A-Level Computer Science: **w25 31**

4(a)	**One** mark per scheduling routine (**Max 2**) from: • Round robin • Shortest job first • First come first served • Shortest remaining time
4(b)	**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] 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]

A-Level Computer Science: **s25 32**

5(a)	**One** mark per point • Running state • Ready state • Blocked state
5(b)	**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 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 Higher throughput of processes

A-Level Computer Science: **s25 33**

6(a)	**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.

A-Level Computer Science: **s25 33**

6(b)	**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. 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. MP8 No issue with starvation of resources.

A-Level Computer Science: **w24 32**

9(a)	**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 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
9(b)(i)	**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
9(b)(ii)	**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

A-Level Computer Science: **w24 33**

8(a)	**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. 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.
8(b)	**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. 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.