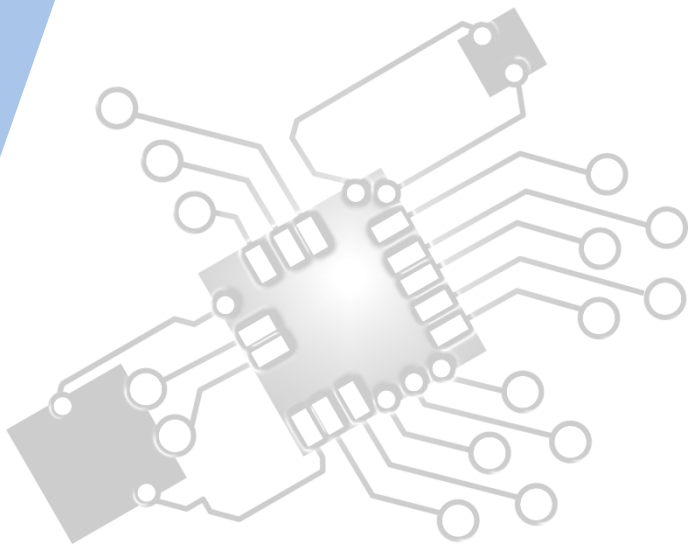
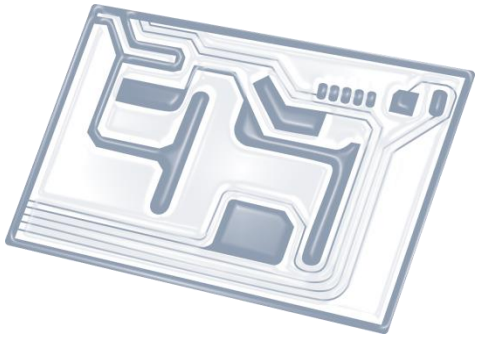
Computer Organisation

IB Computer Science



Content developed by
Dartford Grammar School
Computer Science Department

HL Topics 1-7, D1-4



1: System design

2: Computer Organisation

3: Networks

4: Computational thinking

5: Abstract data structures

6: Resource management

7: Control

D: OOP

HL & SL 2 Overview

Computer architecture

2.1.1 Outline the architecture of the central processing unit (CPU) and the functions of the arithmetic logic unit (ALU) and the control unit (CU) and the registers within the CPU

2.1.2 Describe primary memory. 2 Distinguish between random access memory (RAM) and read-only memory (ROM), and their use in primary memory

2.1.3 Explain the use of cache memory

2.1.4 Explain the machine instruction cycle

Secondary memory

2.1.5 Identify the need for persistent storage

Operating systems and application systems

2.1.6 Describe the main functions of an operating system

2.1.7 Outline the use of a range of application software

2.1.8 Identify common features of applications

Binary representation

2.1.9 Define the terms: bit, byte, binary, denary/decimal, hexadecimal

2.1.10 Outline the way in which data is represented in the computer

Simple logic gates

2.1.11 Define the Boolean operators: AND, OR, NOT, NAND, NOR and XOR

2.1.12 Construct truth tables using the above operators

2.1.13 Construct a logic diagram using AND, OR, NOT, NAND, NOR and XOR gates

1: System design

2: Computer Organisation

3: Networks

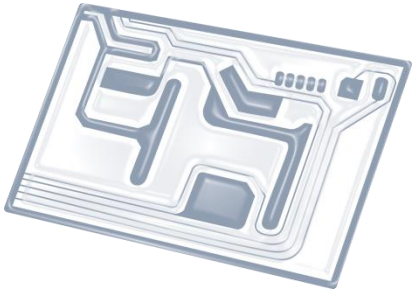
4: Computational thinking

5: Abstract data structures

6: Resource management

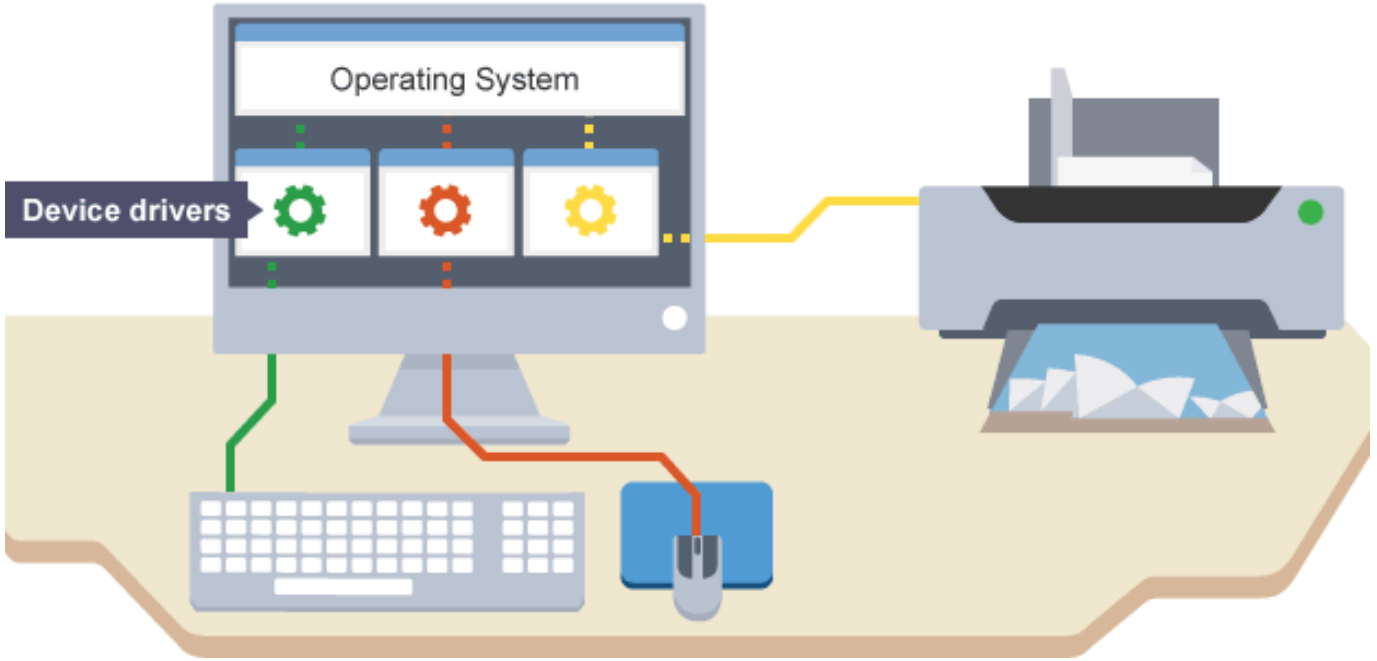
7: Control

D: OOP



Topic 2.1.6

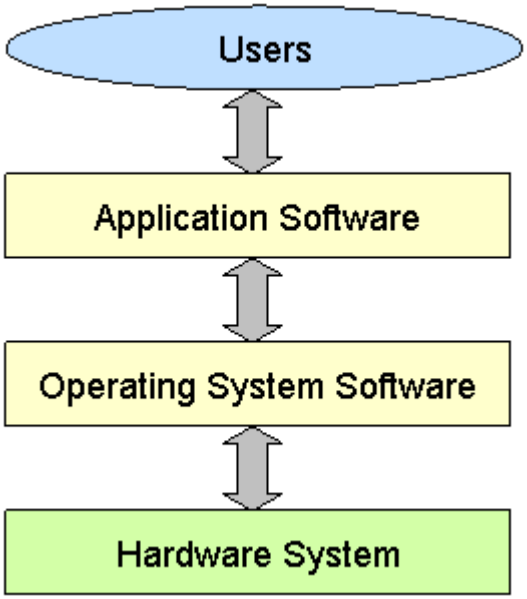
Describe the **main functions** of an **operating system**

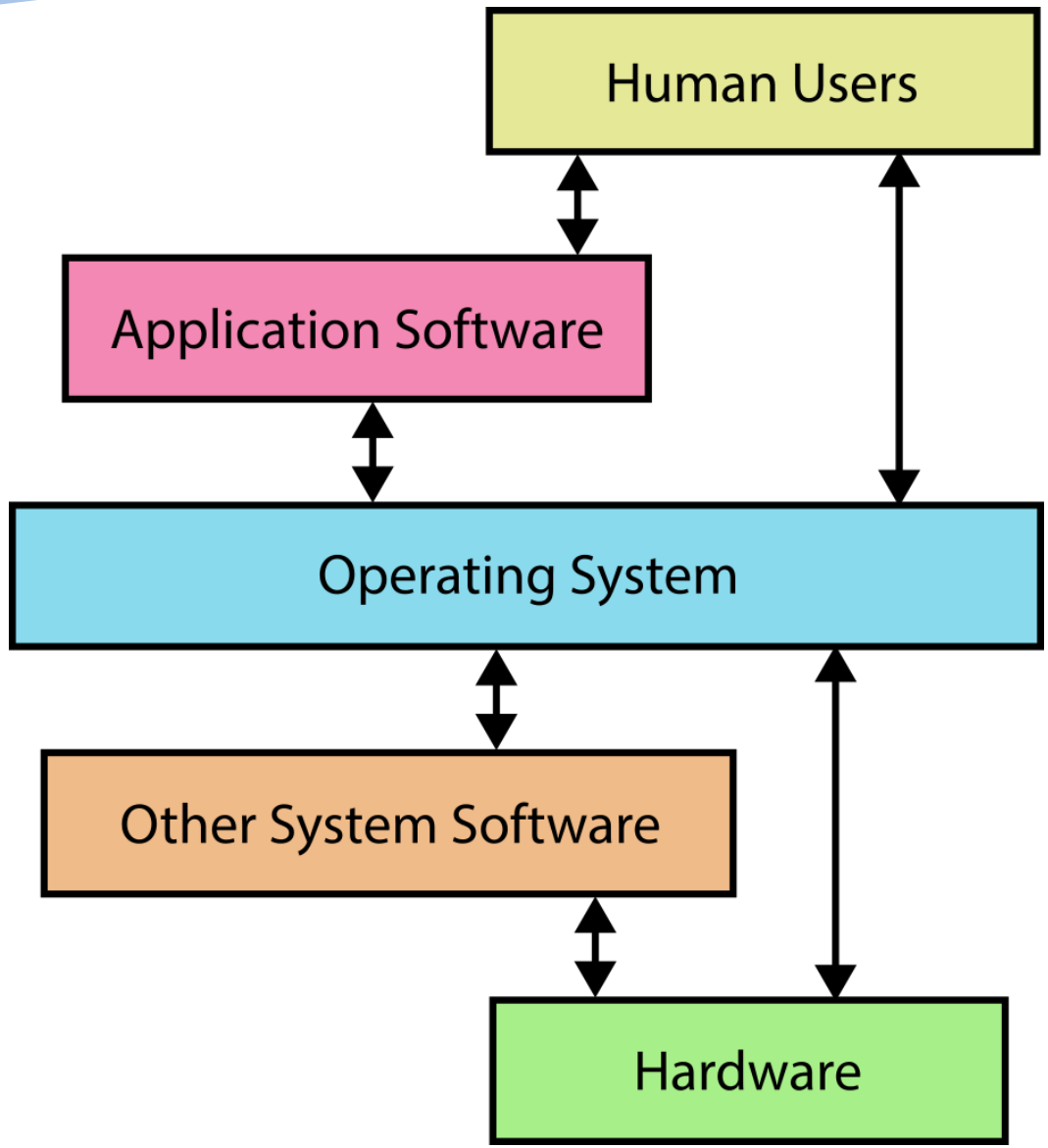


Exam note!

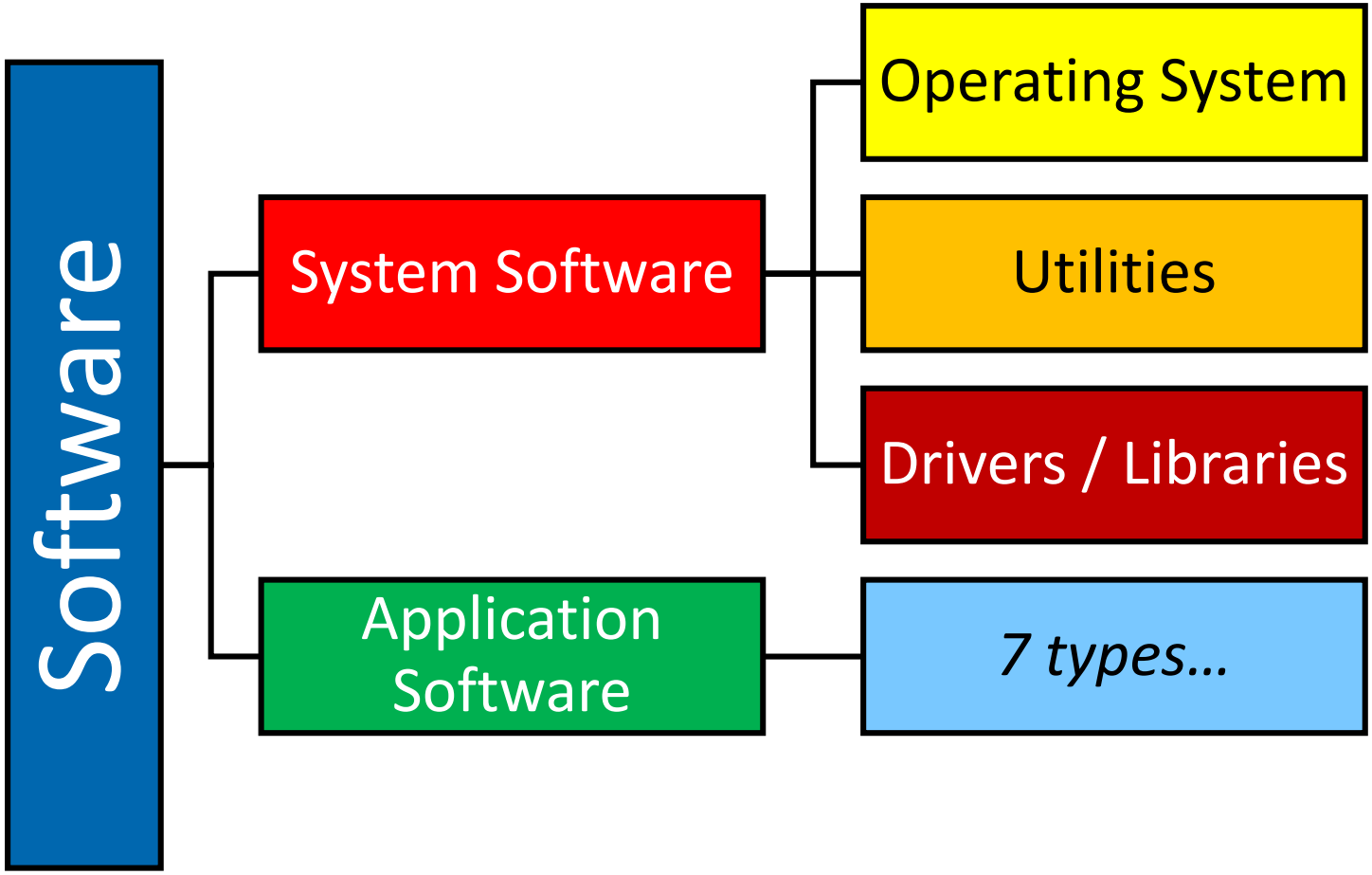
HL students: This curriculum point is repeated in much greater detail in **Topic 6: Resource Management**.

For reference, see **Curriculum Points 6.1.5-6.1.9**



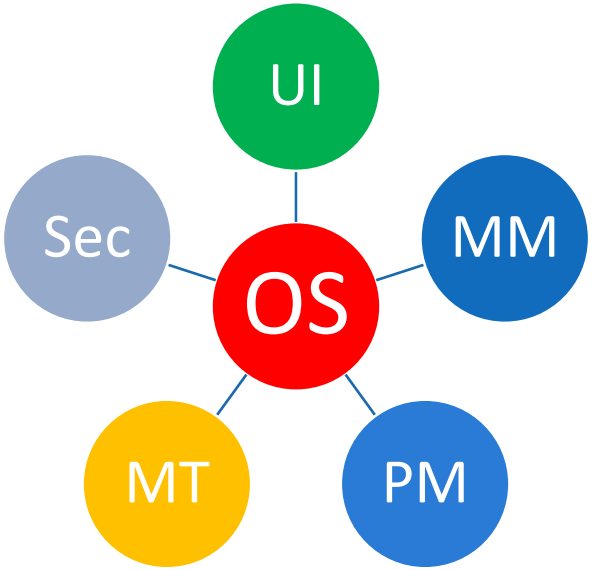

Human Users
Application Software
Operating System
Other System Software
Hardware

Hierarchy of software



Functions of an operating system

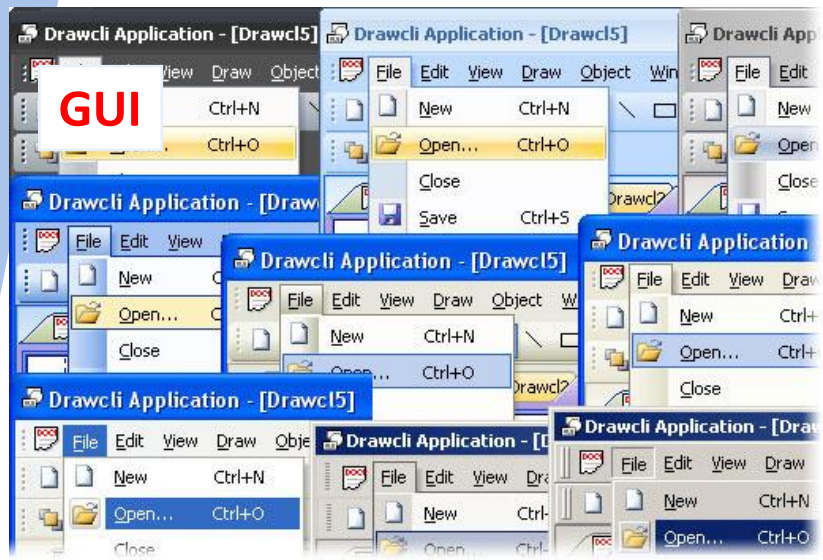
A. Provides a user interface
B. Does memory management
C. Does peripheral management
D. Allows multi-tasking
E. Provides security

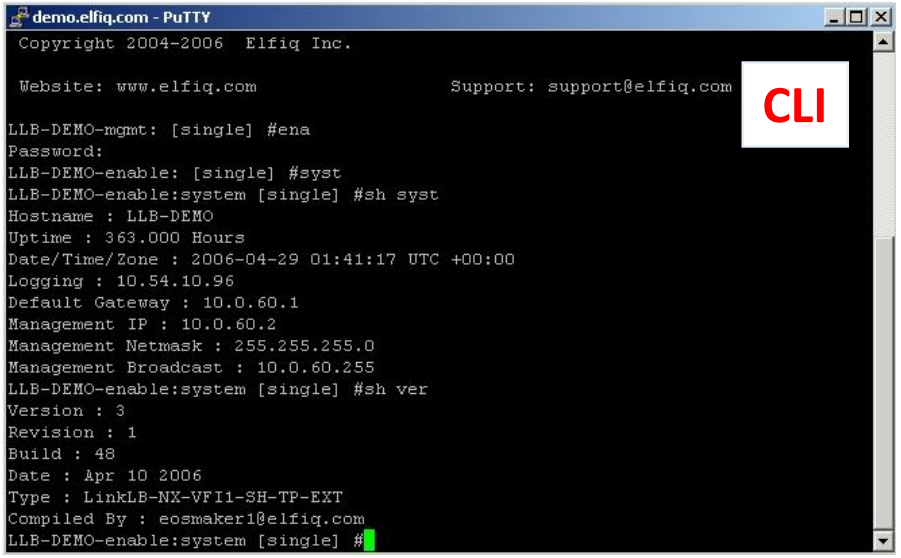


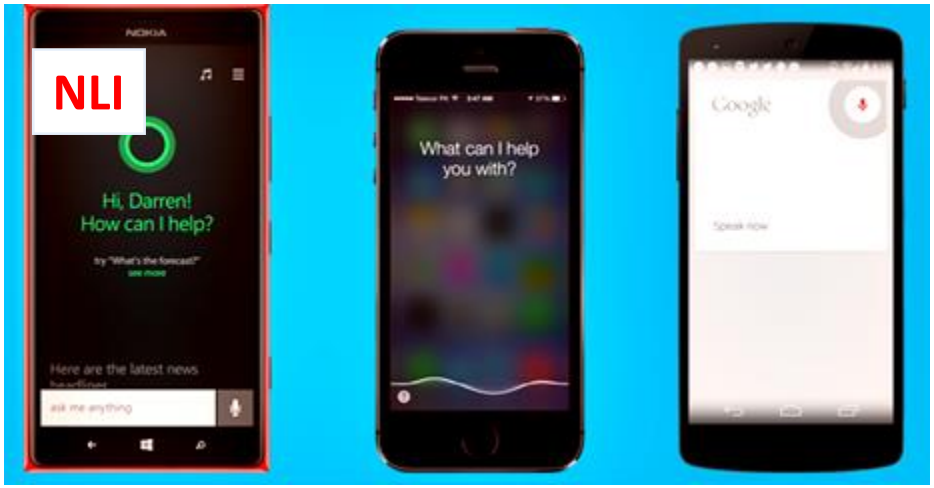
A. User interface

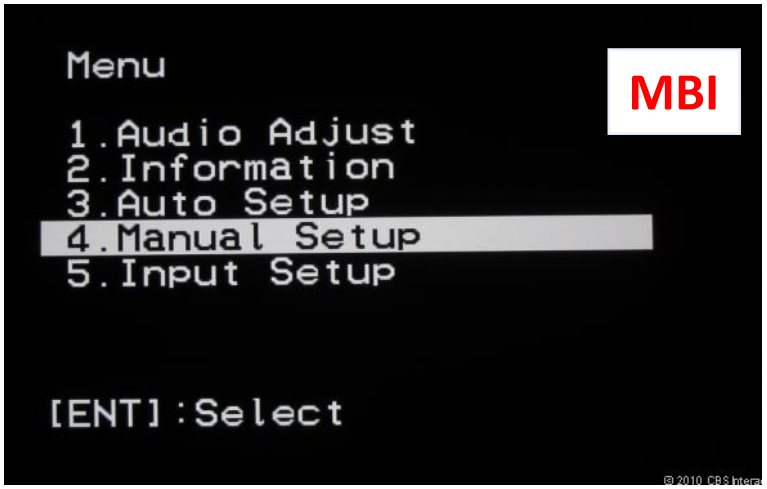
- OS has to provide **a link** between the **user** and the **computer hardware**.
- Types of user interfaces:
  - **Graphical User Interfaces** (GUIs) that have menus and icons,
  - **Command Line Interfaces** (CLIs) where the user types in codes,
  - **Natural Language Interface** (NLIs) where the user speaks to the interface
  - **Menu Based Interface** (MBIs) which gives the user a selection of options.

Example of user interfaces









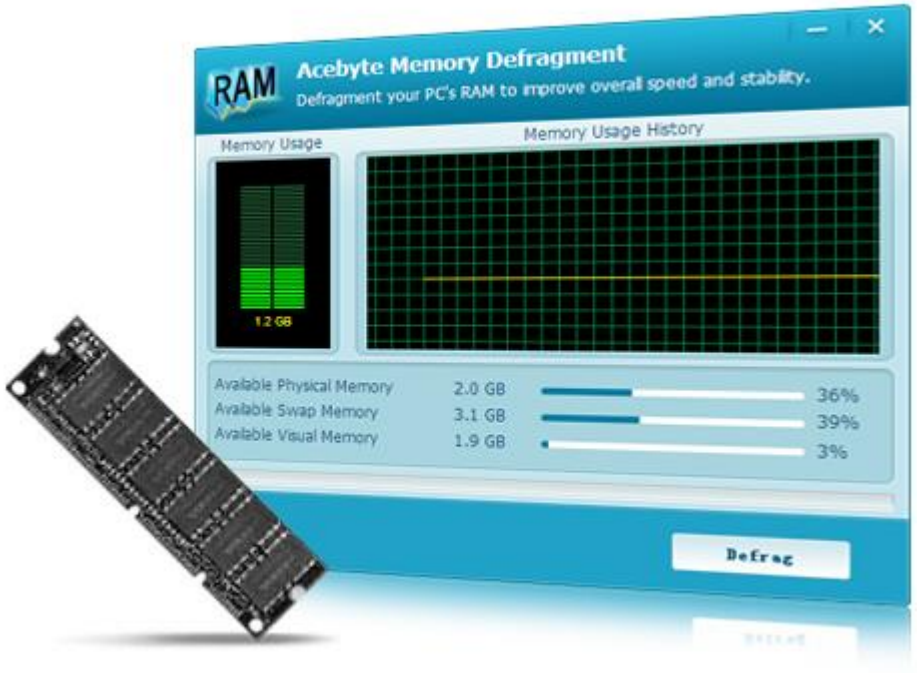
B. Memory management

- Memory management is done by keeping track of **storage devices (like HDD)** and controlling which application has access to which area of **memory (RAM)**.
- Each location in memory can be read, modified, and written to by the OS. When the memory location is full the OS sends a confirmation message.
- Similarly, the OS provides **file management** services by sorting out where data is stored on the disk drives and memory.
- The OS allows users to **organise files** in **folders** as well as to copy and delete fi les.

Example of memory management

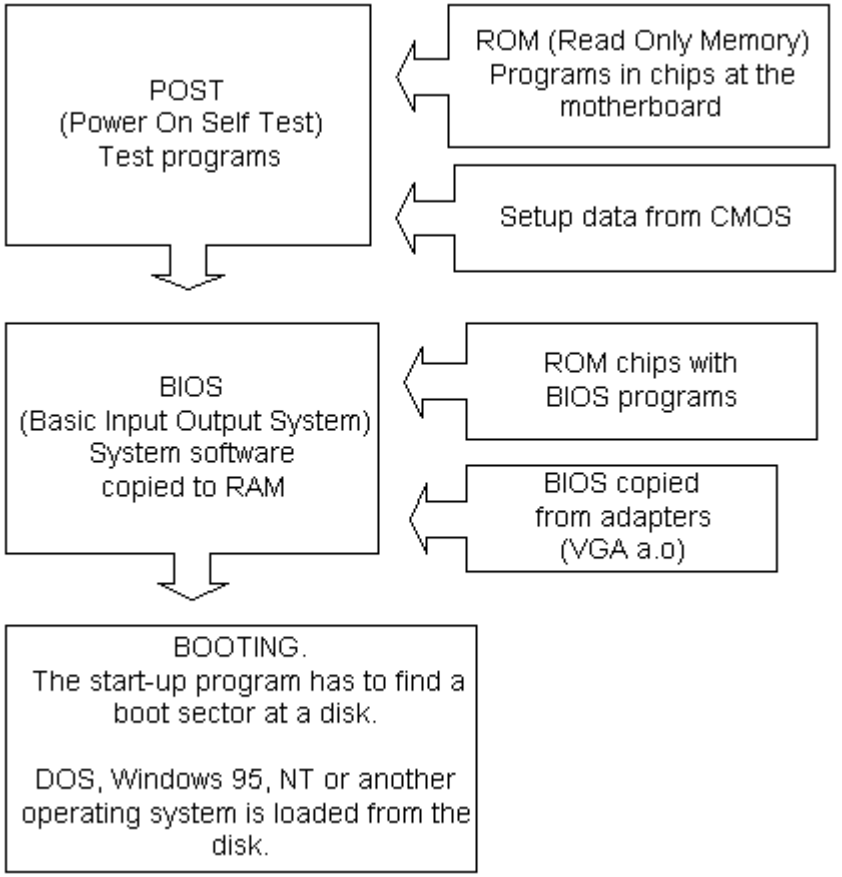
File manager for managing files on HDD

Memory manager for RAM to make sure programs don't use same memory space

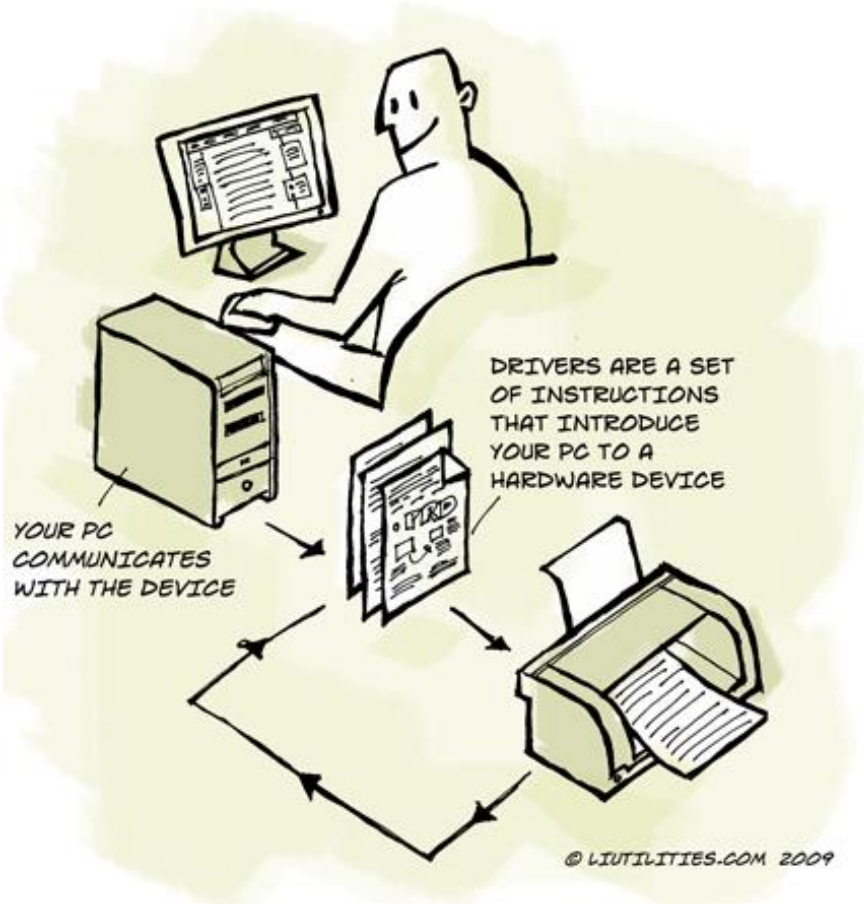


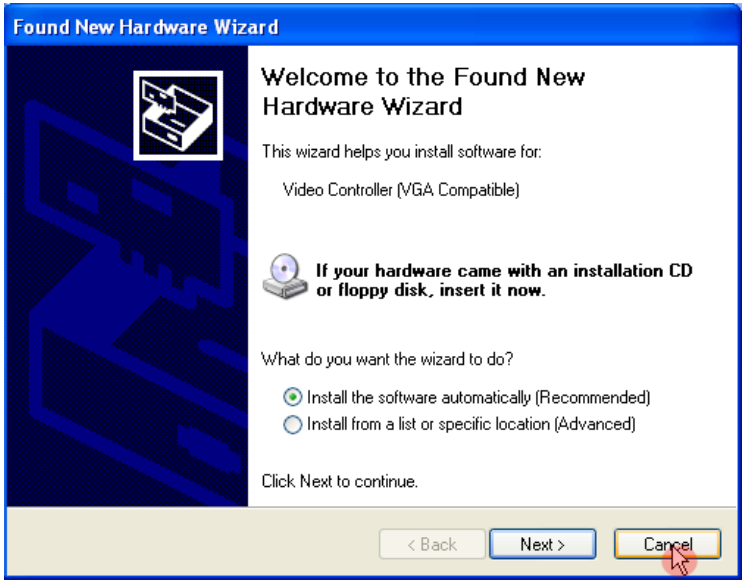
C. Peripheral management

- Keyboard, mouse, monitor and printers are controlled through **device drivers**.
- A **device driver** is a software program which allows hardware devices to be used by the OS. They act as **translators** between the devices and the computer system.



Peripheral management example

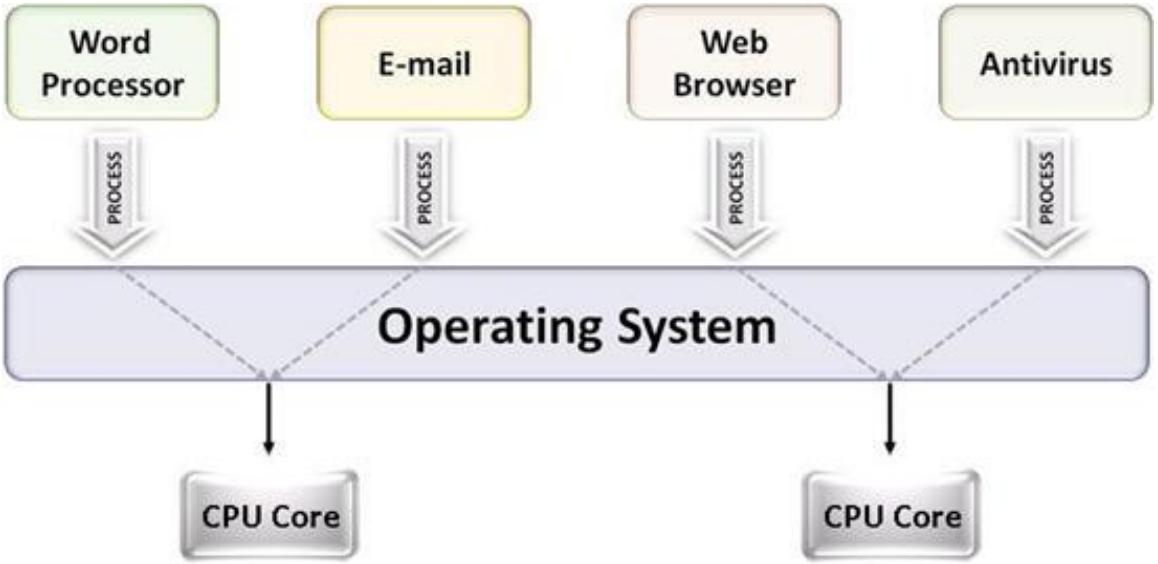




D. Multitasking

- The OS **coordinates** the working of different programs by allocating the CPU time between different programs based on **time** and **priority** of the software application.
- Each task running is given a **slice of time**, or a turn on the CPU.
- Each task has to **wait its turn** unless it is given a higher priority by the OS in which case it gets more or longer time slices.

Multitasking example



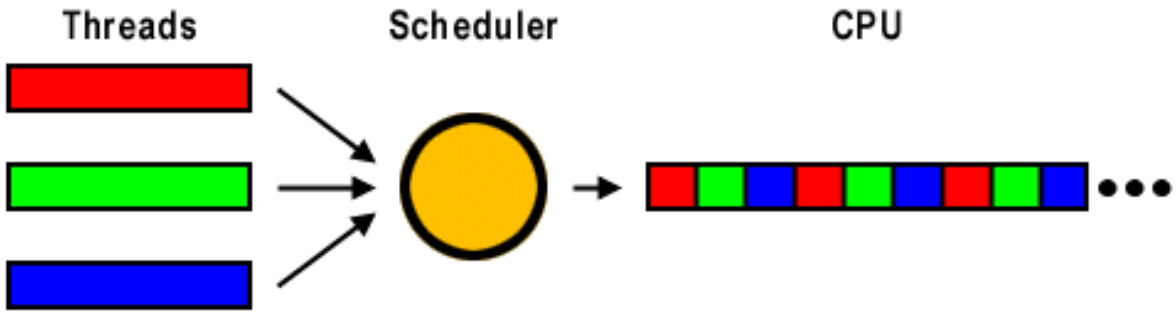
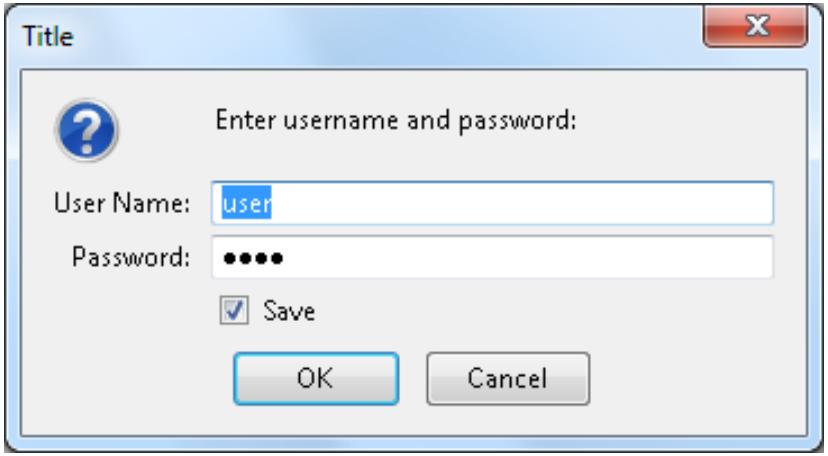


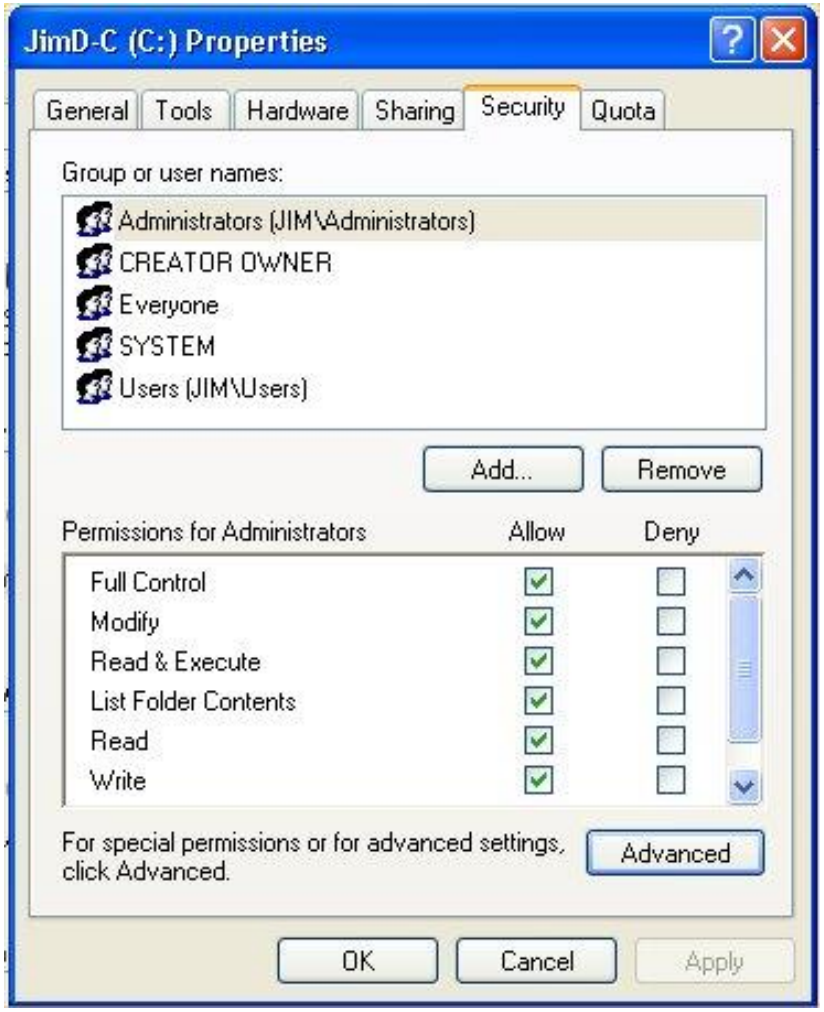
Fig. 1: Thread scheduling

E. Security

- OS **prevents unauthorised access**.
- It ensures security of the system through **usernames** and **passwords**.
- The OS **protects files** from other users reading or writing files.

Security examples





Recap: Functions of an operating system

A. Provides a user interface
B. Does memory management
C. Does peripheral management
D. Allows multi-tasking
E. Provides security