4.1 Types of software and interrupts

System software and application software

Without **software**, the **hardware** items would be useless. Software is a set of instructions that tells computer hardware what to do. It contains programs, procedures, and routines to interact with computer hardware. Fundamentally, we can categorise software into two types. They are **system software** and **application software**.



System Software

The **system software** is basically a platform that controls and manages the hardware (processors, devices, memory). It also provides an interface to run the application software.

Features:

- System software handles the hardware and system resources.
- set of programs to control and manage the operation of computer hardware
- provides a platform on which other software can run
- required to allow hardware and software to run without problems

- controls the allocation and usage of hardware resources.

Examples:

Compilers: A compiler is a computer program that translates a program written in a high-level language (HLL) into machine code (code which is understood by the computer) so that it can be directly used by a computer to perform a required task. The original program is called the source code and the code after compilation is called the object code. Once a program is compiled, the machine code can be used again and again to perform the same task without re-compilation. Examples of high-level languages include: Java, Python, Visual Basic, Fortran, C++ and Algol.

Linkers: A linker is a computer program that takes one or more object files produced by a compiler and combines them into a single program which can be run on a computer. For example, many programming languages allow programmers to write different pieces of code, called modules, separately. This simplifies the programming task since it allows the program to be broken up into small, more manageable sub-tasks.

Device Drivers: A device driver is the name given to software that enables one or more hardware devices to communicate with the computer's operating system. Without drivers, a hardware device (for example, a computer printer) would be unable to work with the computer. All hardware devices connected to a computer have associated drivers. As soon as a device is plugged into the USB port of a computer, the operating system looks for the appropriate driver. An error message will be produced if it can't be found. Examples of drivers include: printers, memory sticks, mouse, CD drivers, and so on.

Operating Systems (O/S): The operating system (OS) is essentially software running in the background of a computer system. It manages many of the basic functions.
For example, operating systems allow:
• input/output operations
• users to communicate with the computer (e.g. Windows)
• error handling to take place
• memory management
• the loading and running of programs to occur
• managing of security (e.g. user accounts, log on passwords).

Application Software

Application Software is the set of programs that runs as per the user's command to fulfil the following task. It runs on the platform designed by system software.

Features:

- Application software handles user's requests and executes specific tasks.
- used to perform various applications (apps) on a computer
- allows a user to perform specific tasks using the computer's resources
- may be a single program or a suite of programs
- users can execute the software as and when they require.

Examples:

Word Processor: Word processing software is used to manipulate a text document, such as an essay or a report. Text is entered using a keyboard and the software provides tools for copying, deleting and various types of formatting. Some of the functions of word processing software include:
• creating, editing, saving and manipulating text
• copy and paste functions
• spell checkers and thesaurus
• import photos/images into a structured page format
• translation into a foreign language.

Spreadsheet: Spreadsheet software is used to organise and manipulate numerical data (in the form of integer, real, date, and so on). Some of the functions of spreadsheets include:
• use of formulas to carry out calculations
• ability to produce graphs
• ability to do modelling and “what if” calculations.

Database: Database software is used to organise, manipulate and analyse data. A typical database is made up of one or more tables. Tables consist of rows and columns. Each row is called a 'record' and each column is called a 'field.' This provides the basic structure for the organisation of the data within the database. Some of the functions include:
• ability to carry out queries on database data and produce a report
• add, delete and modify data in a table.

Control And Measuring Software: Control and measuring software is designed to allow a computer or microprocessor to interface with sensors so that it is possible to:
• measure physical quantities in the real world (such as temperatures)
• to control applications (such as a chemical process)

Apps: Apps is short for applications – a type of software. They normally refer to software which runs on mobile phones or tablets. Common examples of apps include:
• video and music streaming
• GPS (global positioning systems – help you find your way to a chosen location)
• camera facility (taking photos and storing/manipulating the images taken).

Photo Editing Software: Photo editing software allows a user to manipulate digital photographs stored on a computer; for example, change brightness, change contrast, alter colour saturation or remove “red eye”.

Video Editing Software: Video editing software is the ability to manipulate videos to produce a new video. It enables the addition of titles, colour correction and altering/adding sound to the original video.

Graphics Manipulation Software: Graphics manipulation software allows bitmap and vector images to be changed. Bitmap images are made up of pixels which contain information about image brightness and colour.

Utility software

Utility software refers to a set of programs designed to perform specific tasks that enhance the efficiency, functionality, and maintenance of a computer system. Unlike operating systems, which manage the overall operation of the system, utility software focuses on specific tasks or services.

*Some common types of **utility software** include:*

- **Virus Checkers:** Also known as antivirus software, these programs scan files, emails, and web traffic for malware, viruses, and other malicious software. They detect, isolate, and remove threats to protect the computer system from infections.

- **Defragmentation Software:** This utility rearranges fragmented files on a hard drive to optimise storage space and improve disk performance. By organising files more efficiently, defragmentation software reduces file access times and enhances overall system speed.

- **Disk Contents Analysis and Repair:** These tools analyse the contents of disks or file systems to identify errors, corruption, or inconsistencies. They can repair damaged file systems, recover lost data, and fix disk errors to ensure data integrity and system stability.

- **File Compression and File Management:** File compression utilities compress files or folders into smaller archives to save disk space and facilitate file transfer. File management tools help organise, categorise, and manipulate files and folders, making it easier to navigate and manage large volumes of data.

- **Backup Software:** Backup utilities create copies of important data and system files to protect against data loss. They typically offer features such as scheduled backups, incremental backups, and cloud storage integration to ensure that critical data is securely backed up and recoverable in the event of hardware failure or data corruption.

- **Security Software:** In addition to antivirus programs, security software includes firewalls, anti-spyware tools, and intrusion detection systems to safeguard computer systems against unauthorised access, data breaches, and cyber threats. These utilities protect sensitive information, prevent malware infections, and ensure network security.

- **Screensavers:** While primarily used for aesthetic purposes, screensavers also serve as a utility to prevent screen burn-in on CRT and OLED displays. Some screensavers may include additional features such as password protection, displaying clock or calendar information, or running animations for slideshows.

Operating System

An operating system (OS) is the software that acts as an intermediary between computer hardware and user applications. Its primary functions include:

- **Managing Files:** The OS provides a hierarchical file system for organising and storing data. It controls file operations such as creation, deletion, reading, and writing.

- **Handling Interrupts:** Interrupts are signals sent by hardware or software to the CPU, indicating the need for attention. The OS manages interrupts by prioritising them and directing the CPU to respond accordingly.

- **Providing an Interface:** The OS offers a user interface, which can be graphical (GUI) or command-line (CLI), allowing users to interact with the computer system, launch applications, and perform tasks. Managing

- **Peripherals and Drivers:** The OS communicates with peripheral devices such as printers, keyboards, and mice through device drivers. It manages these devices, handling data transfer and ensuring compatibility.

- **Managing Memory:** The OS allocates and deallocates memory resources for running processes, ensuring efficient utilisation of RAM and virtual memory. It also handles memory protection to prevent unauthorised access.

- **Managing Multitasking:** Modern operating systems support multitasking, allowing multiple processes to run concurrently. The OS schedules tasks, switches between them, and allocates CPU time based on priority and resource availability.

- **Providing a Platform for Running Applications:** The OS provides an execution environment for software applications by managing resources, facilitating communication between processes, and providing APIs (Application Programming Interfaces) for developers.

- **Providing System Security:** The OS implements security mechanisms such as user authentication, access control, encryption, and firewall protection to safeguard the system against unauthorised access, malware, and other threats.

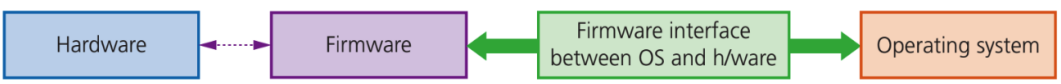
- **Managing User Accounts:** The OS maintains user accounts, each with its own set of permissions and access rights. It controls user authentication, password management, and user-specific configurations.

Running of applications

When a computer starts up, it goes through a process called booting, where the basic input/output system (BIOS) initialises hardware and loads part of the operating system into RAM. The BIOS, stored in an Electrically Erasable Programmable ROM (EEPROM), locates the operating system on a storage device and executes it.

BIOS settings are stored in a Complementary Metal Oxide Semiconductor (CMOS) chip, powered by a battery. Removing or disconnecting the battery resets these settings to factory defaults.

Once the operating system is loaded, it manages application software and system resources like device drivers. As the software runs, different parts of the operating system may be loaded in and out of RAM as needed.



Interrupts

In computer systems, an interrupt is a signal that prompts the processor to stop executing its current task and to start executing a specific routine, called an interrupt handler. Interrupts allow the operating system to perform tasks that require the processor's attention, even while the processor is executing a program. This allows for efficient multitasking, as the processor can quickly switch between tasks in response to interrupts. For example, an interrupt could be triggered by a keyboard input, a timer, or a network event, allowing the processor to respond to these events in a timely manner.

1. **How An Interrupt Is Generated:** An interrupt is generated by an **interrupt request (IRQ) signal**, which is sent by an external device, such as a keyboard, mouse, network card, or timer, to the processor. The IRQ signal is received by the **interrupt controller**, which informs the processor that an interrupt has been requested. The processor then suspends its current task and starts executing the interrupt handler, which is a routine that performs a specific action in response to the interrupt. The interrupt handler typically retrieves information about the cause of the interrupt and performs the necessary actions, such as reading data from the keyboard or sending data to a network. Once the interrupt handler has completed its task, the processor returns to its previous task, and continues executing the program. This process of interrupt handling is managed by the operating system, which controls the priorities of the interrupts and determines which interrupt handler should be executed in response to an interrupt.

2. **Interrupt Service Routine:** An interrupt is handled using an interrupt service routine (ISR), also known as an **interrupt handler**. An ISR is a program or subroutine that is executed in response to an interrupt request.

   The ISR performs the necessary actions to process the interrupt and restore the system to its previous state. The ISR is typically invoked by the operating system's interrupt handler, which is called whenever an interrupt request is received by the processor. The interrupt handler retrieves the source of the interrupt and decides which ISR should be executed.

   The ISR then performs the necessary actions, such as reading data from a device, updating data structures, or sending data to a network. Once the ISR has completed its task, it typically sends an end-of-interrupt (EOI) signal to the interrupt controller, indicating that the interrupt has been processed. The processor then returns to its previous task and continues executing the program.

   It's important to note that the ISR should be designed to execute as quickly as possible, to minimise the time that the processor is unable to execute other tasks. The ISR should also be designed to be reentrant, meaning that it can be safely invoked multiple times simultaneously, even if a previous invocation of the ISR has not yet completed.

3. **What Happens as a Result of Interrupts:** As a result of interrupts, the following things happen in a computer system:

- **Interrupt-driven multitasking:** Interrupts allow the processor to perform multiple tasks simultaneously, by quickly switching between tasks in response to interrupts. This makes it possible for the operating system to manage multiple programs or processes and provide a responsive user interface.

- **Device management:** Interrupts are used to manage input/output (I/O) operations, such as reading data from a keyboard or writing data to a disk. The interrupt handler performs the necessary actions to transfer data to and from the device, allowing the processor to continue executing the program while the I/O operations are in progress.

- **Timing control:** Interrupts can be generated by timers, allowing the operating system to keep track of time and manage the scheduling of tasks. For example, an interrupt could be generated every 1/60th of a second to update the screen display in a video game.

- **Error handling:** Interrupts can be used to detect and handle errors, such as hardware failures or invalid user input. The interrupt handler can perform the necessary actions to notify the user of the error, or to recover from the error and restore the system to a stable state.

In summary, interrupts are a crucial aspect of computer systems that allow the processor to manage multiple tasks, devices, and errors, and provide a responsive and stable computing environment.

Software Interrupts:

Software interrupts, also known as exceptions, are events that are generated by the processor itself, in response to certain conditions, such as division by zero or access to an illegal memory location. Unlike hardware interrupts, software interrupts are generated by the processor as a result of executing a program, and are handled by the operating system's interrupt handler.

- **Division by zero:** A division by zero interrupt is generated when a program attempts to divide a number by zero, which is undefined in mathematics. The operating system's interrupt handler catches this exception and typically terminates the program, or returns an error message to the user.

- **Memory access violations:** A memory access violation interrupt is generated when two processes attempt to access the same memory location simultaneously. This can cause data corruption or other unpredictable behaviour, and is prevented by the operating system's interrupt handler, which assigns exclusive access to the memory location to one of the processes.

Hardware Interrupts:

Hardware interrupts are events generated by external devices, such as a keyboard, mouse, network card, or timer, that signal the processor to temporarily halt its current task and attend to the interrupt. These interrupts are managed by the interrupt controller, which informs the processor of the interrupt request and decides which interrupt handler should be executed in response to the interrupt.

- **Keyboard interrupt:** A keyboard interrupt is generated when a key on the keyboard is pressed. The interrupt handler reads the key code from the keyboard, and performs the necessary actions, such as updating the screen display or adding the key to a buffer.

- **Mouse interrupt:** A mouse interrupt is generated when the mouse is moved or a button is pressed. The interrupt handler reads the mouse position and button state from the mouse, and performs the necessary actions, such as updating the screen display or changing the cursor position.