Software

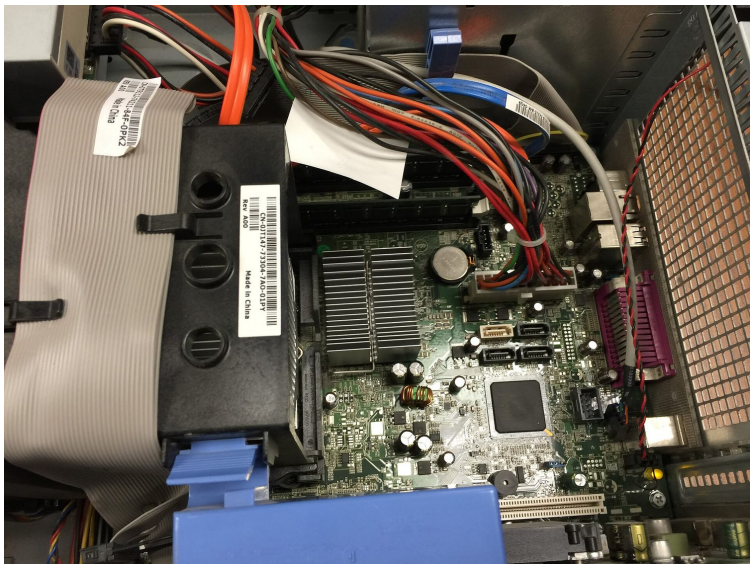
IGCSE Computer Science

Hardware and software

A computer system is made of two parts that work together.

- **Hardware** 硬件 is the **physical** 物理的 parts of the computer —the parts you can touch. Examples: the CPU, memory, keyboard, screen and hard disk.
- **Software** 软件 is the **programs** 程序 that control the computer and tell the hardware what to do.

Hardware cannot do anything useful on its own. Software needs hardware to run on. Both are needed.



The motherboard and the parts attached to it are hardware —the physical parts you can touch

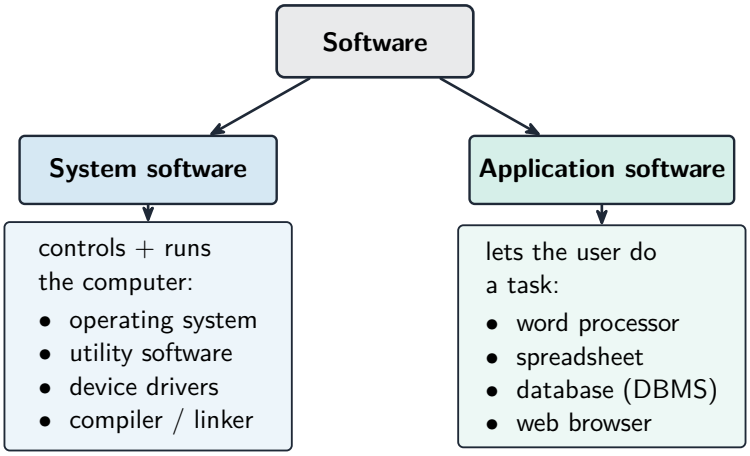
Image: Kurt Kaiser, CC0 (commons.wikimedia.org)

An operating system is software —the programs that tell the hardware what to do

Image: C-DAC/NRCFOSS, GPL (commons.wikimedia.org)

Types of software

Software is split into two types.



Software divides into system software (runs the computer) and application software (lets the user do a task)

System software

System software 系统软件 controls the computer itself and gives a base for other programs to run on. Examples:

- **operating system** 操作系统 (see below);
- **compiler** 编译器 and **linker** 链接器 (tools that turn programs into a form the computer can run);
- **device driver** 设备驱动程序—software that lets the operating system control a piece of hardware;

- **utility software** 实用程序—small tools that look after the computer (for example anti-virus, backup, disk clean-up).

Application software

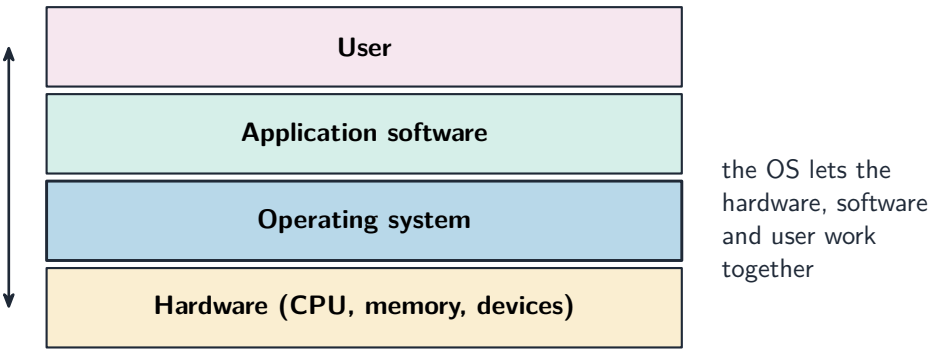
Application software 应用软件 lets the user do a useful task. Examples:

- **word processing** 文字处理 software (writing documents);
- **spreadsheet** 电子表格 software (working with numbers in a grid);
- **database management system** 数据库管理系统 (storing and searching data);
- **web browser** 网页浏览器 (viewing web pages).

	System software	**Application software**
Job	runs and manages the computer	helps the user do a task
Used by	the computer (in the background)	the user (directly)
Examples	operating system, drivers, utilities	word processor, browser, games

The operating system

An **operating system** (OS) is the main system software. It controls the whole computer and lets the hardware, the application software and the user work together. Without an OS the computer is very hard to use.



The operating system sits between the hardware and the application software and user, letting them work together

The OS has many roles:

- **managing files** —saving, opening, naming, moving and deleting files;
- **handling interrupts** (see below);
- **providing a user interface** 用户界面 so the user can control the computer;
- **managing memory** —deciding what is kept in RAM and where;
- **managing peripherals** 外围设备 and their drivers —controlling devices like printers;
- **managing multitasking** 多任务处理—letting several programs run at once;
- **providing a platform** 平台 for running application software;
- **managing security** —user accounts, passwords and access rights.

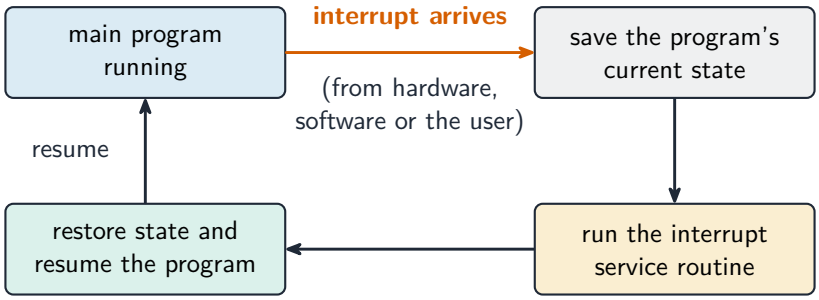
Interrupts

An **interrupt** 中断 is a signal sent to the **processor** 处理器 to tell it that something needs attention now.

An interrupt can come from three sources:

- **hardware** —for example a key is pressed, or a printer runs out of paper;
- **software** —for example a program tries to divide by zero;
- **the user** —for example the user presses a ”cancel” key.

When an interrupt arrives, the processor **stops** what it is doing, saves its place, and **services** (deals with) the interrupt. When that is done, it goes back to what it was doing before. This lets the computer react quickly to important events.



On an interrupt the CPU saves its place, runs the interrupt service routine, then restores its place and carries on

Programming languages

All computers process data using the **central processing unit** 中央处理器 (CPU). To make the CPU do work, people write **programs** in a **programming language** 编程语言. There are two kinds.

High-level languages

A **high-level language** 高级语言 is written in words that look a bit like English (for example Python). It is:

- **easy** for people to read, write and fix;
- **independent of the computer** —the same program can run on different types of computer.

Low-level languages

A **low-level language** 低级语言 is close to what the hardware actually uses. It includes:

- **machine code** 机器码—instructions written in binary (0s and 1s), which the CPU runs directly;
- **assembly language** 汇编语言—machine code written with short word codes (mnemonics) instead of binary.

A low-level language relates to the **specific** 特定的 type of CPU, so a program written for one CPU may not run on another. It is used when the programmer needs full control of the hardware or very fast, small code.

Translators

The CPU can only run machine code. A high-level program must first be changed into machine code. The software that does this is a **translator** 翻译程序. There are two types.

Compiler

A **compiler** translates the **whole** program into machine code **before** it is run. After translating, the machine code can be run many times without translating again.

- translation happens once, in full;
- it reports all the errors together, at the end of translating;
- the finished program runs fast and does not need the compiler to run.

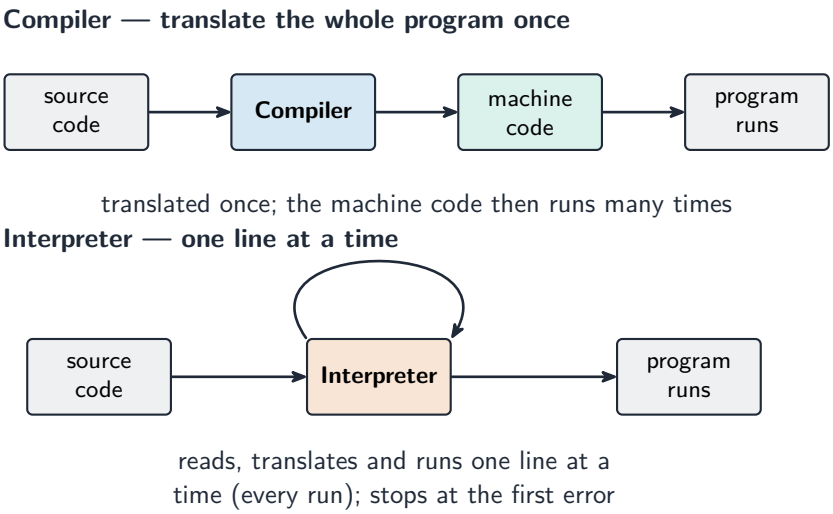
Interpreter

An **interpreter** 解释器 translates and runs the program **one line at a time**.

- it stops at the **first error** it finds, which helps when testing;
- the program runs more slowly, because it is translated each time it runs;
- the interpreter must be present every time the program runs.

	Compiler	**Interpreter**
Translates	the whole program at once	one line at a time
Errors	all reported at the end	stops at the first error
Speed of finished program	faster	slower
Needed to run later?	no	yes

Use a **compiler** when you want to share a fast, finished program. Use an **interpreter** when you are writing and testing a program and want to find errors easily.



A compiler translates the whole program once into machine code; an interpreter translates and runs one line at a time

Integrated development environment (IDE)

An **integrated development environment** 集成开发环境 (IDE) is software that helps you write programs. It puts many useful tools in one place:

- a **code editor** 代码编辑器—for typing in your program;
- a **run time environment** 运行时环境—for running the program to test it;
- a **translator** —a compiler or interpreter to turn your code into machine code;
- **error diagnostics** 错误诊断—messages that help you find and understand mistakes;
- **auto-completion** 自动补全 and **auto-correction** 自动纠错—the IDE finishes or fixes code as you type;
- **prettyprinting** 美化打印—laying out the code neatly with colour and indentation so it is easy to read.