Need for Operating Systems

- An operating system is software that manages the computer (the hardware, the software, the security of the system and the user interface).
- There are seven key purposes of an operating system:
  - **Managing the Processor -** Deciding which process to execute next and handling interrupts to the currently executing process.
  - **Managing the Memory -** Allocating and managing the memory used and relinquished by processes using paging and segmentation. Also manages Virtual memory.
  - **Managing External Devices -** Using device drivers to translate operating system's instructions into those understood by specific models of hardware device, e.g. printer.
  - **Providing a Platform for Software Utilities -** The operating system provides a framework for applications to run, allowing them to open and close files easily etc. The user can also interact with the operating system to manage files and use a range of facilities e.g. file explorer.
  - **Providing Networking -** Allows communication through protocols to other machines and devices.
  - **Managing Security -** Servicing and denying requests to access the resources connected to the computer e.g. Shared Drives.
  - **Providing a User Interface** - Users interact with applications through the interface provided by the operating system. This may be graphical, such as Graphical User Interfaces (WIMP - Windows, Icons, Menus, Pointers), menu driven or command line interface.

Types of Operating System

Multi-tasking Operating Systems

- With computers, smartphones and tablets, we expect to have more than one application open at once:
- In single-processor systems, each active program is scheduled to receive a tiny time slice in quick rotation, giving the impression they are operating at the same time.
- Even modern processors with multiple cores still need to handle many processes at the same time.

Multi-user Operating System

- Most operating systems you will be familiar with from school or work allow more than one person to use a computer at the same time.
- The computer will manage the user's various permissions and access rights when they log on.
- Server operating systems software will handle the requests of multiple people using different computers on a network at the same time.

Distributed Operating System

- You can combine the processing power of multiple computers across a network for a single task.
- In distributed computing, the operating system controls and coordinates the computers, presenting them to the user as if they were a single system.
- As a system becomes busy with more user requests, additional servers might automatically join the system, load-balancing requests without you every knowing.

Embedded Operating Systems

- Computers are not limited to the desktop PCs, laptops, tablets and phones we use every day.
- Processors exist in an almost limitless number of devices such as:
    - Washing machines
    - set-top boxes
    - TVs
    - Car engine management
    - Home assistants
    - Traffic lights
- Embedded operating systems tend to run on dedicated hardware so they run with maximum efficiency, using low-powered processors and very little memory.

Real-time Operating Systems

- In safety-critical environments such as aircraft autopilot systems, hospital monitoring machines, self-drive cars and missile systems, processes have to be guaranteed to execute within a known time frame.
- Plenty of redundancy is built into these systems so they can handle sudden increases in input.
- As such, processors in real-time operating systems rarely run at capacity.

Utility Software

- Tend to be small programs which have one very specific purpose.
- Purpose is normally concerned with the maintenance and well running of a computer system.

Antivirus

- Any programs which help to detect and remove any malicious programs which are designed to harm a system in some form or another.

Disk-Defragmentation Software

- Over time, a hard disk will become fragmented. This occurs as new files are added and old ones deleted.
- Free space often becomes scattered across the disk and a program or file has to be split up into multiple small chunks and spread across the disk to make it fit.
- The more fragmented the disk becomes, the slower the computer will perform.
- Defragmentation utilities attempt to consolidate the split files and free space, so reading and writing to the disk becomes more efficient.

Compression

- Any programs that attempt to reduce the amount of physical space that files take up when they are stored.

File Manager

- Programs that allow users to easily copy, move, rename, delete or modify files and directories on a computer system.

Firewalls

- These are any programs that help to prevent unauthorised access to a computer system or network.
- Schools or other large organisations will often install firewalls on their networks to both prevent intrusion on their network, but to also prevent employee's internally accessing content outside but to also prevent employee's internally accessing content outside of the network during work hours that they deem inappropriate.

Backup Software

- Any programs that allow for manual or automatic backing up of files.
- This could be to another disk, to a removable device, to a network file server or to a cloud based storage system.

Virtual Machines

- Virtual machine definition: “A program that has the same functionality as a physical computer.”
- Emulators trick a program into thinking it's running on its native hardware when, in fact, it's running on an entirely different machine.
- Arcade games from the 1980’s can be emulated on today’s hardware.
- The programs themselves are completely unaware they are being emulated.
- The code is being executed exactly as it was originally - this is an example of a virtual machine.
- Another example of emulation in games development is to create the game on a PC but use an emulator for another device such as a games console or smartphone to test they program before release.

Server Technology and Virtual Machines

- This technique can be used to support a large number of virtual servers spread over a small number of physical servers.
- Here we have four physical servers, but they are running as six virtual servers.
- As demand on the network, additional servers can be spun up and the load balanced across different physical devices.
- Another major advantage of this is if one server stops working, the other can pick up the load and continue working as if nothing happened.

Virtual Machines and Intermediate Code

- Java code is a good example of intermediate code running on a virtual machine.
- Java code is designed to run on many different platforms.
- With traditional programming, you would either have to use the specific language of each device or a suitable compiler for that device.
- Java gets around this by compiling its code into a half-way code known as bytecode or intermediate code.
- This code is translated by a java virtual machine running on a target device, which then translates it into a specific machine code.
- This process makes the code highly portable between devices.

Device Drivers

- Device driver definition: “Software that tells the operating system how to communicate with a device.”
- Your computer has to be able to output to a wide range of devices.
- A document printed from a word processor should look the same no matter what make or model of printer you send it to. However, the technology behind each printer is very different.
- The device driver translates the operating systems instructions to print the document into a series of instructions that a specific piece of hardware will understand.
- A computer program that provides a software interface between operating system and a (specific) hardware device.

Interrupts

- The purpose of the CPU is to fetch, decode and execute instructions.
- However, peripherals and software may need attention and therefore need to signal the CPU. There are two ways of doing this:
  - Polling is where the CPU checks each device to see whether it needs attention. This is very inefficient, as if a device does not need attention then there is no point in checking it.
  - Instead a system called interrupts is used.
- A device sends a signal on the control bus to the CPU to indicate that it requires attention.
- If there is an interrupt, then the CPU needs to stop what it is doing and service the interrupt (by running the code to service the interrupt). This is known as an interrupt service routine.
- This creates a problem as the PC must keep track of the next instruction in the memory. If this has to be changed to the first instruction of the interrupt service routine, then when the ISR is finished, how will the CPU know where to go in the previously executing program?
  - This is solved by using a stack. The contents of the PC is moved down to the stack, so that the first address of the ISR can be moved to the PC.
  - When the ISR is finished, the address in the stack can then be popped off the stack into the PC, and then the CPU knows where to continue executing instructions.
- Interrupts always have a higher priority than the current execution.

Types of Interrupt (+ Examples)

Hardware

- Power / reset button is pressed
- Memory parity error (corrupt memory)

Software

- Illegal instruction encounter
- Arithmetic overflow
- New logon request

Input / Output

- Buffer nearly empty
- Signal the completion of a data transfer to / from a device.

Memory Management

Paging

- The memory is divided into smaller sections which are all the same size (each chunk called a page).
- Each program occupies a different number of pages.
- The program is split into pages so they fit the free space pages.
- The operating system would use a page table to keep track of where in memory each of the pages are stored.
- This method therefore does not require all the pages to be stored continuously.

Segmentation

- Does not take into account what the instructions are actually doing in the sections of code that are separated.
- Keeps logical divisions of programs together rather than having fixed page sizes.
- Segmentation is slower than paging.

Similarities

- Both allow programs to run despite insufficient memory.
- Both pages and segments are stored on disks.
- Both pages and segments are transferred into memory when needed.

Differences

- Pages are a fixed size whereas segments are different sizes.
- Pages are made to fit sections of memory whereas segments are complete sections of the program.
- Pages are physical divisions whereas segments are logical divisions.

Problems with Both

- If physical memory is running low, virtual memory has to be used. If more time is spent moving pages / segments in and out of the memory on the disk (thrashing) than actually processing the instructions, the computer will be very slow.

Virtual Memory

- Moves the sections of the programs that are not being used out on the hard drive.
- When the sections of the program need to be used again they are sent back into the main memory.
- A lot of time is spent in moving instructions in and out of the RAM, known as disk thrashing (which makes the computer very slow.)
- The CPU cannot execute instructions from the disk.

Processor Scheduling

- Multitasking operating systems have to enable multiple programs to be executing at the same time.
- Even within one program, there may be multiple programs to execute simultaneously.
- For this to be possible, a multitasking operating system will require a scheduler.

The Need for Scheduling

- Processes as many tasks as possible in a given time
- Makes maximum use of CPU time
- It will try to minimise the delay between when the user requests something to be done and when that task is completed.
- Makes maximum use of resources such as input-output devices
- Is able to prioritise jobs
- It will ensure that no task is left uncompleted for too long, even if they are low priority
- Is able to alter priorities according to need

Scheduler

- Manages what process to execute next and how much time it is allowed to execute for.
- A new process would join the process read queue.
- When the currently executing process in the CPU either finishes or is stopped it will either finish execution completely (and leave the system) or it may be blocked (because of input / output command) meaning the process cannot continue until more data is received. It could also be given a certain amount of time before it has to pause its execution in which case it will return to the ready queue so another process can be executed.
- There are a number of algorithms the operating system may be using to manage the processes in the ready queue.

Process Ready Queue Algorithms

First Come First Served

- The processes are executed strictly in the order in which they arrive.
- If one process takes a long time, the others must wait.

Shortest Job First

- The scheduler picks the process that takes the shortest amount of time and runs them until they finish.
- The scheduler needs to know how long each process will take in advance.

Round Robin

- Each process is allocated a fixed amount of time known as a time slice or quantum.
- If the process is not complete at the end of the quantum, it returns to the back of the ready queue so the next process can have its turn.
- It is fair, but not always efficient.

Shortest Remaining Time

- Similar to shortest go first, but processes can be suspended if a higher priority process joins the queue.
- When the process no longer has the shortest remaining time it is suspended until it has the shortest remaining time again.

Process Blocked

- Happens when a running process requires data from the hard disk.
- The process is blocked until its request is serviced (interrupt).
- Generated to tell the scheduler that the blocked process can return to the process ready queue.

Multilevel Feedback Queues

- Where the number of queues is increased.
- Different queues have different priorities and the scheduler can move jobs between the queues