Accredited

A LEVEL

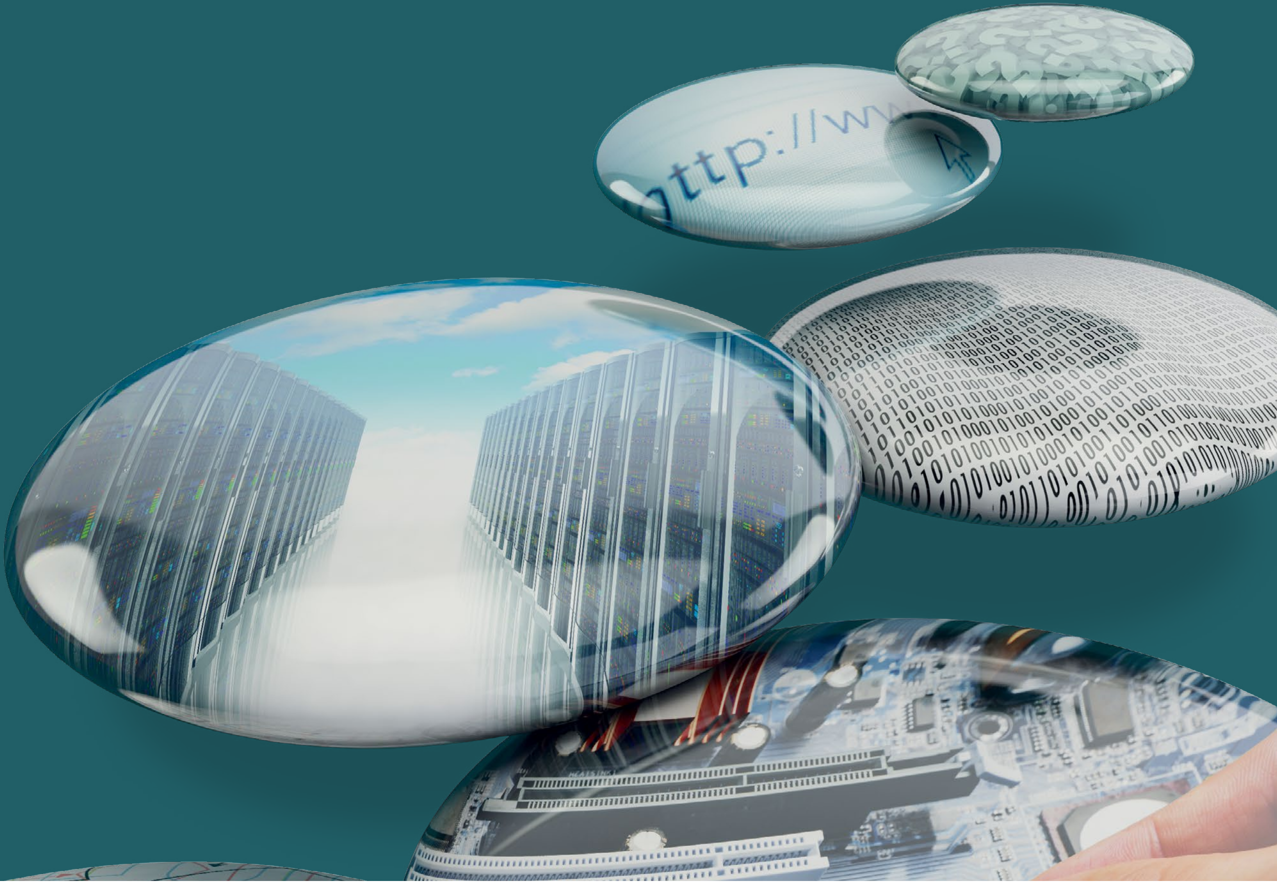
Delivery Guide

H446

COMPUTER SCIENCE

Theme: Function and purpose of operating systems

September 2015



CAMBRIDGE UNIVERSITY PRESS
www.cambridge.org

OCR
Oxford Cambridge and RSA

We will inform centres about any changes to the specification. We will also publish changes on our website. The latest version of our specification will always be the one on our website (www.ocr.org.uk) and this may differ from printed versions.

Oxford Cambridge and RSA Examinations is a Company Limited by Guarantee.
Registered in England. Registered company number 3484466.

Registered office: 1 Hills Road
Cambridge
CB1 2EU

OCR is an exempt charity.

CONTENTS

Introduction

Delivery guides are designed to represent a body of knowledge about teaching a particular topic and contain:

- Content: A clear outline of the content covered by the delivery guide;
- Thinking Conceptually: Expert guidance on the key concepts involved, common difficulties students may have, approaches to teaching that can help students understand these concepts and how this topic links conceptually to other areas of the subject;
- Thinking Contextually: A range of suggested teaching activities using a variety of themes so that different activities can be selected which best suit particular classes, learning styles or teaching approaches.

If you have any feedback on this Delivery Guide or suggestions for other resources you would like OCR to develop, please email resourcesfeedback@ocr.org.uk.

KEY

🗋	Click to view associated resources within this document.
▶ Click here	Click to view external resources.
AS Level only	AS Level content only.

Curriculum Content

Content (from A-level)

a) The need for, and function and purpose of operating systems.

b) Memory management (paging, segmentation and virtual memory).

c) Interrupts: the role of interrupts and Interrupt Service Routines (ISR), role within the Fetch-Decode-Execute Cycle.

d) Scheduling: round robin, first come first served, multi-level feedback queues, shortest job first and shortest remaining time.

e) Distributed, embedded, multi-tasking, multi-user and Real Time operating systems.

f) BIOS.

g) Device drivers.

h) 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.

Thinking Conceptually

a) The need for, and function and purpose of operating systems.

The topic starts by understanding the need for and function of operating systems.

http://en.wikipedia.org/wiki/Operating_system

▶ Click here

http://www.teach-ict.com/as_as_computing/ocr/H447/F453/3_3_1/features_of_os/miniweb/index.htm

▶ Click here

A good way to introduce this is to get students to brainstorm the operating systems that are out there, and you could perhaps have some for students to have a go at some different operating systems either through having some computers with them installed (Windows, Linux, MacOS), some different versions of operating systems on Raspberry Pi SD cards (http://www.raspberrypi.org/downloads/) or through emulators http://community.computingatschool.org.uk/resources/2775

▶ Click here

▶ Click here

Students should discuss why they use certain operating systems. Positives and negative experiences they have had etc.

A single/group card matching task to help students learn about the functionality of an Operating system: http://community.computingatschool.org.uk/resources/2048

▶ Click here

A video covering the basics of operating systems looking into the different managers (Process, File, Device, Memory)

https://www.youtube.com/watch?v=5AjReRMoG3Y

▶ Click here

b) Memory Management (paging, segmentation and virtual memory).

Students should understand that virtual memory uses secondary storage (such as the hard disk) as an extension of physical memory (RAM). Paging using a paging table is the process where data is moved to and from virtual memory when needed.

Segmentation allows programs to be broken up into smaller address spaces and each can have its own type of protection and can be shared amongst other processes.

http://en.wikipedia.org/wiki/Memory_management

▶ Click here

http://www.teach-ict.com/as_as_computing/ocr/H447/F453/3_3_1/memory%20management/miniweb/index.htm

▶ Click here

Interesting article about virtual memory in more practical terms: http://computer.howstuffworks.com/virtual-memory.htm

▶ Click here

You could start a class discussion about student experiences with memory in computers and smartphones such as how much memory is in a smartphone compared to a desktop computer or laptop, whether they have ever upgraded the memory, etc. What are the memory requirements for different operating systems and games? Students could also research about graphics cards and how this type of memory is different from other types of memory.

c) Interrupts, the role of interrupts and Interrupt Service Routines (ISR), role within the Fetch-Decode-Execute Cycle.

Interrupts are useful in computers, as rather than continually polling the inputs for changes, as soon as an event (e.g. a button being pressed) happens, the CPU is interrupted from what it is currently doing and takes a snapshot of all the registers in the CPU at that point in time so that it can return to that state when it has finished dealing with the interrupt. The process that happens as a result of the interrupt is called the Interrupt Service Routine (ISR).

The Fetch-Decode-Execute Cycle is a topic that is covered in specification 1.1.1b, so it would make sense to refer back to any prior teaching you have done on that or teach both parts together.

This website contains some information as a refresher about the Fetch Execute Cycle if students need it

▶ Click here

http://www.bbc.co.uk/education/guides/zws8d2p/revision

▶ Click here

http://www.teach-ict.com/as_as_computing/ocr/H447/F453/3_3_1/interrupts/miniweb/index.htm

d) Scheduling: Round robin, First come first served, Multi-level feedback queues, shortest job first and shortest remaining time.

Scheduling in a computer is where processes are given access to system resources, for example processor time which enables multi-tasking.

Round robin allocates a fixed time per process and then cycles through them until they are done.

First come first served are the simplest as they just put the actions in a queue as and when they arrive in that order.

Multi-level feedback queues categorise processes into their own queues which have their own priority. Each queue may have its own prioritisation algorithm. For instance, user and system processes may have their own queues.

Shortest job first is where the scheduler prioritises smaller jobs first to get them out of the way.

Shortest time remaining is similar to shortest job first except that it is pre-emptive (it interrupts the current processing rather than queuing it to be next)

▶ Click here

http://en.wikipedia.org/wiki/Scheduling_(computing)

▶ Click here

http://www.teach-ict.com/as_as_computing/ocr/H447/F453/3_3_1/scheduling/miniweb/index.htm

A simple and fun activity to show how deadlock can occur

▶ Click here

http://csunplugged.org/routing-and-deadlock

An activity can also be found at the bottom of this document which deals with different types of scheduling.

e) Distributed, Embedded, Multi-Tasking, Multi-User and Real Time operating systems.

There are different types of operating system.

A distributed OS is a collection of independent networked nodes. Each node may have its own hardware associated to it, and there will be some sort of system management software that brings all the interconnected systems together that co-ordinate individual and collaborative activities on the different nodes.

Embedded operating systems are termed as those that have a dedicated function and are part of (or can be part of) a larger device. They are generally lower power consumption, low cost and use a limited amount of hardware resources, which means that they need to be interfaced with in a bespoke manner compared to commonplace OS. An example of an embedded operating system would be embedded Linux on the Raspberry Pi.

Multi-user operating systems allow many different users to take advantage of the computer's resources simultaneously. Multi-user operating systems share processor time. Unix is an example of a multi-user OS.

Multi-tasking operating systems are those that allow you to have many different tasks happening at once, and so this is our standard modern operating system. However, some tasks can require much more processing power such as complex scientific/bank calculations. In this case, supercomputers are used but because they are expensive, instead of being single-user they are designed as multi-user systems so that many users can send tasks to it and share the resources.

Real time operating systems are supposed to run under predefined deadlines and so can be classed time critical. For instance, on an aeroplane when the pilot wants something to happen, it should do so straight away. One very important thing to remember about RTO systems is that "the right answer too late is wrong", since calculations for movements of a robot, for example, that arrive too late may no longer be valid if they are late.

http://en.wikipedia.org/wiki/Operating_system#Types_of_operating_systems

▶ Click here

http://www.teach-ict.com/as_as_computing/ocr/H447/F453/3_3_1/features_of_os/miniweb/index.htm

▶ Click here

f) BIOS.

BIOS stands for Basic Input Output System, and is firmware that is designed to run when the PC is first switched on. It is responsible for detecting, initialising and testing the hardware components, as well as booting the operating system.

http://en.wikipedia.org/wiki/BIOS

▶ Click here

http://www.teach-ict.com/as_as_computing/ocr/H447/F453/3_3_1/modern_pc/miniweb_pc/index.htm

▶ Click here

This site contains some BIOS simulators that can be explored by students to understand what they do

http://service.lenovo.partner-management.com/et.cfm?eid=1437

▶ Click here

▶ Click here

This howstuffworks article gives a good overview of BIOS in more practical terms

http://computer.howstuffworks.com/bios.htm

g) Device drivers.

When interfacing a piece of physical hardware to a computer, sometimes you may need to install a driver, although these days if you have a more common piece of hardware, then the likelihood is that your driver may already be included in the operating system

▶ Click here

http://en.wikipedia.org/wiki/Device_driver

▶ Click here

http://www.teach-ict.com/gcse_new/computer%20systems/buffers_drivers/miniweb/pg4.htm

Students should investigate what sort of device drivers are used for and why they are needed. Have they ever had to install a driver? What is a HID device? How do you think the introduction of the Internet and USB (as opposed to parallel and serial ports) has made drivers less annoying (i.e. updates can be more automatic, you can't lose the disk, less different cables to worry about, you usually have more than 1 USB, whereas there was usually only one parallel port).

h) 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.

A virtual machine is an operating system that emulates dedicated hardware. The user has the same experience on the virtual machine as they do on a dedicated machine. This enables the user to choose which operating system that they would like to use. It saves cost from reducing the need for separate hardware.

▶ Click here

http://www.teach-ict.com/as_as_computing/ocr/H447/F453/3_3_2/translators_compilers/miniweb/pg11.htm

▶ Click here

Students could watch this demo of a virtual machine being used and could discuss why someone would want to do this https://www.youtube.com/watch?v=ZmPY9nEFPMk

Also, when using a virtual operating system, you can have virtual device drivers that fool the operating system into thinking that it is direct communicating with a certain piece of hardware, but in actuality it is going through some function calls to the host operating system

▶ Click here

http://en.wikipedia.org/wiki/Device_driver#Virtual_device_drivers

Thinking Conceptually

Common misconceptions or difficulties students may have

There are quite a few different words (listed below) that students may not have encountered before in this topic. A student glossary is an excellent idea; learners can add terms that are unfamiliar to them and define them in their own words.

Some examples:

- Primary/secondary memory – RAM is primary memory, whereas hard disks or flash drives are secondary memory.
- Paging – Also called swapping – is the process of moving data between RAM and virtual memory. It moves data out of RAM that is not immediately needed and replaces it with ones that are. This process is slower than using just using primary memory alone since secondary storage is slower and the data has to move around more. It achieves this by using a page table to keep track of where data resides.
- Segmentation – When programs are loaded up into memory, segmentation allows the primary memory to store only the bit it needs at that particular time. For example, Dynamic Link Libraries can be stored on the hard disk until called by main memory by a program.
- Virtual Memory – The memory manager can trick the operating system into thinking that it has more physical memory than it actually does by making use of secondary storage such as the hard disk or flash drives.
- Interrupt – Interrupts what the CPU is currently doing and performs the ISR, then lets the CPU return to what it was doing before.
- ISR – The interrupt service routine is called when an interrupt occurs. Afterwards the CPU can carry on with what it was doing before.

Other keywords: Fetch-Decode-Execute Cycle, Scheduling, Round robin, First come first served, Multi-level feedback queues, Shortest job first, Distributed OS, Embedded OS, Multi-Tasking OS, Multi-user OS, RTOS, BIOS, Drivers, Virtual machine.

Conceptual links to other areas of the specification – useful ways to approach this topic to set students up for topics later in the course.

This unit content will support elements 1.1.1 'Structure and function of the processor' where students are introduced to the structure of a processor and the Fetch Decode Execute Cycle.

Operating systems can quite easily be taught as a standalone part of the specification, and would be logical to go onto 1.2.2 Applications generation, which requires the same level of in-depth understanding.

Thinking Contextually

Activities	Resources
Activity – How computers work (cse4k12.org) http://cse4k12.org/how_computers_work/index.html Activity involving getting the students to act out a simple computer simulation. Each student takes on the role of a different part of a simplified computer and they work in groups to run a simple program. The end result of this program is to draw a picture on a simulated computer display.	▶ Click here
Group Activity – :"Doing a million things at once" (csInside) http://csi.dcs.gla.ac.uk/workshop-view.php?workshopID=2 Gives students an idea of multi-tasking and why computers might 'freeze'. It illustrates the idea of First come first served and Round robin. The activity starts off by questioning if students use instant messaging programs, and, if so, how many other programs they have open at the same time. This leads into a discussion of how a computer might achieve that. You will then need to divide the class into groups of 6/7. One member of each group should be chosen as the Processor. This pupil must be a good writer and reader, and must be comfortable doing both in front of the group. One member of each group should be chosen as the Scheduler. The pupil should be the most organised in the group and should be happy telling the others what to do. The remaining group members will be tasks numbered from 1–4/5 (depending on group size).	▶ Click here

Thinking Contextually

Activities	Resources
Activity preparation: You will need to print out the task sheets for both First come first served and Round robin. A stopwatch is also useful.	
Group Activity – :"One problem, 1000 machines" (csInside) http://csi.dcs.gla.ac.uk/workshop-view.php?workshopID=13 Gives students an idea of the limitations of single processors and discusses the advantages and disadvantages of distributed computing. The task starts off by discussing the idea of why we would want to share the load of a task using practical and mathematical examples. • You then need to have three teams – each team is a 'machine' - Choose 1 team to be the Addition table (max 4 people) - Choose 1 team to be the Multiplication table (max 4 people) - Choose 1 team to be the Value/Result table (max 5 people) - Any remaining people can act as the Network	▶ Click here
Activity preparation: You need to make sure that you have printed everything from the resources zip file (it is probably easier to print out the labels onto paper rather than messing about with post-it notes – 'see instructions'). It will also be important to think about the room layout before you start the class/activity. To make things easier, the printable powerpoint has instructions for each person.	

👍 👎

We'd like to know your view on the resources we produce. By clicking on the 'Like' or 'Dislike' button you can help us to ensure that our resources work for you. When the email template pops up please add additional comments if you wish and then just click 'Send'. Thank you.

If you do not currently offer this OCR qualification but would like to do so, please complete the Expression of Interest Form which can be found here: www.ocr.org.uk/expression-of-interest

OCR customer contact centre

General qualifications
Telephone 01223 553998
Facsimile 01223 552627
Email general.qualifications@ocr.org.uk

A DIVISION OF
CAMBRIDGE ASSESSMENT

For staff training purposes and as part of our quality assurance programme your call may be recorded or monitored.