

5 System Software – Part 1

Name: _____ Class: _____ Date: _____

Vocabulary

Write each word three times.

English	Write it 3 times (English)
operating system	_____
process	_____
virtual memory	_____
device driver	_____
utility program	_____
translator	_____
compiler	_____
interpreter	_____
assembler	_____

Recall

At IGCSE you used computers every day. Bring this with you:

- The operating system is the main software that runs the machine.
- You ran programs (applications) on top of it.

At AS we list what the operating system does, and meet the programs that translate code – the software that makes all other software possible.

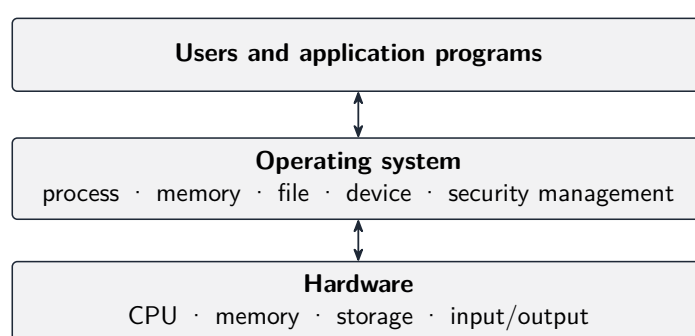
New ideas

Read these first. Each idea has a short worked example. Then do the Practice.

■ 1 The operating system

The **operating system** manages the computer's hardware and software so that programs can run. Its jobs include:

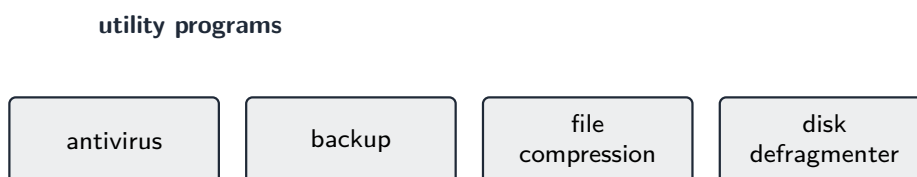
- managing each running program (a **process**);
- managing memory, including **virtual memory** (using disk space as extra RAM when memory is full);
- talking to each piece of hardware through a **device driver**.



Worked example. When you open a photo editor, the OS gives it memory, loads it from disk, and lets it use the screen and mouse through device drivers – the program never has to know the hardware details itself.

■ 2 Utility programs

A **utility program** is a small tool that keeps the system healthy – antivirus, backup and file compression are common examples.

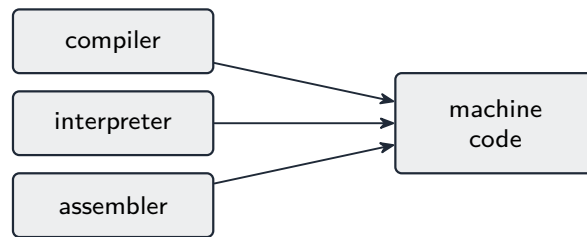


Worked example. A backup utility copies your files to another disk on a schedule, so a crash or theft does not lose your work.

■ 3 Translators

Code you write must be turned into machine code by a **translator**:

- a **compiler** translates the whole program at once, then it can run on its own;
- an **interpreter** translates and runs one line at a time;
- an **assembler** turns assembly language into machine code.



Worked example. Python usually uses an interpreter (runs line by line, easy to test as you go); C uses a compiler (translated once into a fast program that runs on its own).

Watch out: an interpreter does *not* leave behind a machine-code file you can keep – it translates the code again every time the program runs.

Practice

5.1 Give one example of a utility program. [1]

5.2 State what an assembler translates, and into what. [1]

5.3 State two jobs done by the operating system. [2]

5.4 Explain what virtual memory is. [2]

5.5 State one difference between a compiler and an interpreter. [2]

5.6 Explain one advantage of using an interpreter while you are still writing and testing a program. [2]

5.7 Challenge. A finished program will be sent to customers who must not see the source code and who want it to run fast. Would you give them a compiled or an interpreted version? Give a reason. [2]
