Operating Systems

Introduction

Operating System is the main **program** that runs on a computer. It communicates with the hardware. It’s the **bridge** between the hardware and all other software programs. Widely used operating systems on computers are Windows, MacOS, Ubuntu but on mobile phones Android, iOS. Mobile phones and computers have different operating systems because their resources, operations, inputs and outputs are different.

| **https://docs.google.com/uc?id=1CPU5tKlt4ao0u3jAVn-teW1Df0zmpOFV** |
| --- |
| *Some Operating System Symbols* |

The operating system **controls and monitors the system activities, allocates and assigns system resources and schedules the operations**. Such as:

* Detecting an input from the keyboard,
* Sending output to the monitor or tracking [files](https://lms.clarusway.com/mod/lesson/view.php?id=1052) or folders in the storage area,
* Controlling peripheral devices such as disk drives and printers,
* Running different programs at the same time and not letting interfere with each other,
* Checking users if they are authorized or not are some examples of controlling and monitoring system activities.

Operating System allocates and assigns the resources like **memory space, disks, input/output devices and CPU time** to the application programs. At any time during runtime, there can be conflicting requests for a resource. Operating System has to deal with all these requests, define a priority list and return either a positive or a negative response. While using system resources Operating System manage them avoiding **deadlock**. Deadlock is a situation when two or more processes wait for each other to finish and none of them ever finish.

| **https://docs.google.com/uc?id=1OUiELIhZOBvqwllXIa1rPR5HE2QhJYVl** |
| --- |
| *Operating System is between hardware and software* |

Every operating system has a **kernel**. It connects applications to the actual processing of data. It also manages all communications between software and [hardware components](https://lms.clarusway.com/mod/lesson/view.php?id=44) to ensure usability and reliability.

Q: What is Operating System?  
A: Operating System is a software program that enables the computer hardware to communicate and operate with the computer software

 - Interview Q&A

Q: What is a deadlock in operating systems?  
A: Deadlock is a situation when two or more processes wait for each other to finish and none of them ever finish. Consider an example when two trains are coming toward each other on same track and there is only one track, none of the trains can move once they are in front of each other. A similar situation occurs in operating systems when there are two or more processes hold some resources and wait for resources held by other(s).

Scheduling Operations

Operating Systems use three types of operations to make use of the resources efficiently. These are

* Multiprogramming,
* Multiprocessing,
* Multithreading.

In multiprogramming, different programs run simultaneously and share the same CPU. For example while playing a game, you can listen to music in the background.

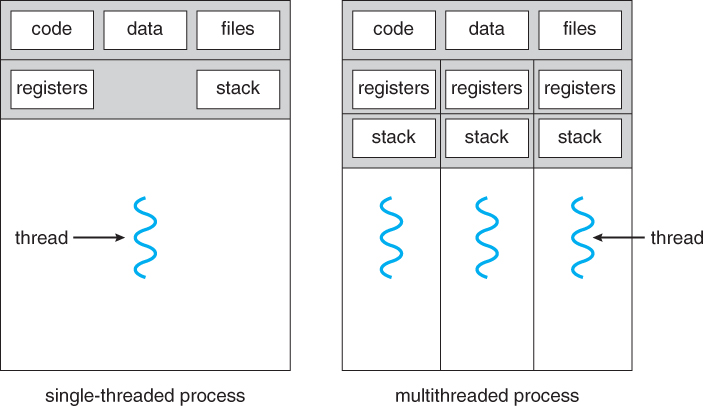
Most of the time the CPU stays idle because all other components are slower than CPU. So the multiprogramming operation takes this as an advantage and gives the opportunity to **run programs at the same time**. But this doesn’t mean that a CPU makes double or triple processes at the same time unless the computer has more than one cores. In this situation, the CPU just **switches between the programs**.

Multiprocessing and multithreading both are ways to achieve multi tasking. An example of multi tasking for houswife:

| **https://docs.google.com/uc?id=1f0Zxnqdvpk2WdyPdq--MdUlUxGcR07VT** |
| --- |
| *Multi tasking example* |

Multiprocessing is **using multiple CPUs** for running different programs at the same time. You can imagine that hands are different CPUs and while you are writing paper with the right hand if you can use your phone with the left hand you are doing multi-process. But if you are doing different works with the same hand(CPU) but with different fingers (threads) you are doing multi-threading.

Multithreading operation allows a **single program to run different tasks at the same time**. For example, while subscribing to a website, clicking the submit button would cause you to wait until the confirmation e-mail is sent to your mailbox if it is not sent on a separate thread. In a normal case, the confirmation e-mail is sent over a separate thread and you continue to interact with the page properly.



Q: What is the difference between multiprocess and multithread?  
A: Multiprocessing uses different CPU and memories so they used different sources but multithreading use same resources such as memory.

## **Types of Operating Systems**

### Different Types of OS

For general-purpose computers, there are three popular [operating systems](https://lms.clarusway.com/mod/lesson/view.php?id=56" \o "Operating Systems). These are Microsoft Windows, Mac OS and Linux in computers but most of us interact with one more operating system nowadays which is our smart phone's operating system. The most used two mobile [operating systems](https://lms.clarusway.com/mod/lesson/view.php?id=56" \o "Operating Systems) are Android and iOS.

These [operating systems](https://lms.clarusway.com/mod/lesson/view.php?id=56" \o "Operating Systems) all get the same jobs done (browsing the web, editing documents, playing games, etc), but they approach the problem in a technically different way.

Below are the most popular [operating systems](https://lms.clarusway.com/mod/lesson/view.php?id=56" \o "Operating Systems):

* UNIX
  + Linux
    - Android
      * Debian
      * Ubuntu
    - RHEL
  + MacOS / Darwin
  + FreeBSD
* Windows

| **https://docs.google.com/uc?id=1aYGIsC7l7GZ_CzsU31UXFBYHJ2214olW** |
| --- |
| Different Types of [*Operating Systems*](https://lms.clarusway.com/mod/lesson/view.php?id=56) |

Simply Linux, MacOS, and Windows separate each other. Linux is different from Windows in that it is a Unix-like OS, and Free & [Open Source](https://lms.clarusway.com/mod/lesson/view.php?id=860). It is different from MacOS in that it is Free & [Open Source](https://lms.clarusway.com/mod/lesson/view.php?id=860) but both are Unix-like OS.

Q: Which operating system/systems do you use and have you ever used [open source](https://lms.clarusway.com/mod/lesson/view.php?id=860" \o "Open Source) operating system?  
A: (You should know which operating system do you use with its version) I am using Ubuntu 19.10/Windows 10.1. I have used OSX last year. Linux [operating systems](https://lms.clarusway.com/mod/lesson/view.php?id=56" \o "Operating Systems) are [open source](https://lms.clarusway.com/mod/lesson/view.php?id=860" \o "Open Source) and I am using Linux-Ubuntu operating system. I am also using Android/iOS operating system on my mobile phone. Android operating system is also [open source](https://lms.clarusway.com/mod/lesson/view.php?id=860" \o "Open Source).

## **Open Source**

### Open Source and Free Source

Open Source Software is a type of computer software in which source code is released under a license in which the copyright holder grants users the rights to study, change, and distribute the software to anyone and for any purpose. Open-source software may be developed in a collaborative public manner. Open-source software is a prominent example of open collaboration.

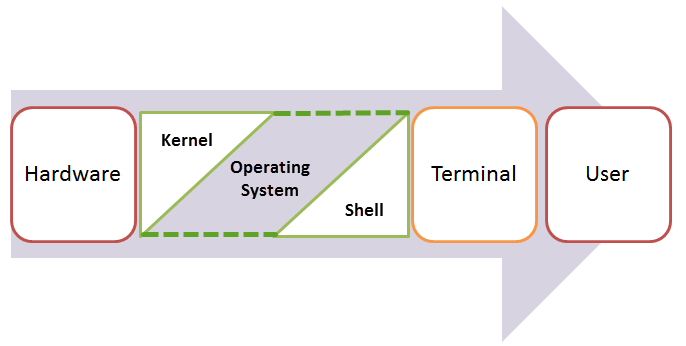
Another Software type is Free Software. **Free software** is the software that grants the user the freedom to share, study, and modify it. The word of Free comes from freedom not price. If the software is a free source you can do whatever you want with the software.

| **https://docs.google.com/uc?id=1eEFCkoIUX5RUyzhf7in6wbHaEm6z5gMQ** |
| --- |
| GNU and Linux Symbols |

Terminal Terminologies

The **Terminal** is an interface that allows users to access the command line. With command-line instructions, users can do almost everything within their operating system. Each line on terminal called as **command line**. Every line acts as a command on your operating system. (We will learn this topic at Linux Essentials Course)

**Prompt** is the name of the computer and user name with a dollar symbol at the command line.



**Shell** runs inside of the terminal. It is the interpreter for the command line at terminal. Command-line shells include flow control constructs to combine commands. In addition to typing commands at an interactive prompt, users can write **scripts** which is a computer language with a series of commands.

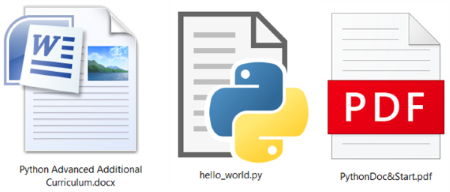
Q: What is a shell?  
A: Shell is an **interface** between the user and the kernel. Even though there can be only one kernel; a system can have many shell running simultaneously. So, whenever a user enters a command at command line from terminal, the shell communicates with the kernel to execute it and then gives the output.

What is File?

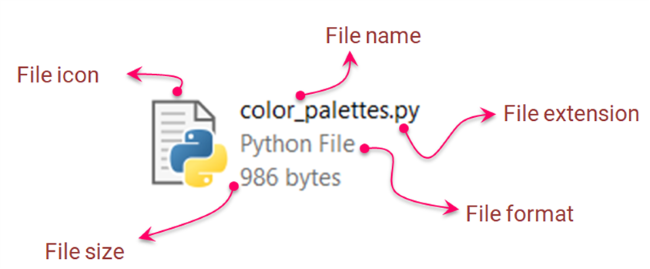
On most [operating systems](https://lms.clarusway.com/mod/lesson/view.php?id=56" \o "Operating Systems), **files** are spaces on your computer where data is stored. The many items you use on your computer in everyday life are just files. *Texts*that you type and store your codes, *pictures*and *videos*you encounter during the course or **python-3.8.1.exe** that you should have downloaded to your computer for installation are the files.

Think about it: Just as words can be written to paper, so can information be written to a computer file. You can open these files and modify their contents at any time, store them on various media, (such as hard drives, removable disks, or the cloud), or transfer them from one media to another.

In an operating system's graphical user interface, files are often represented by **icons**: thumbnails showing the content. Although it differs in all OS, files have their own icons. Let's take a look at the Windows OS's some file icons. For example, here are MS Office Word, Python and PDF files.



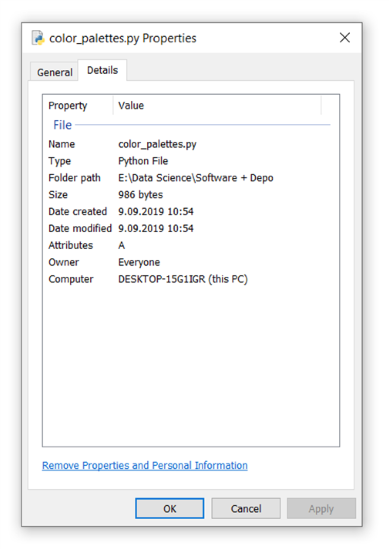
You can examine the basic information of a file in detail below :



### File Components

#### **Metadata**

All the detailed information about a *file*stored on the computer is called the **metadata** of that file. As an example, take a look at the following image. For instance, in Windows 10, you can right-click on a *file*and then click *properties* then choose *details* to display the **metadata** :



In general, almost all modern [operating systems](https://lms.clarusway.com/mod/lesson/view.php?id=56" \o "Operating Systems) have this (above image) descriptive information as **metadata**. Every time you modify anything about the file, the metadata will be updated accordingly.

#### **File Attributes**

The other point that would be useful to know about the files is the **file attributes**. It's a special kind of metadata. Each file has several *attributes*, but these *attributes* vary greatly from one **OS**to another.

In other words, **File attributes** are metadata associated with computer files that define file system behavior. Each attribute can have one of two states: **set** and **cleared**.

For example, let’s look at the following *attributes*: these basic file attributes are contained in every Windows file.

* **Read-Only**(R): When set, it makes the file read-only and indicates that a file can not be altered.
* **System** (S): When set, it indicates that the hosting file is a critical system file that is necessary for the computer to operate properly.
* **Hidden** (H): When set, it makes the file hidden and indicates that the hosting file is unseen by default.

Directories & Folders

There is a structure for storing the files in an orderly manner, just like the modules, [packages](https://lms.clarusway.com/mod/lesson/view.php?id=919" \o "Packages), and subpackages and making them easily accessible on demand. This organized structure where files are stored is called a **folder** or **directory**.

What we call a *directory* or a *folder* is actually managed in the same logic as a folder for storing documents and files in a real office environment. If you look at the folder icons used by the [operating systems](https://lms.clarusway.com/mod/lesson/view.php?id=56" \o "Operating Systems) shown below (Windows, Linux, Mac, respectively), you can see this clearly.

| **https://docs.google.com/uc?id=14PT344FTFNgDQPRuW6s3T9KmIt-DhgYe** |
| --- |
| *Folder Icons* |

A folder/directory can contain files or other folders/directories. A directory inside the other directory is generally known as a **subdirectory**.

In all [operating systems](https://lms.clarusway.com/mod/lesson/view.php?id=56" \o "Operating Systems), directories/folders are structured in a hierarchical order. There is usually one (or multiple if it is partitioned) directory called the **root directory**, and there are as many subdirectories as needed inside the root directory.

Naming the root directory may vary from one operating system to another. The following diagram shows us a sample of the hierarchical structure of the directories of the Windows OS.

| **https://docs.google.com/uc?id=1PNXf6V_9NH_8NIU3k1r3ry6P8pAEcQEM** |
| --- |
| *Windows Directories Hierarchy* |