

Application Software and Operating Systems KO

| | | | | | | |
|--|---|--------------------------------|---|--|--|---|
| Type | Purpose | Examples | Utility Software | | | |
| Word Processor | For creating, editing, formatting and printing text documents. | Microsoft Word, Google Docs | What is utility software? | | | |
| Database Management Software | For storing, retrieving, manipulating and managing data. | MySQL, Oracle | <ul style="list-style-type: none">In A Level Computer Science, utility software is designed to help analyse, configure, optimise, or maintain a computerIt supports the operating system, which is different to application software that performs tasks that benefit end-users | | | |
| Web Browser | For browsing the web and accessing information online. | Google Chrome, Firefox | Utility Software | Purpose | Role | Examples/Notes |
| Graphics Manipulation | For editing and manipulating images, photos and graphic designs. | Adobe Photoshop, GIMP | Disk Defragmentation | To rearrange the files on a hard drive to increase efficiency | By putting files into contiguous blocks and minimizing empty spaces, disk defragmentation speeds up file access and can improve overall system performance | Modern SSDs generally don't require defragmentation |
| Spreadsheet Software | For organising, calculating and analysing numerical data. | Microsoft Excel, Google Sheets | File Management | To organize, search, rename, and relocate files stored on the system | Includes creating, deleting, moving, and renaming files and folders | Windows Explorer, macOS Finder |
| Presentation Software | For creating visual and multimedia presentations. | Microsoft PowerPoint, Keynote | Device Driver | To provide an interface between the hardware and the operating systems | Ensures that the OS and programs can communicate with the hardware without needing to know the hardware's precise details | The OS usually manages drivers, but users may sometimes need to update them |
| Antivirus Software | To protect the computer system from viruses and other malicious software. | Norton, McAfee | System Cleanup | To free up space on the system by removing unnecessary files and data | System cleanup utilities remove temporary files, system cache, unused applications, and other "junk" that can slow down the system | CCleaner, Disk Cleanup on Windows |
| Email Client | For managing and accessing email across different email providers. | Outlook, Thunderbird | Security | To protect the computer system from various threats like viruses, malware, and spyware | Monitors the system and controls the computer's activities to protect it from threats | Norton, McAfee, Windows Defender |
| Video Editing Software | For editing and producing videos and films. | Adobe Premiere, Final Cut Pro | | | | |
| Integrated Development Environment (IDE) | For developing, testing and debugging software code. | Visual Studio, IntelliJ IDEA | | | | |
| Virtualisation Software | For creating and running virtual machines on a single physical machine. | VMware, VirtualBox | | | | |

Open Source & Closed Source Software

- Think of software like a recipe. It's a set of instructions that tell a computer what to do
- **Open Source Software (OSS)** is like a **shared recipe**. Anyone can look at it, change it, or share it with others. It's all about community and collaboration
- **Closed Source Software (CSS)** is like a **secret recipe** that only particular chefs know. People can taste the dish at a restaurant but can't see the exact ingredients or change the recipe. This is how closed-source software works; the instructions are kept secret
- Knowing a little bit about these two types of software can help in understanding what's happening behind the scenes and in making better choices about the software that is used

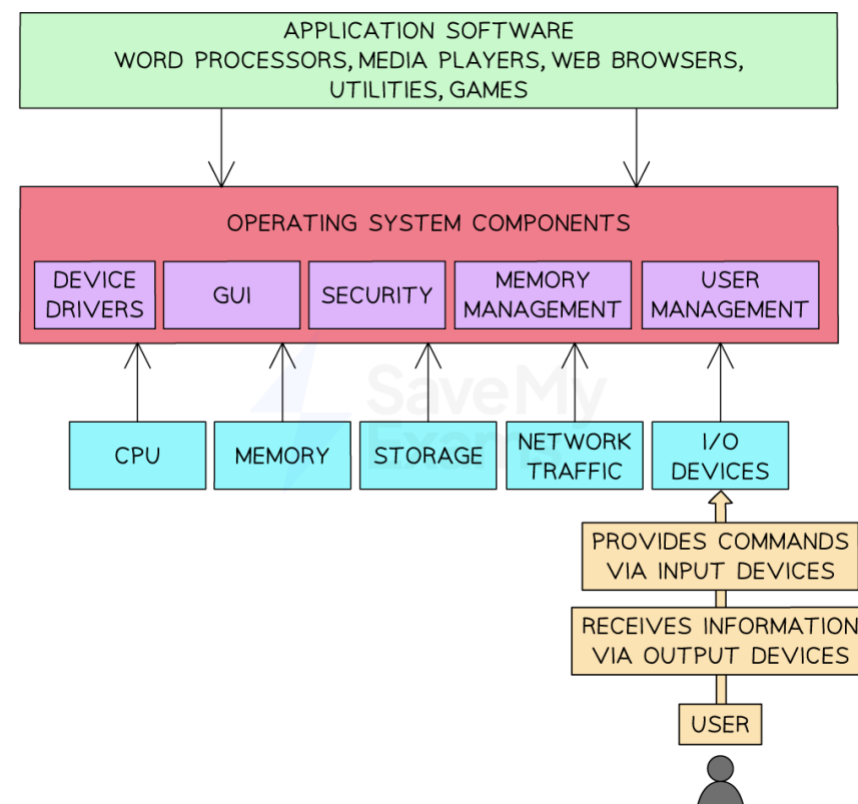
| Definition, Examples, and Typical Usage Scenario | | | |
|--|---|------------------------------------|---|
| | Definition | Examples | Typical Usage Scenario |
| Open Source Software | Users can view, modify, and distribute the source code | Linux, Apache HTTP Server | Ideal for collaborative projects, customization, transparency |
| Closed Source Software | The source code is hidden and proprietary. | Microsoft Windows, Adobe Photoshop | Ideal for businesses requiring polished, supported products, intellectual property protection. |

| Benefits and drawbacks to the creator | | |
|---------------------------------------|--|--|
| | Benefits | Drawbacks |
| Open Source Software | Collaboration, community engagement, faster innovation | Less control, burdened with requests from users |
| Closed Source Software | Greater control, revenue through sales, IP protection | Slower innovation, full responsibility for updates & flaws |

| Benefits and drawbacks to the user | | |
|------------------------------------|---|---|
| | Benefits | Drawbacks |
| Open Source Software | Often free, customisable, transparent | Might be less user-friendly, compatibility issues, may contain bugs |
| Closed Source Software | More polished products, professional support, consistency | Costly, less customisable, potential trust issues |

Describe the purposes of operating systems.

- Used to **control the hardware** of the system/resource management... through software like hardware drivers/system software
- Used to provide a **platform on which applications can run** ... deals with issues that the software may have with e.g. storage of files
- Provides a **user interface** with operator ... to allow communication between user & hardware
- Handles communications... using rules/protocols to govern the communication
- Handles translation of code ...e.g. compiler/interpreter
- Has many **utility programs**... used to carry out housekeeping on system



Different types of operating systems

A **single-user** operating system will only allow a single user to use the computer at any one time.

Multi-tasking runs multiple programs at the same time

Multi-user allows multiple users at the same time (must be clear that candidate is not discussing an OS that simply has multiple accounts)

Distributed allows multiple computers to work together on a single task

Embedded

- ...has a dedicated/limited function
- ...is read-only / cannot be changed

Real-time:

Commonly used in time-critical computer systems, a real time OS is designed to perform a task within a guaranteed time frame.

State an application of a computer system which requires a real-time operating

- Airline booking system. because it is important that the system is updated before the next input so no double booking
- Computer game... because it is important that the system is updated before the next input to avoid any game lag
- within self-driving cars: any situation where a response within a certain time period is crucial to safety.

Describe what is meant by an operating system.

- A set of software/programs ...designed to manage the hardware of the system.
- Describe a single-user operating system. (3)
- Allows one user ... at a time to use the system
- Allocates each user with rights
- Keeps the user files separate

Describe a multi-tasking operating system.

- Allows more than one task/software to run (apparently) simultaneously
- Use of separate windows for each task
- Each is given an amount of processor time before... going on to next
- E.g. playing music while typing essay

Describe a multi-user OS

- One computer with many terminals/more than one user at a time
 - Each terminal given time slice... in turn
 - Each time slice very small (roughly 1/100 of a second)
 - Use of flags
 - Use of priorities/privileges
 - Data is separated/ security provision essential/user rights
- Use: e.g. Supermarket checkout system/online gaming/mainframe serving many terminals

Different Types of Interfaces

Graphical User Interface (GUI)

- Uses **icons, windows, frames**, and **menus** for interaction.
- A **pointer** (e.g. mouse cursor) is used to navigate and select.
- Intuitive and user-friendly, especially for non-technical users.

Form-Based Interface

- The computer **displays prompts**, and the user provides input in predefined fields.
- Common in data entry systems (e.g. online forms, databases).
- Reduces user error by restricting input formats.

Menu-Based Interface

- Presents a **limited set of options** or commands for the user to choose from.
- Examples:
 - **Tourist information systems**
 - **Touchscreen kiosks** using pointing devices or fingers
- Easy to use, especially for users with little training.
- Ideal for systems where a **simplified navigation** experience is important.

Natural Language Interface (NLI)

- Allows users to **communicate in everyday human language**, either typed (keyboard) or spoken (microphone).
- Examples include **expert systems** or virtual assistants.
- Benefits:
 - Users can focus on the task rather than the interface.
 - Reduces the learning curve for complex systems.
- Challenges:
 - **Complex processing** is required to understand varied language input.
 - Simplifying queries can improve performance and reliability.

Utility Software

Utility software (also known as housekeeping programs) is a type of **system software** designed to help manage, maintain, and control computer resources. It performs **common tasks** that support the operating system and enhance the functionality of the computer.

| Utility | Purpose | Example Use |
|----------------------|--|--|
| Compression Software | Reduces the file size to save space or improve transfer speed. | Compressing documents to email |
| Hardware Drivers | Allows the OS to communicate with hardware devices. | Used when installing a new printer or scanner. |
| Anti-virus Software | Detects and removes malicious software (viruses, malware). | Protects the system from threats when browsing or opening downloads. |
| File Handlers | Enables file operations like create, delete, copy, move, rename, open, or close . | A student manages coursework files by organising them in folders. |
| Backup Utilities | Creates copies of important files to prevent data loss . | Automatically backs up an author's work daily to an external drive or cloud. |

Interrupt handling

START REGISTERS PRIORITY STACK ISR FLAG

1. Interrupt checked for at **start/end** of each **fetch-execute cycle**
2. If the interrupt is of a lower/equal **priority** to the current process then the current process continues
3. (If interrupt raised) contents of **registers** copied to **stack**
4. Flags are set to determine if interrupts are enabled / disabled
5. Program counter changed to point to Interrupt Service Routine (ISR) // ISR runs
6. After interrupt complete, previous register values restored back from stack
7. Flag is reset
8. If higher priority interrupt received during servicing of interrupt...
9. ...this is added to stack and new interrupt dealt with

The **ISR (Interrupt Service Routine)** is *software* — it's the function/code that runs when an interrupt happens. **Loading the ISR** means starting the correct piece of code to handle the interrupt.

Example ISR's:

Timer Interrupt - Every 1ms or 1s, your timer triggers, and you toggle an LED or update a clock.

Keyboard interrupt - You press a key, the keyboard hardware signals the CPU, and the ISR reads the key.

Network Packet Received - A new network packet arrives at the Ethernet port; ISR quickly moves it into a buffer.

Scheduling algorithms

The purpose of scheduling is to

- Make efficient use of processor time
- Make efficient use of resources
- Maximise the number of users
- Ensure that there is no apparent delay for users
- Maximise throughput of the CPU

Why scheduling is used.

- Maximise number of users
- ...with no apparent delay
- Maximise number of jobs processed
- ...as quickly as possible
- Obtain efficient use of processor time / resources
- ...dependent upon priorities
- ...to ensure all jobs obtain processor time/long jobs do not monopolise the processor

Explain why operating systems use scheduling.

- Ensure all tasks are processed
- ...by changing priorities where necessary
- Process as many jobs as possible
- ... in the least possible time
- Maximise number of interactive users
- ...receiving fast response times

Round robin scheduling

is one method that may be used by a **multi-user** operating system. Describe round robin scheduling. (3)

- Each user allocated a time slice
- When time slice is up, system moves to next user
- If next user needs processor, user given time slice
- Repeat until all users serviced
- Users may have different priorities
- Time slices are very small/fractions of seconds
- No apparent delay for any user

| Algorithm | | Benefits | Drawbacks |
|-----------------------------|--|---|--|
| Round Robin | Each job gets a fixed time slice . Jobs cycle until completed | All processes get a fair share of the CPU Good for time-sharing systems Predictable, as every process gets equal time | Choosing the right time quantum can be difficult This can lead to a high turnaround time and waiting time for long processes |
| First Come, First Served | Jobs are processed in the order they arrive | Simple and easy to understand Fair in the sense that processes are served in the order they arrive | This can lead to poor performance if a long process arrives before shorter processes High-priority tasks wait for their turn in the queue |
| Multi-Level Feedback Queues | Jobs placed in different priority queues ; jobs can move between queues based on behaviour. | Smaller tasks are prioritised Creates a prioritisation system where similar-sized tasks are queued together | More complex than other algorithms Setting the correct parameters (e.g., number of queues, ageing rules) can be complex |
| Shortest Job First | Executes jobs with shortest total time first. | Minimises waiting time Efficient and fast for short processes | Requires knowing the burst time of processes in advance Long processes can starve if short processes keep arriving |
| Shortest Time Remaining | Preemptive version of SJF; always runs job with least time left . | Ideal for jobs that have shorter burst times It is preemptive, so it can be aligned with CPU for best performance (time quantum) | Like SJF, it requires knowing the burst time of processes in advance High context switching overhead due to preemption |

| |
|--------------------------|
| Memory Management |
|--------------------------|

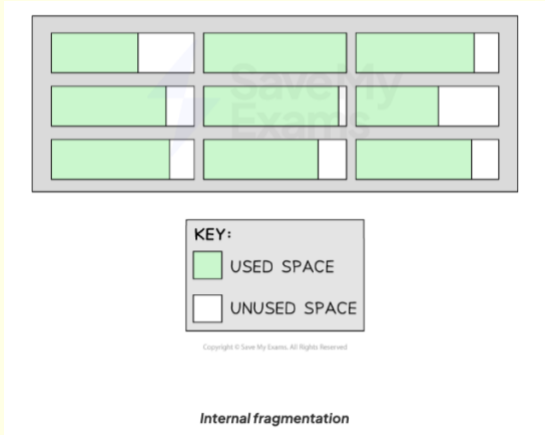

What is memory management?

- Memory management is a fundamental role of the operating system, dealing with the allocation and deallocation of the computer's primary memory
- When a user opens an application, its data is loaded from storage into active memory so that it can run smoothly
- When a user opens a file from the file system, e.g. word document, the CPU loads this file data, as well as application data, into the primary memory
- Primary memory is a limited resource in the system, so it needs careful management
- Benefits of memory management are:
 - Efficient allocation of memory enables multitasking, allowing multiple programs to run at once
 - Memory management maintains security, it does not let programs access memory reserved for other programs
- Memory management is made more efficient through 3 techniques:
 - **Paging**
 - **Segmentation**
 - **Virtual Memory**

Examiner Tips and Tricks:

Make sure you can name one benefit and one drawback for each memory management method in this revision note.

Virtual memory is a method of memory management - Not to be confused with Virtual Storage, which is storing files in the cloud.

| Memory Management Technique | Description | Example | Benefits | Drawbacks |
|-----------------------------|--|---|--|--|
| Paging | <p>Divides memory into fixed-sized blocks (pages)</p> <p>Pages are physical divisions</p> | <p>A process needing 200KB of memory is divided into four 64KB pages, leaving 8KB unused in the last page</p> | <p>When an application is launched, data will be moved from the hard disk into Pages for faster access</p> | <p>This can lead to internal fragmentation</p> <p>The box below this shows many Pages, each with varying sizes of internal fragments</p>  |
| Segmentation | <p>Divides memory into variable-sized segments based on logical parts of a process and complete sections of programs.</p> <p>Segments are logical divisions.</p> | <p>In a video editing application, different segments may be created for video data, audio data, effects, and UI elements</p> | <p>Segmentation is space-efficient due to only allocating space depending on the amount an application needs</p> | <p>This can result in external fragmentation</p> <p>This shows different application data assigned to a Segment</p> <p>The arrangement of data in the segment becomes more fragmented over time because as blocks are taken away it's not possible to guarantee a new block will occupy the same amount of space</p> <p>Right shows a defragmented version of the Segment to highlight the total unused space</p>  |

| | |
|--|--|
| <p>Virtual Memory</p> <p>Why is virtual memory needed.</p> <ul style="list-style-type: none"> • To allow programs to run that need more memory than is available <p>Describe how virtual memory is used.</p> <ul style="list-style-type: none"> • Use of backing store as if it were main memory/temporary storage • Uses paging/fixed size units • Swaps pages between memory & backing store • ...to make space for pages needed • Holds part of program not in use • Allows programs to run that need more memory than is available <p>Describe the problem of disk thrashing.</p> <ul style="list-style-type: none"> • Occurs when using virtual memory/moving pages between memory & disk • Disk is relatively slow • High rate of disk access • More time spent transferring pages than on processing | <p>Segmentation and paging</p> <p>To enable memory to be shared use segmentation or paging when managing memory. State two ways in which segmentation and paging are similar.</p> <ul style="list-style-type: none"> • Both are ways of partitioning memory • Both allow programs to run despite insufficient memory/both are used for virtual mem- • Segments and pages are both stored on backing store • Segments and pages are assigned to memory when needed • State one difference between segmentation and paging. (1) • Segments are different sizes but pages are fixed size • Segments are complete sections of programs, but pages are made to fit sections of memory • Segments are logical divisions, pages are physical divisions <p>Explain one problem that may occur when using paging and segmentation.</p> <ul style="list-style-type: none"> • Disk thrashing • ...where more time is spent swapping pages than processing ...and the computer may 'hang' |
|--|--|

BIOS

- BIOS refers to the Basic Input/Output System of a computer
- BIOS is a piece of firmware stored on a small memory chip on the motherboard
- On system start, the BIOS is the first software to run
- It performs a POST (Power-On Self-Test), a diagnostic testing sequence that ensures all the hardware components are working properly
- If the BIOS encounters any errors during this test, it will either halt the boot process or issue an error message
- If the POST succeeds, the BIOS will run the Bootstrap loading sequence, which is the program responsible for starting the operating system

Virtual Machines

- Virtual machines (VMs) are entire operating systems running inside another operating system

| Benefits | Drawbacks |
|---|--|
| <ul style="list-style-type: none">• Virtual machines (VMs) are entire operating systems running inside another operating system• Allows running different operating systems (e.g. Linux on Windows) simultaneously on a single machine.• VMs are a way to create isolated test environments, that leave the host operating system unaffected• Multiple VMs can run on one physical machine, maximising hardware usage and reducing the need for multiple physical devices.• deal for running potentially unsafe software or malware for analysis in a controlled environment.• | <ul style="list-style-type: none">• VMs are slower than physical machines because resources (CPU, RAM, disk) are shared with the host system and other VMs.• Running multiple VMs can consume significant system resources, leading to slower performance or system crashes if not managed properly |

Device Drivers

- A device driver is a piece of software that enables communication between an operating system and specific hardware devices such as:
 - Printers
 - Graphics cards
 - Network cards
- Device drivers allow the OS to control and interact with those devices
- Because many external devices have embedded system software, a driver bridges the gap between a major operating system and a tiny hardware OS
- Device drivers make it possible to perform specific operations on the hardware e.g. a printer driver enables the OS to send print commands and manage print jobs
- Most hardware manufacturers write their own device driver software, meaning a single operating system may have several printer drivers installed