the power is switched off, all contents in RAM is lost. RAM is used for information that is being used at that point in time.
- When information is needed at a later stage, it is moved to storage. Your computer's drive, your flash drive or your phone's storage are examples of storage. Information in storage remains in place even after the computing device is switched off. Information can be stored as files or in a database.

Displaying information

Information can be presented in various ways. The most common way is visual output, that is the information typically displayed on a screen for you to see or read. Screens are the most common form of output on computers.

Another form of visual output is printed content. Output can be presented as sound, such as music, or as actions of a machine, such as an assembly line robot painting a car.

The Operating System (OS)

An operating system is a set of manual and automatic procedures that enable a group of people to share a computer installation efficiently. The key word in this definition is sharing: it means that people will compete for the use of physical resources such as processor time, storage space, and peripheral devices; but it also means that people can cooperate by exchanging programs and data on the same installation.

The kernel

The kernel is at the very core of the operating system. It has complete control over everything in the system. The kernel is part of the operating system that is always resident in memory. In memory, the top part of the address space is reserved for the kernel. Think of it as the "VIP area" or the top management of a company, the CEO, COO etc. the kernel also consists of very important files called device drivers.

Booting & BIOS

An operating system is typically resident on the computing device's storage, such as a Solid-state drive or a hard disk drive. When you press the power button on a computing device, you initiate a process called booting.

Every computer has a set of special instructions called BIOS (Basic Input Output System), which runs immediately after you press the power button. This set of instructions powers on and initialises hardware, performs tests to make sure that

the hardware is ready to run, then loads another set of instructions called the boot loader. The bootloader is a special low-level program that takes over as soon as hardware is initialised. The bootloader's job is to locate the OS on the storage device and load the kernel into RAM. Once the kernel is loaded and ready to execute, it takes over from the bootloader and loads secondary services, drivers, and apps, preparing the computer for use.

The computer essentially pulls itself up, step by step, until it is ready for operation. Booting is complete when the runtime state is achieved. You usually know when this state has been achieved; that is when you can use the computer.

State Machines

All computers are state machines, a term that you will grow familiar with if you decide to take computer science further. It essentially means they go through various states to accomplish their tasks. The beginning of the boot process is the zero state of a computer. Sometimes we need to reset a machine to its zero state, and this is achieved by rebooting. This generally means unloading all programs and then restarting the boot process.

Rebooting can either be hard or soft:

- In hard rebooting, power is cut off from the system when unloading is complete
- In a soft reboot, only the contents of RAM are cleared to zero

Android uses OEM specific bootloaders because the OS is contained in firmware. Linux typically uses the grub (GNU Grand Unified Bootloader), which is a product of the GNU project, and Windows uses Windows Boot Manager

Operating System roles

Operating systems perform a lot of essential tasks. A few of the most important ones include:

- Memory Management: the OS gets to tell apps what can and cannot stay in RAM
- Processor Management: the OS tells the processor which tasks to run, when and for how long. This helps the processor to run as efficiently as possible
- The operating system manages devices using drivers. It makes sure that there are no conflicts between hardware devices
- The OS handle file management - keeping track of these and making sure they are not overwritten
- Security - your computer will never let you access anything that you're not supposed to, using passwords, passcodes, and permission management on files and on programs
- System performance and resource management - The OS prevents you fighting for time with the system, and always tries to strike a balance. That way, you get your work done and the hardware has enough time to process things for you
- Error handling - the OS makes sure any errors are handled internally without causing your computer to crash
- Application platform - most important of all, an OS provides a platform on which applications run.

Kernels

An operating system consists of 3 main components, the Kernel, the Shell, and utilities. There are five types of kernels:

Monolithic kernel

- user and kernel services are in the same space. This increases the size of the OS but results in faster execution. Linux systems typically have a monolithic kernel

Microkernel

- the user space and kernel services are separate. This reduces the size of the OS. This kernel uses message parsing for communication between applications and services, resulting in faster execution

Hybrid kernel

- A combination of the monolithic and microkernel, to take advantage of the speed of the monolithic kernel and the modularity of the microkernel. An example of this kernel is the windows kernel

Nanokernel

- As the name suggests, has a very small code base. It offers hardware abstraction, but no system services

Exokernel

- A very complex kernel design where as few abstractions as possible are imposed. This allows for application specific customisation, but results in a very complex kernel

Separations

Operating systems separate processing into kernel space processing and user space processing. This is essential to ensure isolation of individual computational processes. This prevents a catastrophic failure if ever there is a program failure.

The CPU is not the only smart component in a computer. The keyboard, mouse, drives, monitor, fingerprint sensor or USB ports have their own control and status registers, and these are not made the same way. These are used to start and stop the device, and for the computer to be able to communicate correctly with the devices. It would take a lot of resources for every application that uses hardware to have its own controller software.

Device drivers

Linux has a genius way of handling this- this code is part of the kernel and is referred to as device drivers. This software is essentially the software that sits between a computer and a device.

The more technical description of a device driver is a shared library of privileged, memory resident, low level hardware handling routines.

- When using device drivers, all hardware devices are abstracted and represented by files, which can be read, written to, opened, or closed just like any other file, using the standard methods that the operating system uses

As an example:

- When you press the brightness button on your computer, the operating system simply writes your new brightness setting to a file, then calls the display driver
- The display driver then translates the setting to the actual commands that the display controller uses
- The display then sets the voltage that the panel needs to set your requested brightness

This happens in a split second and almost always works unless there is something wrong with the driver.

Writing device drivers is a job that a programmer can take up. Drivers need to be kept as efficient as possible, so programmers are constantly hard at work making existing drivers better. This is why your phone, computer, tablet, tv and sometimes even headphones need software updates from time to time.

Types of operating systems

Linux and Unix-like systems

Genetic Unix

- Shares the same code base as the original Unix system

Branded Unix

- Deviated from the original code base

Functional Unix

- Behaves similarly to Unix-certified systems, usually free versions

Mac operating system

The most common to the regular user would be mac OS, the OS that ships with every single machine produced by apple.

- This started off as a branch from the Unix system, to the BSD (Berkeley Software Distribution) and then to the Apple proprietary system
- It is just like Unix, right down to the shell commands
- macOS is solely developed by Apple Inc and has now taken its own development direction, but it still maintains the same Unix roots which give it security, stability, and speed
- Like its ancestor, it's written in C, but then adds C+, Objective-C, Swift and assembly language to the mix
- Mac OS has been generally praised over the years for general lack of malware, but however, around 2006, malware started appearing on the macOS system. They then started including anti malware software in the OS
- MacOS makes up about 13% of the OSs used in the world

With the arrival of GNU (short for GNU's not Unix), Richard Stallman attempted to create a "Unix-like" software, only that it would be free. In contrast, proprietary software is illegal to copy or modify without prior permission. Free software under the GNU General Public License meant that anyone could acquire a copy, add code, and modify existing code legally. This gave rise to Linux, a free, open source operating system that has many, many variants.

These Linux variants are referred to as distributions.

Linux distributions

A Linux distribution is a "flavour" of Linux that has been tuned and tweaked to suit a particular use case, such as Puppy Linux for lightweight systems or Linux mint aimed at users migrating from windows and just want a quick start.

Commercial variants

Linux comes in the form of commercial variants such as Red hat Enterprise Linux and SUSE enterprise. Desktop versions of Linux use a window manager such as x-window, and a desktop environment; the popular ones being Gnome, KDE, Mate and Xfce. Linux runs on over 96% of servers and is pretty much the only name you will come across mentioned in the same breath as servers and supercomputers.

Linux runs on an embedded system; that is, devices where the OS is built into the firmware and is typically made specifically for that system. The most popular form of Linux is probably what we have all interacted with at one point of our lives -Android.

Linux is typically used on servers- it has the lion's share here - and used also on desktop computers. A number of IoT (internet of things) devices use Linux, such as printers, TVs, fridges etc.

Android

Android is by far the most popular operating system of all. It is a free product of Google, distributed under the Apache license.

It is based on a modified version of the Linux kernel, plus a bunch of other software to optimise it for a touch interface. Android was released in November 2007 and has since grown explosively to be the single most used operating system to date, boasting a 2 billion user base, with over 85% of the smartphone market share to date.

It has grown so much that it's no longer just used on smartphones, but cars, mp3 players, printers, TVs, game consoles, cameras and even PCs.

The Linux kernel at the heart of Android incorporates further changes from the normal Linux kernel.

Android is actually a Linux distribution according to the Linux foundation, the non-profit consortium that works on standardising Linux. Android had a somewhat unique framework in that it has a kernel, libraries, and an additional layer known as ART (Android Runtime), a virtual machine in which android applications run. This is the reason why Android devices typically require more RAM than other similar devices.

Android evolution

- Android has evolved over the years to be a primarily touch based interface, having been used primarily on touch-based smartphones and tablets
- The operating system is praised for being powerful and customisable, unlike many other mobile operating systems. Android is customised by OEMs (original equipment manufacturers) with what are called “skins”, so called because they hardly alter android core functionality but have a unique user interface
- There are a lot of different android skins available, the most popular including OneUI by Samsung, OxygenOS by OnePlus, EMUI by Huawei, MIUI by Xiaomi and Sony Xperia UI

Skins

Skins are typically not transferrable from OEM device to OEM device, as Android is typically installed as firmware. Firmware is built specifically for a particular device and requires major modifications for it to run on a different device.

Samsung firmware, for example, will not run on a Sony handset. Even more specifically, the Samsung Note 20 ultra’s firmware will not run on the Note 20, even though the phones look very similar. This is usually the case with most computing devices, apart from PCs.

Android device OEMs usually include google play services in their firmware. This provides core functionality for Google Apps, as well as app updates through the company’s app store, Google Play Store.

Windows

Windows is a very common operating system, used on a lot of PCs, servers, and embedded systems. Windows is a product of Microsoft Corporation, and unlike Linux, it is closed source, meaning you cannot legally modify your copy of windows, and can only use it as is.

- Windows is typically the preinstalled OS on many desktop and laptop computers, and almost everyone who has used a computer has used this operating system
- Windows is primarily written in C++, while the kernel is written in C. some newer components of the UI are written in C# (pronounced c sharp) and in .net (pronounced dot net)
- Windows is by far the most popular desktop operating system, amassing over 77% of the desktop operating system market share worldwide and is used on 35% of all computing devices worldwide

It only comes naturally that apps, games, and other software come to windows first then to other platforms.

Relation to programming

Operating systems are a collection of files and instructions created by programmers.

Programmers are always hard at work improving the performance of this code. Typically, Linux, windows and MacOS get annual or Biannual updates, improving the stability, security, and performance of these programs.

Other developers develop APIs (application programming interfaces) which are used by application programmers to connect to the operating system’s services without having to code those modules from scratch. Android makes use of APIs which is used by programmers to access operating system functionality without having to code the functionality from scratch. This is also true for a lot of other operating systems, but it is particularly noteworthy on android. Windows, Linux, MacOS and all other operating systems have their own APIs.

These are particularly important in proprietary operating systems where developers cannot access the underlying code, even if they wanted to code the operating system modules from scratch.

References

- Afteracademy.com. (2019). What is Kernel in Operating System and what are the various types of Kernel? [online] Available at: https://afteracademy.com/blog/what-is-kernel-in-operating-system-and-what-are-the-various-types-of-kernel
- Anderson, T. (2020). Linux in 2020: 27.8 million lines of code in the kernel, 1.3 million in systemd. [online] Theregister.com. Available at: https://www.theregister.com/2020/01/06/linux_2020_kernel_systemd_code
- GeeksforGeeks. (2020). Device Driver and It's Purpose - GeeksforGeeks. [online] Available at: https://www.geeksforgeeks.org/device-driver-and-its-purpose/
- Information Processing Cycle (with diagram) | Computer Application (2020). Available at: https://planningtank.com/computer-applications/information-processing-cycle
- Larger, L., Soriano, M.C., Brunner, D., Appeltant, L., Gutierrez, J.M., Pesquera, L., Mirasso, C.R. and Fischer, I. (2012). Photonic information processing beyond Turing: an optoelectronic implementation of reservoir computing. Optics Express, [online] 20(3), p.3241. Available at: https://www.osapublishing.org/oe/fulltext.cfm?uri=oe-20-3-3241&id=227221
- Sciencedirect.com. (2011). Kernel and Operating System - an overview | ScienceDirect Topics. [online] Available at: https://www.sciencedirect.com/topics/computer-science/kernel-and-operating-system
- StatCounter Global Stats. (2020). Desktop Operating System Market Share Worldwide | StatCounter Global Stats. [online] Available at: https://gs.statcounter.com/os-market-share/desktop/worldwide
- StatCounter Global Stats. (2020). Operating System Market Share Worldwide | StatCounter Global Stats. [online] Available at: https://gs.statcounter.com/os-market-share
- Techterms.com. (2020). API. [online] Available at: https://techterms.com/definition/api#:~:text=Stands for %22Application Programming Interface,interact with an external system.&text=Operating system APIs are typically, kit for the corresponding program.
- Tldp.org. (2020). [online] Available at: https://tldp.org/LDP/tlk/dd/drivers.
- What is Information Processing? - Definition from Techopedia (2020). Available at: https://www.techopedia.com/definition/25605/information-processing