Past-paper walk-through

Purposes of an Operating System (OS) ▪ 16.1

eXercise

Questions in this set

- 9618/31 N25 Q4 ▪ 6 marks
- 9618/32 J25 Q5 ▪ 7 marks
- 9618/33 J25 Q6 ▪ 6 marks
- 9618/32 N24 Q9 ▪ 5 marks
- 9618/33 N24 Q8 ▪ 7 marks

Reading the mark scheme

- **B1** a mark that stands on its own – the point is either made or not.
- **M1** a *method* mark: the right approach, even if the number is wrong.
- **A1** an *answer* mark, and it usually needs its M mark first.
- **C1** a working mark on the way to the answer.
- // separates answers the examiner accepts equally.
- **ecf** = error carried forward: a later part is marked on your own earlier answer.
- **owtte** = "or words to that effect" – the wording need not match.

Question 4

9618/31 N25 ▪ Q4 ▪ 6 marks

A scheduling routine determines how processes are managed by the operating system.

(a) Identify **two** scheduling routines.

1. 2. **[2]**

(b) Describe **two** ways in which the complexities of the computer hardware are hidden from the user.

1. 2. **[4]**

Solution 4

(a)

Mark scheme

- One mark per scheduling routine (Max 2) from:
- ▪ Round robin
- ▪ Shortest job first
- ▪ First come first served
- ▪ Shortest remaining time

Solution 4

(b) Mark scheme

- One mark for identification and one mark for a description (Max 4)
- Two from:
- Provision of a User Interface // Provision of a Graphical User Interface [1]
- Allows the user to interact with the computer in a more intuitive way // Icons and menus are used to control devices by simply 'pointing and clicking' [1]
- Use of device drivers [1]
- Makes it easier to control peripherals such as printers within the operating system of the computer rather than on the separate device itself [1]

Solution 4

- Device mapping [1]
- Different devices (physical and virtual) are easy to identify on the network, check their status, or use [1]
- The user interacts only with the Application / top layer (of the TCP/IP protocol suite) [1] leaving the lower layers and their complexities hidden from the user [1]

Question 5

9618/32 J25 ▪ Q5 ▪ 7 marks

The management and scheduling of processes are tasks carried out by an operating system.

(a) Identify **three** process states.

1. 2. 3. **[3]**

(b) Describe the function of the shortest job first scheduling routine **and** give a benefit of this routine.

Function

Benefit **[4]**

Solution 5

(a)

Mark scheme

- One mark per point
- ▪ Running state
- ▪ Ready state
- ▪ Blocked state

Solution 5

(b) Mark scheme

- One mark per mark point (Max 3)
- MP1
- The processes are queued as they arrive
- MP2
- The process with the shortest time to complete/burst time is selected first and executed
- MP3

Solution 5

- The process will continue until complete or put in a waiting state once execution has begun // it is non-pre-emptive
- MP4
- The scheduler will continue to choose shorter processes over longer processes if they continue to be added to the queue can cause starvation for longer jobs
- One mark for a benefit (Max 1)
- MP5
- Shorter jobs don't have to wait for longer jobs to complete before processing // Significantly reduces the average overall waiting time for processes // Ensures starvation doesn't occur for process with shorter burst times
- MP6

Solution 5

- Higher throughput of processes

Question 6

9618/33 J25 ▪ Q6 ▪ 6 marks

The management and scheduling of processes are tasks carried out by an operating system.

(a) Describe **one** reason why scheduling is necessary in process management. **[2]**

(b) Explain the function of the round robin scheduling routine **and** give a benefit of this routine. **[4]**

Solution 6

(a)

Mark scheme

- One mark for each mark point (Max 2)
- ▪ Process scheduling is required to ensure that all processes are executed in a timely manner
- ▪ ... and enables multitasking/multiprogramming/multiprocessing
- ▪ ... to minimise CPU idle time
- ▪ ... to ensure that no process is starved of resources. // ... to ensure fair access to resources.
- ▪ ... ensures jobs/processes are completed in order of priority.

Solution 6

(b) Mark scheme

- One mark for each mark point (Max 3)
- MP1
- Processes are queued as they arrive.
- MP2
- It is a pre-emptive scheduling routine.
- MP3
- A fixed time quantum is given to each process. // Each process has an equal time slice.

Solution 6

- MP4
- When a time slice ends, the status of the process is saved/queued so it can continue from where it left off in its next time slice.
- MP5 and the next process is executed for its time slice; its previous state is reinstated/restored, if applicable.
- MP6
- If a process completes within its time slice, the next process is executed for its time slice.
- One mark for benefit (Max 1)
- MP7
- Reduces average response time by limiting each process to a fixed amount of time.

Solution 6

- MP8
- No issue with starvation of resources.

Question 9

9618/32 N24 ▪ Q9 ▪ 5 marks

9 **(a)** The kernel is the central component of an Operating System (OS).

Outline how the kernel of an OS acts as an interrupt handler.

Question 9

(b) (i) State what is meant by the term **multi-tasking** in an Operating System.

Question 9

(ii) Describe how multi-tasking is implemented in an Operating System.

Question 9

Solution 9

(a) Worked steps

Follow one interrupt from the moment it is raised. Two marks from:

- The kernel **receives a signal** when an interrupt is generated.
- It **checks the priority** of the new interrupt against what is currently running.
- The system **enters kernel mode** if the interrupt has the higher priority.
- The kernel consults the **interrupt dispatch table** to find the right service routine ...
- ... having **saved the state of the interrupted process** — the register contents — on the kernel stack.
- Once the interrupt has been serviced, the kernel **restores that state** and the interrupted process resumes.

Solution 9

Two marks means two points, but choose ones that are consecutive in that sequence rather than two ways of saying "it deals with the interrupt". The save-and-restore pair is the strongest choice: it is what makes an interrupt invisible to the program it interrupted. **Mark scheme**

- One mark per mark point (Max 2)
- MP1 the kernel receives a signal when an interrupt is generated
- MP2 the kernel checks the priority and reviews the status/priority of the current interrupts
- MP3 system enters kernel mode if the type of interrupt is of higher priority than the current process

Solution 9

- MP4 the kernel consults the interrupt dispatch table / IDT
- MP5 ... and saves the state of the interrupted process / contents of the registers on the kernel stack
- MP6 the kernel restores the process state e.g. contents of registers once the interrupt is serviced
- 2

Solution 9

(b)(i)

Worked steps

One mark, one sentence.

- Multi-tasking allows a computer to carry out — or **appear to carry out — more than one process at a time**.

The hedge is worth including. On a single-core processor nothing is truly simultaneous; the processor switches between tasks fast enough that they seem concurrent.

Mark scheme

- One mark per mark point (Max 1)
- MP1 multi-tasking allows computers to carry out / seem to carry out more than one process at a time
- 1

Solution 9

(b)(ii) Worked steps

Two marks, and they are about the *mechanism*, not the effect.

- **Processor time and hardware resources are shared** between the tasks.
- **Scheduling** decides which process runs next, so the tasks proceed efficiently and without clashing over resources.
- A task of **higher priority can interrupt** one that is currently running.

The link back to part (a) is worth making: it is the interrupt mechanism that makes multi-tasking possible at all, because it is what takes the processor away from a running process and hands it to the scheduler. **Mark scheme**

Solution 9

- One mark per mark point (Max 2)
- MP1 processor time/common hardware and resources is/are shared between tasks
- MP2 scheduling is used to decide on the processes to be carried out to ensure multi-tasking operates correctly / efficiently
- / without clashes
- MP3 one task of a higher priority can interrupt another task that is currently running
- 2

Question 8

9618/33 N24 ▪ Q8 ▪ 7 marks

8 **(a)** Describe the process of **segmentation** for memory management.

Question 8

(b) Explain what is meant by **disk thrashing**.

Question 8

Solution 8

(a) Worked steps

Four marks. Describe segmentation as a process, from division through to address translation.

- The logical (virtual) address space is divided into **variable-sized blocks called segments**, each corresponding to a logical part of the program.
- Each segment has a **name and a size**, and segments are **numbered**.
- During execution, segments are **loaded from virtual memory into physical memory**.
- An address is given as a **segment number and an offset** within that segment.
- The segment number is an **index into the segment map table**, which maps the logical address to a physical one.

Solution 8

The contrast to make, since it is what the examiner is checking you understand: **paging divides memory into fixed-size pages** decided by the hardware, whereas **segmentation divides it into variable-size segments** that follow the program's own structure — a procedure, an array, a data area. That is why a segment has a name and a page does not. **Mark scheme**

- One mark per mark point (Max 4)
- MP1 In segmented memory, the logical / virtual address space is broken into varying sized blocks called segments /
- sections.
- MP2 Each segment has a name and size.

Solution 8

- MP3 During execution segments from logical / virtual memory are loaded into physical memory.
- MP4 The address is specified by the user
- MP5 ... it contains the segment name and offset value.
- MP6 Segments are numbered
- MP7 ... and this number is used as an index in the segment map table.
- MP8 The offset value determines the size of the segment.
- MP9 A segment map table maps logical / virtual addresses to physical addresses / contains the segment number and
- offset.
- 4

Solution 8

- 9618/33
- October/November 2024

Solution 8

(b) Worked steps

Three marks, and they form one causal chain — give it in order.

- Disk thrashing is a problem that arises when **virtual memory** is in use.
- As main memory fills, **more and more pages must be swapped** in and out.
- The swapping causes a **very high rate of disk access** and excessive head movement.
- Moving a disk head is **slow** compared with memory access, so each swap costs a great deal of time.
- Eventually **more time is spent swapping than processing**, and the program slows to a halt or appears to freeze.

Solution 8

State the **consequence** as well as the cause. "Pages are swapped frequently" is the middle of the chain; the answer the question wants ends with the machine doing more paging than work. **Mark scheme**

- One mark per mark point (Max 3)
- MP1 Disk thrashing is a problem that may occur when virtual memory is being used.
- MP2 As the main memory fills up, more and more pages need to be swapped in and out of virtual memory.
- MP3 This swapping leads to a very high rate of hard disk access / excessive disk head movements.

Solution 8

- MP4 Moving a hard disk read/write head takes a relatively long time / long latency time.
- MP5 Eventually, more time is spent swapping pages than processing data thrash point, which can cause the program to
- freeze or not run.
- 3
- 9618/33
- October/November 2024