

When programmers write programs they don't want to have to write a different program for every different hard disk, CPU and network card. All of these devices are hard to control and there are many manufacturers. Instead programs are written for a particular **operating system (OS)**.

Operating systems are a type of **systems software** that deals with the following features:

- **CPU management** – The operating system manages how **applications** are **executed** on one or more **CPUs**. The Operating System will divide the amount of time that each application's processes are allowed to use the CPU. This happens very fast and the user will normally believe that many applications are running at the same time even though one CPU is used. This process is known as **scheduling**.
- **Memory management** – Programs are stored in **pages** of memory. The OS makes sure that other programs can't access them, for security. The OS also deals with **virtual memory** and **swapping** pages in and out of **RAM** when it is full. The programmer doesn't have to worry about this.
- **Peripheral management** – Peripheral devices are input or output (I/O) devices. The operating system will have **device drivers** that communicate directly with the hardware for these devices. Device drivers may be prebuilt in the operating system or custom-built drivers that come with the hardware when it is bought. If the application needs to access devices such as the hard disk, printer or network it will do this through the operating system. This is how the same printer or internet connection can be shared by many different applications.
- **Security** – The operating system is responsible for **user management** tasks such as administering **user accounts** and **passwords** along with what each user has permission to do. For example, on a home computer you will be allowed to install software, but this is unlikely to be the case on a company or school computer. Other security features may include preventing certain applications from being run if the user doesn't have permission, or preventing access to certain files or folders for some users.
- **Applications** – When an icon is clicked or a command entered to launch an application, the operating system will carry out a number of processes: 1. Check the user is allowed to run the application; 2. Allocate an area RAM for the program and data to be stored in; 3. Copy the program to RAM; 4. Begin executing the program.

There are many different operating systems. The most common ones are **Microsoft Windows**, **Apple's Mac OS** and **iOS**, **Linux** (a type of **Unix** operating system) and **Google Android** (based on the Linux kernel). Operating systems and the type of hardware they run on can be referred to as **platforms**.

At the centre of an operating system is the **kernel**. This deals with low-level operations such as memory management.

Different programs run different **processes**. A **multi-tasking**

operating system will be able to run different processes at the same time on one CPU. The operating system does this by giving each process a short amount of CPU time to run. This is called **process management**. A process will have three states it can be in: **running**, **waiting** (to be run while another process is running) and **blocked** (waiting for something to happen, e.g. a key press).

Around the kernel sits the **shell**. This contains the **graphical user interface (GUI)**, which contains **icons**, **windows**, **menus** and the **mouse pointer**. It may also contain a text-based **interface** known as the **command line** or **terminal**.

