5.1 Operating Systems

Name: _____ Class: _____ Date: _____ **Total: 45 marks**

KEY WORDS operating system • memory management • process management • device driver • interrupts

Objective

Learn to answer exam questions on the **operating system** (OS): why every computer needs one, and the five **key management tasks** the syllabus names. This is the opening sheet of topic 5: utility software follows on sheet 5.1.1, program libraries on 5.1.2, translators on 5.2 and the IDE on 5.2.1.

You must be able to:

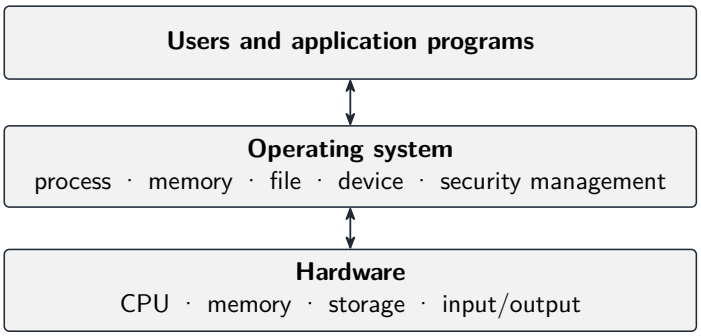
- explain why a computer system requires an operating system
- describe the key management tasks: **memory**, **file**, **security**, **hardware** and **process** management
- explain how memory management and process management support **multitasking**
- tell a hardware interrupt from a software interrupt

1 Worked examples

Study these first. Each one shows a **method** that a question later on this sheet needs.

■ Why a computer needs an operating system

Hardware on its own can only fetch and run instructions: it knows nothing about files, programs or users. The OS is the software layer between the programs and the hardware.



■ "Describe the purpose of an OS": the five-mark answer

Five different points, one mark each:

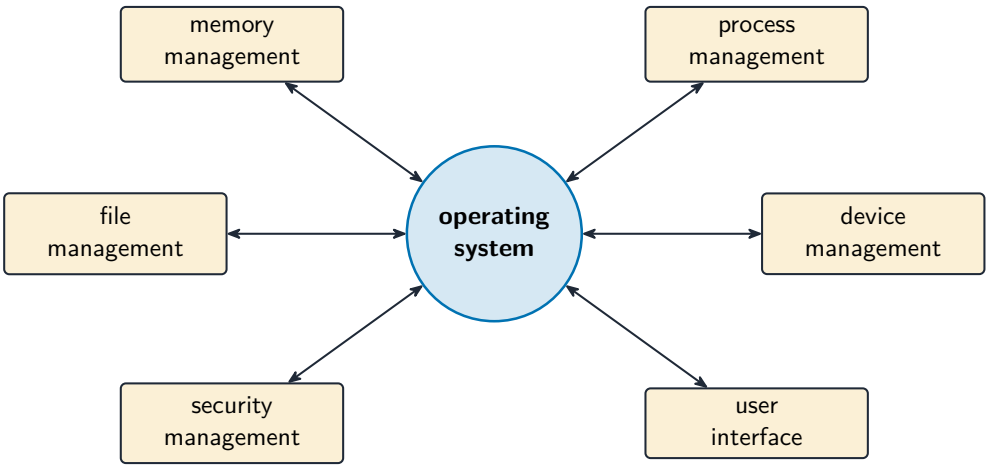
1. It provides an **interface** between the user, the application programs and the hardware.
2. It **hides the complexity** of the hardware from the user and from programs.
3. It **manages the hardware resources**: processor time, memory, storage and input/output devices.
4. It **loads and runs** application programs, giving every program the same platform.
5. It lets several programs run at once (**multitasking**) and keeps them, and the users' data, **secure**.

"It runs the computer" or "it manages resources" alone earns nothing: each mark is one named job.

■ The five key management tasks

The syllabus names five tasks. A "describe" question wants what the OS **does**: one named task and one action for each mark.

task	what the OS does
memory management	allocates memory to each program; keeps programs apart; frees memory when a program ends
process management	creates and ends **processes**; decides which process gets the processor next; switches between them
hardware management	controls input/output devices through **device drivers**; buffers data to slow devices; responds to interrupts
file management	creates, names, copies, moves and deletes files; keeps the **directory** structure and where each file is stored; enforces **access rights**
security management	user accounts and passwords; access rights; encryption; security updates; a log of who did what

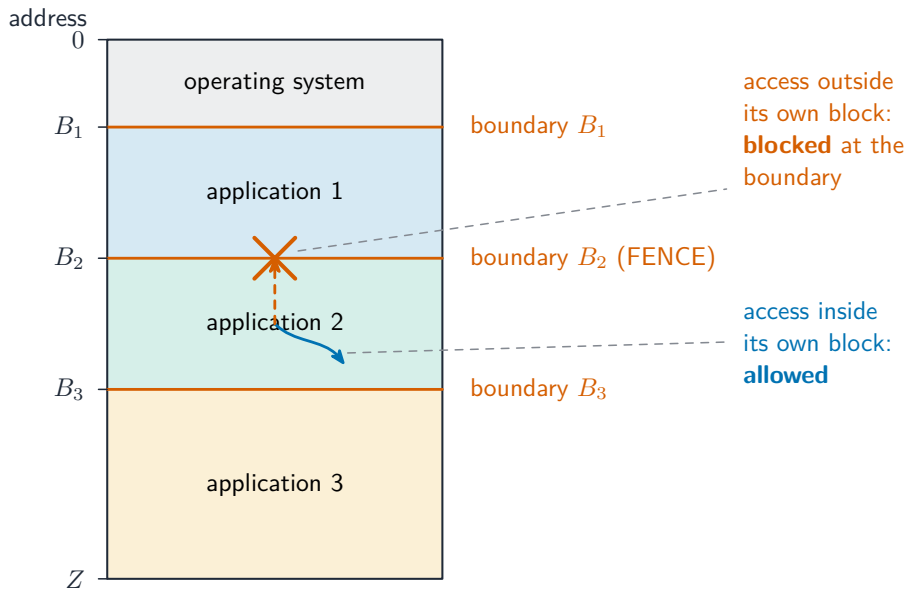


The OS also provides the **user interface** and manages the network connection.

■ Multitasking: memory and process management together

Explain how memory management and process management support multitasking (4 marks):

1. Memory management loads several programs into memory at the same time, each in its **own protected area**.
2. It keeps track of which addresses belong to which program, so one cannot overwrite another.
3. Process management shares the processor: each process runs for a **time slice** , then the OS saves its state and switches to the next.
4. The switching is so fast that all the programs appear to run at once.



■ Interrupts

An **interrupt** is a signal that makes the processor stop what it is doing and run the OS's interrupt handler. Hardware management deals with them.

kind	comes from	examples
hardware interrupt	a device	a key is pressed; the printer is out of paper; a disk transfer has finished; a power failure
software interrupt	a running program	division by zero; an invalid instruction; a program tries to use memory it does not own; a request for an OS service

Ask: *did a device cause it, or did a program's own instruction cause it?*

Past-paper practice: complete sheets 5.1 and 5.1.1 before attempting past-paper set 5.1.1.

2 Practice

Now use the methods above. The first questions need one idea each.

2.1 Complete the sentences. Use words from the list. Each word is used once. [4]

operating system

hardware · *interface* · *resources* · *secure*

The operating system provides an ..... between the user and the ..... . It manages the computer's ..... , such as memory and processor time, and keeps the users' data ..... .

2.2 Name the management task that carries out each job. [5]

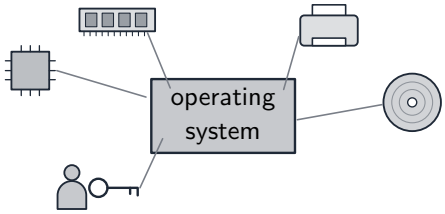
(a) decides which process gets the processor next

(b) keeps each program's data in its own area of memory

(c) uses a device driver to send data to a printer

(d) keeps a record of where each file is stored on the disk

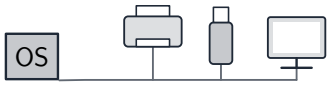
(e) checks a user's password when they log in



2.3 Tick (✓) one box in each row to show whether the event is a hardware interrupt or a software interrupt. [3]

Event	hardware interrupt	software interrupt
A key on the keyboard is pressed.		
A program divides a number by zero.		
The printer runs out of paper.		
A program asks the OS to open a file.		
A disk drive finishes transferring data.		
A program tries to use memory it does not own.		

2.4 State **two** tasks performed by hardware management. [2]



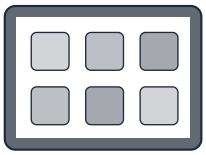
2.5 State **two** tasks performed by security management. [2]

2.6 State what is meant by **multitasking**. [1]

3 Exam-style questions

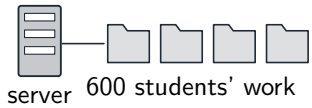
3.1 A new tablet computer is sold with an operating system already installed.

Explain why the tablet cannot run its apps without an operating system. [4]



3.2 A school server stores the work of 600 students.

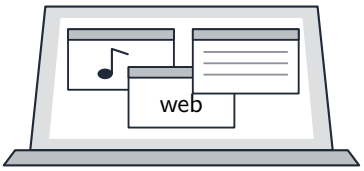
(a) **Describe three** file management tasks that the operating system carries out on this server. [3]



(b) Two students in different classes each save a file called `essay.docx`. **Explain** how file management keeps the two files apart. [2]

3.3 A student has a music player, a web browser and a word processor open at the same time on a laptop.

Explain how memory management and process management allow the three programs to run at the same time. [4]



3.4 While the student is typing, the laptop's battery becomes critically low.

(a) **State** whether this event causes a hardware interrupt or a software interrupt. Give a reason. [2]

(b) **Describe** what the operating system does when the interrupt occurs. [3]

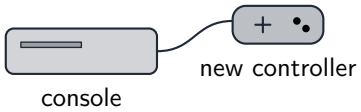
3.5 An online shop runs on a server that many employees log in to.

(a) **Describe two** security management tasks that the operating system performs on this server. [4]

(b) **Explain** why these tasks are carried out by the operating system rather than by each application program. [2]

3.6 A games console has an operating system. When a player plugs in a new controller, the console recognises it and the game can use it at once.

Describe two hardware management tasks the operating system performs for the new controller. [4]



Solutions

Each answer shows the steps, then the **result**.

2.1

- interface; hardware; resources; secure

2.2

(a) process management

(b) memory management

(c) hardware management

(d) file management

(e) security management

2.3

- a key press, the printer and the disk drive are devices: **hardware** interrupts
- division by zero, a request to the OS and an illegal memory access come from the program's own instructions: **software** interrupts

Event	hardware interrupt	software interrupt
A key on the keyboard is pressed.	✓	
A program divides a number by zero.		✓
The printer runs out of paper.	✓	
A program asks the OS to open a file.		✓
A disk drive finishes transferring data.	✓	
A program tries to use memory it does not own.		✓

2.4 any **two** of:

- installs and uses device drivers so that peripherals work
- controls the input and output devices, and buffers data going to a slow device
- responds to interrupts from devices
- shares one device between several programs

2.5 any **two** of:

- creates user accounts and checks passwords (authentication)
- sets and enforces access rights on files and folders
- encrypts stored data; installs security updates; keeps a log of what users do

2.6

- several programs are held in memory and appear to run at the same time, the processor switching between them

3.1 any **four** of:

- the hardware only fetches and executes instructions; the OS is the interface between it and the apps
- the OS hides the hardware's complexity: an app needs no code to drive the screen or the storage
- the OS loads each app into memory and gives it processor time
- the OS shares memory, processor and devices between the open apps
- the OS provides the user interface from which apps are started

3.2

(a) any **three** of:

- creates, names, saves, copies, moves and deletes files and folders
- keeps the directory structure and a record of where each file is stored on the disk
- allocates disk space to a file when it is saved, and frees it when the file is deleted
- enforces each user's access rights: read, write or execute

(b) each student's file is stored in a **different directory** (folder)

- a file is identified by its full path, so two files with the same name in different directories are different files

3.3

- memory management loads all three programs into memory at the same time, each in its own area
- it keeps the areas apart, so the browser cannot overwrite the word processor's data
- process management gives each process a time slice of the processor in turn, saving the state of one process and switching to the next
- the switching is so fast that the music keeps playing while the student types

3.4

(a) a **hardware** interrupt

- it comes from a device (the battery / power circuit), not from an instruction in a program

(b) the processor stops the current process at the end of the current instruction, and the OS saves its state

- the OS runs the interrupt handler for this interrupt: it warns the user, saves data or puts the laptop to sleep
- when the handler has finished, the OS restores the saved process and it continues

3.5

(a) any **two**, each described:

- authentication: every employee has an account and must give the correct password to log in
- access rights: each employee can only read or change the files their job needs; the OS blocks other access
- the OS encrypts stored data, keeps a log of who did what, and installs security updates

(b) the OS controls all the hardware and every file, so it can enforce the rules for every program

- a rule inside one application could be bypassed by another program, or by a program written to ignore it

3.6 any **two**, each described:

- it finds and loads the **device driver** for the controller, which translates the game's commands into the signals the controller understands
- it responds to the **interrupts** the controller sends when a button is pressed, and passes the input to the game
- it allocates the controller to the game and buffers its input so that no press is lost