Component 1 | 1.2.1| Operating systems – System software

Specification & learning objectives

A Level	Specification point description
1.2.1a	The need for, function and purposes of operating systems
1.2.1b	Memory management (paging, segmentation and virtual memory)
1.2.1c	Interrupts, the role of interrupts and Interrupt Service Routines (ISR), role within the fetch decode execute cycle
1.2.1d	Scheduling: round robin, first come first served, multi-level feedback queues, shortest job first and shortest remaining time
1.2.1e	Distributed, embedded, multi-tasking, multi-user and real time operating systems
1.2.1f	BIOS
1.2.1g	Device drivers
1.2.1h	Virtual machines, any instance where software is used to take on the function of a machine including executing intermediate code or running an operating system within another

Resources

PG Online textbook page ref: 30-38

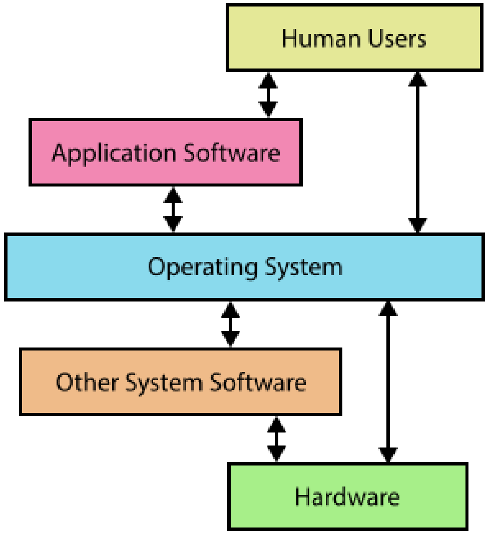
Hodder textbook page ref: 131-133

CraignDave videos for SLR 4

Component 1 | 1.2.1| Operating systems – System software

Key question: Why do computers need an operating system like Windows/Linux/MacOS?

An operating system is the core software that controls how a computer operates.
Well known operating systems include Windows, Linux, Unix, iOS and Android. Although these operating systems are very different in terms of the code they run, they do share a common purpose:



Human Users

Application Software

Operating System

Other System Software

Hardware

Control the hardware of the computer system.
e.g. hard disk, graphics card, mouse.

Manage software, in terms of loading (and unloading) into main memory.

Provide security, such as user name and password control.

Provide an user interface, to allow a person to interact with the computer.

Component 1 | 1.2.1| Operating systems – System software

Paging		**Segmentation**	
A method of manipulating memory which uses pages to stored code in fixed size blocks and allows programs to run despite insufficient memory. Uses virtual memory.		A method of manipulating memory which uses segments to store code in different sized, logical sections. Uses virtual memory.	
Allows programs to run despite insufficient memory using virtual memory	When virtual memory is used, if it takes too long for pages to be moved to the disk the computer will slow down (Disk thrashing)	Allows programs to run despite insufficient memory using virtual memory	When virtual memory is used, if it takes too long for segments to be moved to the disk the computer will slow down (Disk thrashing)
Pages are all of the same size		Segments have logical divisions which are more efficient	
Pages fit sections of memory		Segments are different sizes to match the sections of a program	
		Segments include complete sections of programs for easier reference	

Key question: What causes an interrupt to the CPU and how is it handled?

ISR (Interrupt Service Routine)

Determines what happens when an interrupt is raised.

Interrupt

A signal which stops the fetch decode execute cycle from running normally in order to prioritise a different a device.

Types of interrupt:

Hardware:

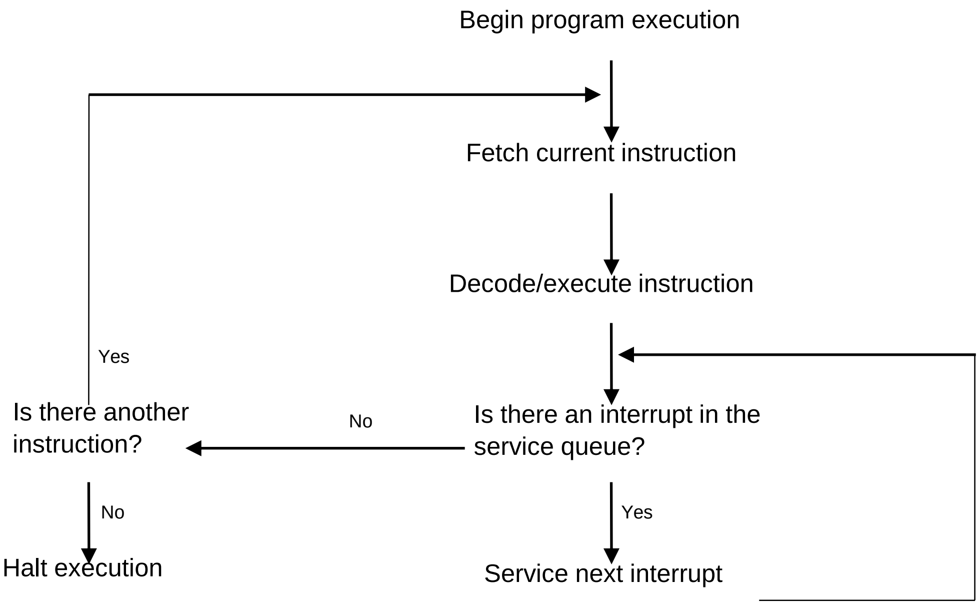
- Power pressed
- Memory parity error

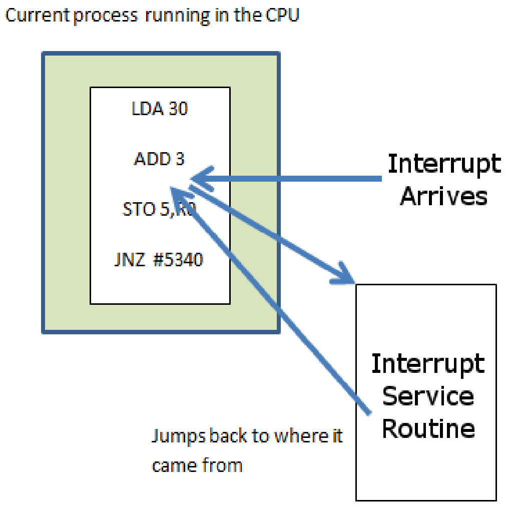
Software:

- Illegal instruction
- Arithmetic overflow
- New log-on request

Input/output:

- Buffer almost empty


Begin program execution
Fetch current instruction
Decode/execute instruction
Is there an interrupt in the service queue?
No
Is there another instruction?
Yes
No
Halt execution
Yes
Service next interrupt

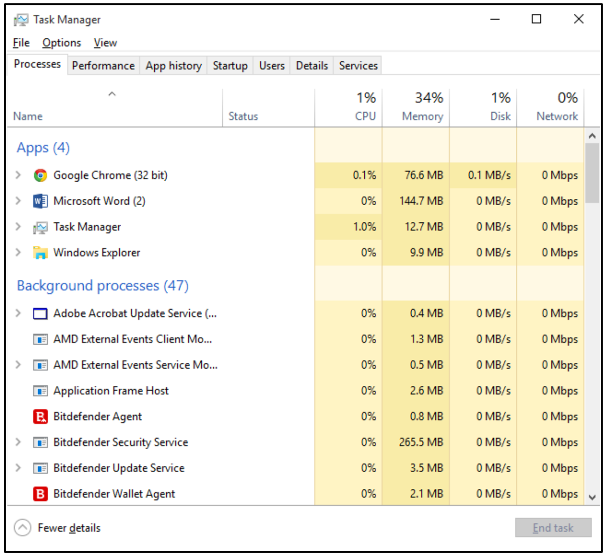

Current process running in the CPU
LDA 30
ADD 3
STO 5,R0
JNZ #5340
Interrupt Arrives
Interrupt Service Routine
Jumps back to where it came from

Program
Interrupt Service Routine
Program

Component 1 | 1.2.1| Operating systems – System software

Key question: From all the open programs in memory, how does the CPU decide which process to execute?

Process management involves the scheduling and switching of programs and threads. Modern interactive operating systems appear to allow multiple programs to run at the same time; for example, the user could play an MP3 file and browse the web at the same time. In effect, this is what appears to happen but in reality each processor can run only one process at a time.

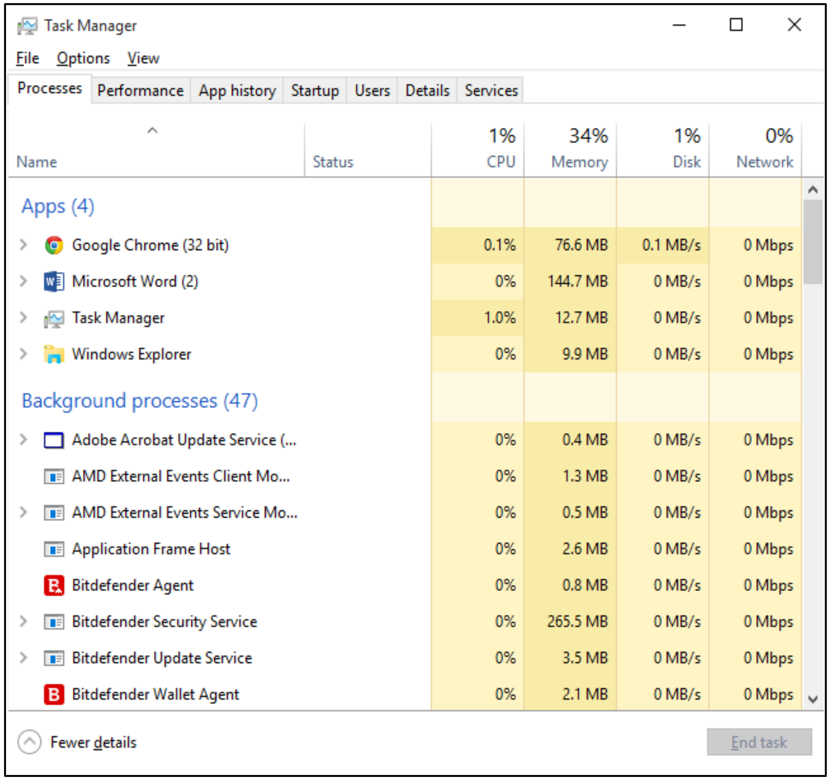


Name	Status	1% CPU	34% Memory	1% Disk	0% Network
Apps (4)					
Google Chrome (32 bit)		0.1%	76.6 MB	0.1 MB/s	0 Mbps
Microsoft Word (2)		0%	144.7 MB	0 MB/s	0 Mbps
Task Manager		1.0%	12.7 MB	0 MB/s	0 Mbps
Windows Explorer		0%	9.9 MB	0 MB/s	0 Mbps
Background processes (47)					
Adobe Acrobat Update Service (...		0%	0.4 MB	0 MB/s	0 Mbps
AMD External Events Client Mo...		0%	1.3 MB	0 MB/s	0 Mbps
AMD External Events Service Mo...		0%	0.5 MB	0 MB/s	0 Mbps
Application Frame Host		0%	2.6 MB	0 MB/s	0 Mbps
Bitdefender Agent		0%	0.8 MB	0 MB/s	0 Mbps
Bitdefender Security Service		0%	265.5 MB	0 MB/s	0 Mbps
Bitdefender Update Service		0%	3.5 MB	0 MB/s	0 Mbps
Bitdefender Wallet Agent		0%	2.1 MB	0 MB/s	0 Mbps

Many modern computers come with multiple cores, each of which can handle a single process, which means that a certain number of processes can be run simultaneously. However, at the same time, modern computer systems tend to be running hundreds of processes at once and so there is still a considerable need to switch between processes. An operating system schedules programs and switches between them by maintaining a process control block.

Component 1 | 1.2.1| Operating systems – System software

Key question: From all the open programs in memory, how does the CPU decide which process to execute?



Name	Status	1% CPU	34% Memory	1% Disk	0% Network
Apps (4)					
Google Chrome (32 bit)		0.1%	76.6 MB	0.1 MB/s	0 Mbps
Microsoft Word (2)		0%	144.7 MB	0 MB/s	0 Mbps
Task Manager		1.0%	12.7 MB	0 MB/s	0 Mbps
Windows Explorer		0%	9.9 MB	0 MB/s	0 Mbps
Background processes (47)					
Adobe Acrobat Update Service (...		0%	0.4 MB	0 MB/s	0 Mbps
AMD External Events Client Mo...		0%	1.3 MB	0 MB/s	0 Mbps
AMD External Events Service Mo...		0%	0.5 MB	0 MB/s	0 Mbps
Application Frame Host		0%	2.6 MB	0 MB/s	0 Mbps
Bitdefender Agent		0%	0.8 MB	0 MB/s	0 Mbps
Bitdefender Security Service		0%	265.5 MB	0 MB/s	0 Mbps
Bitdefender Update Service		0%	3.5 MB	0 MB/s	0 Mbps
Bitdefender Wallet Agent		0%	2.1 MB	0 MB/s	0 Mbps

Scheduling algorithms:

FCFS (First come first served):

Tasks are executed to completion and in order regardless of time.

SJF (Shortest job first):

The shortest task is executed first to completion. The algorithm needs to know the time each job will take in advance.

RR (Round robin):

Each task is given a certain amount of time. If it hasn't finished it re-joins the end of the queue.

SRT (Shortest remaining time):

The shortest task is executed to completion or until a task with a shorter remaining time joins the queue.

MLFQ (Multi-level feedback queues):

Multiple queues are used with different priorities and jobs are moved between the queues depending on their behaviour.

Component 1 | 1.2.1| Operating systems – System software

Key question: What are the features of different types of operating system?

There are many different Types of operating system, each designed for different circumstances. Type include: multi-tasking, multi-user, real-time, distributed and embedded operating systems.

A multi-tasking OS gives each process a time slice (see scheduling) to allow them to share a processor core. To the user it appears as if it's doing multiple things at once.

A multi-user OS, like Windows, can service multiple users at once, managing their preferences individually so each user has their own unique environment. It can also refer to a server operating system able to manage requests of multiple users over a network.

A Distributed OS uses parallel processing to spread the load across several computers. Jobs are split into several tasks, each run on a different computer, coordinated by the OS

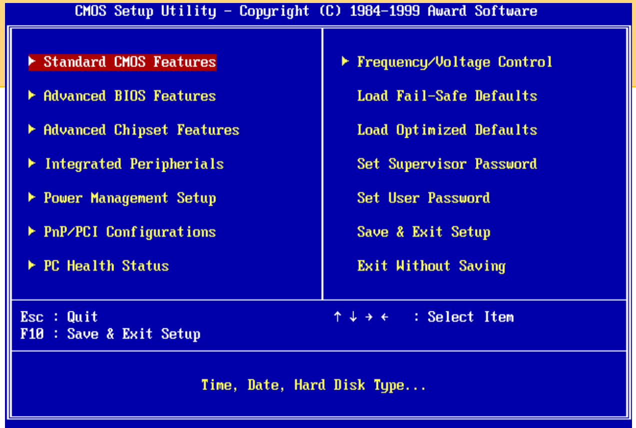
An Embedded OS is designed to run on very specific hardware. The OS is usually minimal, it may only require a few buttons as a user interface. However, each embedded system has varying requirements so they vary massively from a general purpose OS.

A Real-time OS is used in time and often safety critical applications. A real time OS places priority over certain processes and interrupts so that they can be executed within a known time frame. Typical examples are in hospital and aerospace equipment however a user controlled OS like Windows is also considered real time.

Component 1 | 1.2.1| Operating systems – System software

Key question: What is the relationship between these terms: BIOS, ROM, CMOS, POST, bootstrap and kernel?

- The purpose of the BIOS is to boot up the computer.
- BIOS is short for Basic Input Output System in the ROM.
- A BIOS also has an user interface to allow some settings to be altered.
- A BIOS is not part of the operating system.
- The BIOS is normally stored in non-volatile flash memory on the motherboard.
- The BIOS is started when the CPU receives a hardware-reset signal.
- POST or Power On Self Test checks that the system is OK.
- After POST the BIOS looks for a bootable drive.
- The operating system is loaded from the drive by the bootstrap.
- Personalised settings are loaded from the boot file.
- The operating system that is loaded into the memory is called the kernel, which is the fundamental part of the operating system which handles the input/output and processor/memory controls.



```
CMOS Setup Utility - Copyright (C) 1984-1999 Award Software

▸ Standard CMOS Features          ▸ Frequency/Voltage Control
▸ Advanced BIOS Features            Load Fail-Safe Defaults
▸ Advanced Chipset Features         Load Optimized Defaults
▸ Integrated Peripherials           Set Supervisor Password
▸ Power Management Setup            Set User Password
▸ PnP/PCI Configurations            Save & Exit Setup
▸ PC Health Status                  Exit Without Saving

Esc : Quit                        ↑ ↓ → ←   : Select Item
F10 : Save & Exit Setup

                  Time, Date, Hard Disk Type...
```

Component 1 | 1.2.1| Operating systems – System software

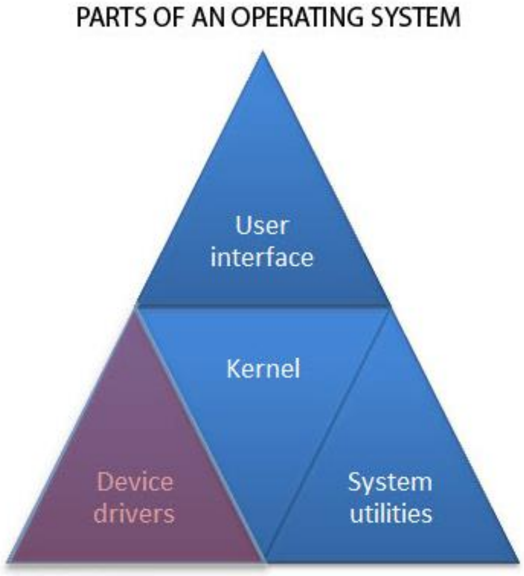
Key question: What is the purpose of a device driver?

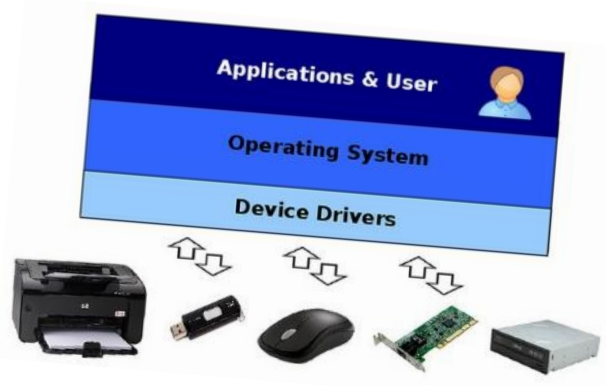
- Every piece of hardware that communicates with the operating system will require a device driver, for example:

INTERNAL: the hard disk, RAM, graphics card, sound card etc

EXTERNAL: printer, monitor, mouse, keyboard etc

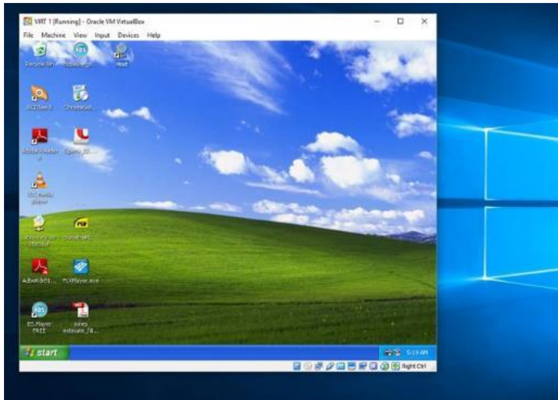
- A device driver enables the operating system to control and communicate with the device.
- There could be hundreds of device drivers pre-installed with the operating system, and the right ones for that particular computer set-up is loaded on boot-up.
- The exact detail of which device driver is needed by the operating system is kept in a file - in Windows, the file is called the 'registry' and in Linux the details will be stored as a number of 'configuration files'.
- This is why if you remove an operating system such as Windows from a hard disk, and install Linux instead, you will need to make sure you have all the correct device drivers available for each piece of hardware.





Key question: What is a virtual machine?

- A virtual machine is a program designed to simulate a fully functioning computer, but in software.
- A VM requires a host computer to run.
- There can be many VMs running on a single host.
- There are virtual machines written to emulate the hardware of a computer.
- There are virtual machines designed to run intermediate code from a specific computer language.
- Java and Actionscript can run in their associated virtual machine.
- Running a VM allows multiple operating systems to be used on the host computer.
- There are both free and commercial virtualisation packages available.



Typical exam questions

1. Explain the purpose and role of the BIOS. **[3]**

2. Explain two similarities and two differences between segmentation and paging. **[2]**

3. Explain why interrupts are used in a computer system. **[2]**

4. Describe three functions of an operating system. **[3]**

Component 1 | 1.2.1| Operating systems – System software

Target: □ Overall grade: □

Minimum expectations & learning outcomes

□	Terms 36-57 from your A Level Key Terminology should be included and formatted.
□	You must explain at least four reasons why a computer needs an operating system.
□	You must include a table which clearly shows the differences between paging and segmentation including the benefits and drawbacks of virtual memory.
□	You must explain why an interrupt can occur and using an illustration when and how it is handled during the fetch-decode-execute cycle.
□	You must include a comparison of the key features of the various scheduling algorithms.
□	You must explain the different classifications of operating systems and justify their use.
□	You must explain the role of the BIOS, device drivers and virtual machine in a computer system.
□	Answer the exam questions.

Feedback

Breadth	Depth	Presentation	Understanding
□ All	□ Analysed	□ Excellent	□ Excellent
□ Most	□ Explained	□ Good	□ Good
□ Some	□ Described	□ Fair	□ Fair
□ Few	□ Identified	□ Poor	□ Poor

Comment & action required

Component 1 | 1.2.1| Operating systems – System software

Reflection & Revision checklist

Confidence	Clarification
☹ 😐 ☺	Candidates need to have an understanding of why an operating system is required, along with the different tasks it performs within a computer system (e.g. resource management, file management, interrupt handling, security, providing a platform for software to run, providing a user interface and providing utilities).
☹ 😐 ☺	Candidates need to understand how operating systems manage memory. They need to understand the need for, purpose and function of paging to divide memory into usable fixed-size pages and how this aids in the transfer of memory for example virtual memory.
☹ 😐 ☺	Candidates need to understand what is meant by segmentation and how memory is divided into segments to allow access to memory.
☹ 😐 ☺	Candidates need to understand what is meant by virtual memory and why this is needed in a computer system.
☹ 😐 ☺	Candidates need to understand how paging is used in virtual memory, and the benefits and drawbacks of having and using virtual memory in a computer system.
☹ 😐 ☺	Candidates need to understand the purpose of interrupts within a computer system, why an interrupt might be generated and what happens within the CPU and memory in order to call an interrupt service routine.
☹ 😐 ☺	Candidates need to understand the need for scheduling of tasks by an operating system and the benefits that scheduling brings.
☹ 😐 ☺	Candidates need to understand that there are different scheduling algorithms, with each having benefits and drawbacks for tasks with specific characteristics.
☹ 😐 ☺	Candidates need to understand how the following scheduling algorithms work; round robin, first come first served, multi-level feedback queue, shortest job first and shortest remaining time.
☹ 😐 ☺	Candidates need to understand the different (and often overlapping) classifications of operating systems (distributed, embedded, multi-tasking, multi-user and real time), including the key features of each. They should be able to recommend (and justify) a type of operating system for a given scenario.
☹ 😐 ☺	Candidates need to understand the role of the BIOS in a computer system, and the steps that the BIOS goes through to start a computer.
☹ 😐 ☺	Candidates need to understand what is meant by 'device drivers' and why they are needed for communication between hardware and the operating system.
☹ 😐 ☺	Candidates should be able to describe what is meant by a virtual machine, how they can be used to execute intermediate code, how they can be used to run a software driven machine inside a physical machine and the benefits and drawbacks of each approach.