

System Software

1. Compilers
2. Linkers
3. Device Drivers
4. Operating system(OS)
5. Utilities

features: (provide the services that the **computer requires**)

1. Controls and manages the operation of the computer **hardware**.
2. Provides a platform on which all **software can run properly**.
3. Provides a **human computer interface** (HCI).
4. Controls the allocation and usage of **resources** (software and hardware).

Application Software

1. Spreadsheet
2. WordProcessor
3. Database
4. Editing software
5. Internet browser

features: (provide the service that the **user requires**)

1. Used to perform **various tasks** on a computer.
2. Allows the user to perform **specific tasks** on a computer.
3. Meets the **requirements of the user**.

Utility programs:

provides a number of useful programs to **protect the computer** and the user and also give the user software tools to carry out some of the **day-to-day maintenance**.

1. Virus checker
2. Defragmenter
3. Disk repair
4. File compression
5. Back-up software
6. Security
7. Device drivers
8. Screensavers

Operating System

1. **handling interrupts**
2. **managing multitasking** (carry out more than one task simultaneously)
 1. all resources are allocated specific processor time
 2. all processes can be interrupted as and when necessary
3. **managing memory**
 1. manage the data between RAM and HDD/SSD
 2. protecting two applications using the same memory at the same time
4. **providing an interface**
5. **managing system security**
 1. integrity, confidentiality and availability of data
6. **managing files**
 1. maintains file directories
 2. file name conventions
7. **managing user account**
 1. user login
 2. user's data store in separate part of memory
8. **managing peripherals and drivers**
 1. manage queues and buffers to ensure data is being handled correctly
9. **providing a platform for running applications**

Running of applications

(Basic Input Output System)**BIOS**: a type of of firmware. provide low-level control of devices.

1. BIOS load part of operating system into RAM
2. Apps Apps are under the control of the operating system and need to access system software, such as device drivers.

Interrupts

a **signal** send from a device or software to the microprocessor. cause the microprocessor to temporarily stop what it is doing so it can service the interrupt.

cause of interrupt:

1. **Timing single**
2. **Input/Output process**
3. **Hardware fault**
4. **User interaction**
5. **Software error**

examples:

1. software interrupts
 1. divide by zero error
 2. can't find .exe file
 3. two process trying to access the same memory location at the same time
2. hardware interrupts
 1. printer has paper jam
 2. printer run out of ink
3. timing single error
 1. clock issue and operations are not synchronised
4. input/output hardware process
 1. HDD requesting new data
5. user interaction
 1. pressing keyboard keys
 2. clicking a mouse button

Assembly Language

Written using mnemonics and labels.

Needs translation before execution.

Machine Code

Written in binary.

Can be executed straight away.

Operation		Opcode		Address mode		Operand	
4bits	2bits	2bits	2bits	2bits	2bits	16bits	16bits
Register addressing							

High-level language

easy and quick to develop and maintain programs

Basic, Python, Java

Low-level language

develop special routings that use of memory. the computer's instruction set and hardware

assembly language, machine code.

Advantage and disadvantage of High-level and Low-level language

	High-level language	Low-level language
advantage	1.Independent of the type of computer being used. 2. Easier to read, write and understand programs. 3. Quicker to write programs. 4. Programs are easier and quicker to debug . 5.Easier to maintain programs in use.	1. Can make use of special hardware . 2. Can make use of special machine dependent instructions . 3. Code doesn't take up much space in primary memory. 4. Code performs a task very quickly .
disadvantage	1. Programs can be larger . 2. Programs can take longer to execute . 3. Programs may not be able to make use of special hardware.	1. It takes longer to write and debug programs. 2. Programs are more difficult to understand .

Translators

Computers use programs with binary instructions so programs must be translated into binary for the computer to follow them. Needs translation before execution.

Compiler

1. Translates a **high-level language program into machine code** in **one go**.
2. An **executable file** of machine code is produced.
3. One high-level language statement can be translated into several machine code instructions.
4. Once compiled, **programs are run without the compiler**.
5. A compiled program is usually distributed for general use.
6. If errors are detected, **an error report is produced** (instead of a compiled program).

Interpreter

1. Executes a high-level language program **one statement at a time**.
2. No executable file of machine code is produced.
3. One high-level language program statement may require **several machine code instructions** to be executed.
4. Interpreted programs **cannot be run without the interpreter**.
5. An interpreter is often used when a program is being developed.
6. If an error is **detected in a statement**, execution stops and an error message is output.

Assembler

1. Translates a low-level assembly language program into machine code.
2. An executable file of machine code is produced.
3. One low-level language statement is usually translated into one machine code instruction.
4. Assembled programs are used without the assembler.
5. An assembled program is usually distributed for general use.

Advantage and disadvantage of interpreter and compiler

	interpreter	compiler
advantage	Easier and quicker to debug , test and edit programs during development.	A compiled program: 1. can be stored ready for use 2. can be executed without the compiler 3. takes up less space in memory when it is executed 4. is executed in a shorter time
disadvantage	Programs: 1. cannot be run without the interpreter 2. can take longer to execute .	Takes longer to write, test and debug programs during development.

Integrated development environment(IDE)

a suite of software development tools used by programmers to aid the writing and development of programs.

1. code editor
2. translator
3. runtime and debugger
4. error diagnostics
5. auto-completion
6. auto-correction
7. auto-documenter and pretty printing