1993 Paper 3 Question 6

Operating System Functions

Describe the functionality you would expect to find in the file system directory service of a multi-user operating system. [10 marks]

Describe two ways in which multiple names for the same file can be supported, and what problems arise as a result. [10 marks]

1993 Paper 4 Question 7

Operating System Functions

In relation to virtual memory, describe the terms *segment*, *page* and *translation lookaside buffer* (TLB). [6 marks]

The operating system for a microprocessor supports a virtual memory model which implements both segmentation and paging. The only hardware assistance for the virtual memory system in the microprocessor is an on-chip TLB.

Outline the data structures held by the operating system. [5 marks]

Describe the actions of the operating system in response to an address exception due to not matching the address issued by the processor in the TLB. [5 marks]

How can the operating system use access permissions to aid its page replacement policy? [4 marks]

1994 Paper 3 Question 4

Operating System Functions

Describe the use of an *inverted page table* for the implementation of virtual addressing and a paging virtual memory system. Give details of the operation of the page table and its associated data structures. [15 marks]

What can be done to overcome the thrashing problem encountered with a direct mapped inverted page table? [5 marks]

1994 Paper 3 Question 5

Operating System Functions

What is the *Access Control matrix*? Describe the *capability* and *access control list* views of this matrix. [10 marks]

In a capability-based system, describe the techniques which can be used to protect the capabilities from unauthorised modification. [10 marks]

1994 Paper 4 Question 4

Operating System Functions

Describe the use of a *table of pointers* and a *table of extents* to store the list of disc blocks which compose a file. Pay particular attention to how such data structures can be designed to enable the efficient support of very large files, and give estimates of their performance for finding the location of a block chosen at random.

[14 marks]

Describe how such data structures would be used in the implementation of a *log-structured* file system with consideration of the impact of garbage collection.

[6 marks]

1995 Paper 3 Question 7

Operating System Functions

What properties of file access and of discs does a *log-structured* file system use to optimise file system performance? Describe scenarios in which the relevant assumptions are valid and invalid. [8 marks]

Describe the operation of such a file system, including the actions of the *sweep*, or *garbage collection*, stage, and how the system recovers from crashes. [12 marks]

1995 Paper 4 Question 7

Operating System Functions

An operating system for a *smart card* is required — that is a common run-time system for the smart card, which hosts some set of applications loaded into the card at the time the card is issued.

What operations should the memory management component of the operating system support in such a system? [12 marks]

The hardware engineers are concerned about complex memory management hardware consuming too much power, but the customer demands memory protection between the applications. Describe some solutions that could meet these demands. [8 marks]

1996 Paper 3 Question 8

Operating System Functions

A computer with a 32-bit virtual addressing scheme uses pages of size 4 Kbyte. Describe, with the aid of diagrams, two practical schemes for managing its virtual address space, comparing them with regard to speed of access, efficiency (of space), and ease of memory sharing between processes. [10 marks]

A Winchester-style disc has its head currently located at track 100, and the head is moving towards track 0. Given the reference string (27, 129, 110, 186, 147, 41, 10, 64, 120, 11, 8, 10) representing the (ordered) sequence of requests for disc tracks, give the sequence of disc addresses visited by the disc head under the SSTF, SCAN and C-SCAN disc scheduling algorithms. In each case briefly describe the algorithm, and compute the average cost of a disc access in terms of the mean number of tracks traversed per access.

In what way is each of these algorithms biased in its service of disc requests? Describe an algorithm which reduces the bias. [10 marks]

1996 Paper 4 Question 7

Operating System Functions

What is meant by the term *demand paging* in a virtual memory management system, and how is it implemented? [5 marks]

Briefly describe five techniques which the operating system and/or hardware can implement to improve the efficiency of demand paging. [5 marks]

What is the *working set* of a program, and how can an operating system use it in the management of virtual memory? [3 marks]

Describe the clock (second chance) algorithm for selecting a VM page for replacement when a page fault occurs. How is the performance of this algorithm affected by the memory size of the computer system, and how may this be avoided? [7 marks]

1997 Paper 3 Question 8

Operating System Functions

A computer is equipped with a CPU with a 32-bit virtual address space, a 32-entry TLB with access time 10ns and 32 Mbyte DRAM with access time 100ns. Its secondary storage is provided by an IDE hard disc with transfer rate 1 Mbyte/s, and which rotates at 3600 revolutions per minute and has an average seek time of 10ms. The computer uses demand paged virtual memory with 1 kbyte pages.

(a) Explain the function of the TLB with the aid of a diagram. [4 marks]

(b) Design a page table structure which the operating system can use to implement virtual memory on this system and describe how a virtual address is translated using it. Are there any drawbacks of your approach? [4 marks]

(c) What is *demand paging*? Briefly describe a policy which the operating system can use to share the available physical memory between competing processes. [3 marks]

(d) Calculate the effective memory access time for the system if virtual memory is managed using the scheme described in your answer to (b), if page tables are kept locked in memory, the probability of finding a translation in the TLB is 98%, and the probability of a page fault is 10^{-6}. Exclude operating system overhead from your calculation, and briefly explain your answer. [9 marks]

1997 Paper 4 Question 7

Operating System Functions

In the management of virtual memory, what is *thrashing*, and how does it occur? [5 marks]

What is the *working set* of a process, and how can it be computed? [5 marks]

List *five* techniques that can be used in an operating system to improve the performance of demand paged virtual memory. [5 marks]

What is a *capability*, and how can it be used for access control in a computer system? [5 marks]

1998 Paper 3 Question 8

Operating System Functions

Multimedia applications may be characterised by their requirement for timely completion of their computation or data processing in order to function usefully. Why are traditional operating systems unable to deliver such guarantees? Comment on the features which an operating system should implement in order to support such applications successfully, considering issues of memory management, device access, and scheduling techniques. [20 marks]

1998 Paper 4 Question 7

Operating System Functions

What is a *translation lookaside buffer* (TLB)? Describe its operation with the aid of a diagram. How is the TLB affected by processor context switches for (*i*) threads and (*ii*) processes? [10 marks]

A process has four page frames allocated to it. (All of the following numbers are decimal, and all numbers start from zero.) The time of the last loading of a page into each page frame, the time of last access to the page, the virtual page number in each frame and the Referenced (R) and Modified (M) bits for each page frame are shown in the table below. Times are in clock ticks from the process start time at time 0.

Virtual Page #	Frame #	Time Loaded	Time Referenced	M	R
2	0	60	161	0	1
1	1	130	160	0	0
0	2	26	162	1	0
3	3	20	163	1	1

A page fault to virtual page 4 has occurred. Which page frame will have its contents replaced under each of the following replacement algorithms? Briefly explain why in each case.

(*a*) FIFO

(*b*) LRU

(*c*) Second Chance (Clock)

(*d*) Enhanced Second Chance

[6 marks]

Given the above state of memory before the fault, and the reference string of virtual page numbers: (4, 0, 0, 0, 2, 4, 2, 1, 0, 3, 2), calculate how many page faults would occur under the LRU policy if a working set with a window size of 4 were used instead of a fixed allocation of 4 frames. Show clearly when each page fault would occur. [4 marks]

1999 Paper 3 Question 8

Operating System Functions

FIFO, LRU, and Clock are three page replacement algorithms.

(*a*) Briefly describe the operation of each algorithm. [6 marks]

(*b*) The Clock strategy assumes some hardware support. What could you do to allow the use of Clock if this hardware support were not present? [2 marks]

(*c*) Assuming good temporal locality of reference, which of the above three algorithms would you choose to use within an operating system? Why would you not use the other schemes? [2 marks]

What is a *buffer cache*? Explain why one is used, and how it works. [6 marks]

Which buffer cache replacement strategy would you choose to use within an operating system? Justify your answer. [2 marks]

Give *two* reasons why the buffering requirements for network data are different from those for file systems. [2 marks]

1999 Paper 4 Question 7

Operating System Functions

The following are three ways which a file system may use to determine which disk blocks make up a given file.

(a) chaining in a map

(b) tables of pointers

(c) extent lists

Briefly describe how each scheme works. [3 marks each]

Describe the benefits and drawbacks of using scheme (c). [6 marks]

You are part of a team designing a distributed filing system which replicates files for performance and fault-tolerance reasons. It is required that rights to a given file can be revoked within T milliseconds ($T \geqslant 0$). Describe how you would achieve this, commenting on how the value of T would influence your decision. [5 marks]

2000 Paper 3 Question 7

Operating System Functions

Why are the scheduling algorithms used in general-purpose operating systems such as Unix and Windows NT not suitable for real-time systems? [4 marks]

Rate monotonic (RM) and *earliest deadline first* (EDF) are two popular scheduling algorithms for real-time systems. Describe these algorithms, illustrating your answer by showing how each of them would schedule the following task set.

Task	*Requires Exactly*	*Every*
$\mathcal{A}$	2ms	10ms
$\mathcal{B}$	1ms	4ms
$\mathcal{C}$	1ms	5ms

You may assume that context switches are instantaneous. [8 marks]

Exhibit a task set which is schedulable under EDF but not under RM. You should demonstrate that this is the case, and explain why.
[Hint: consider the relationship between task periods.] [8 marks]

2000 Paper 4 Question 7

Operating System Functions

Why is it important for an operating system to schedule disc requests? [4 marks]

Briefly describe each of the SSTF, SCAN and C-SCAN disc scheduling algorithms. Which problem with SSTF does SCAN seek to overcome? Which problem with SCAN does C-SCAN seek to overcome? [5 marks]

Consider a Winchester-style hard disc with 100 cylinders, 4 double-sided platters and 25 sectors per track. The following is the (time-ordered) sequence of requests for disc sectors:

$$\{\ 3518,\ 1846,\ 8924,\ 6672,\ 1590,\ 4126,\ 107,\ 9750,\ 158,\ 6621,\ 446,\ 11\ \}$$

The disc arm is currently at cylinder 10, moving towards 100. For each of SSTF, SCAN and C-SCAN, give the order in which the above requests would be serviced. [3 marks]

Which factors do the above disc arm scheduling algorithms ignore? How could these be taken into account? [4 marks]

Discuss ways in which an operating system can construct logical volumes which are (a) more reliable and (b) higher performance than the underlying hardware. [4 marks]

2001 Paper 3 Question 7

Operating System Functions

(*a*) What are the key issues with scheduling for shared-memory multiprocessors? [3 marks]

(*b*) *Processor affinity*, *take scheduling* and *gang scheduling* are three techniques used within multiprocessor operating systems.

(*i*) Briefly describe the operation of each. [6 marks]

(*ii*) Which problem does the processor affinity technique seek to overcome? [2 marks]

(*iii*) What problem does the processor affinity technique suffer from, and how could this problem be overcome? [2 marks]

(*iv*) In which circumstances is a gang scheduling approach most appropriate? [2 marks]

(*c*) What additional issues does the virtual memory management system have to address when dealing with shared-memory multiprocessor systems? [5 marks]

2001 Paper 4 Question 7

Operating System Functions

(*a*) In the context of virtual memory management:

(*i*) What is *demand paging*? How is it implemented? [4 marks]

(*ii*) What is meant by *temporal locality of reference*? [2 marks]

(*iii*) How does the assumption of temporal locality of reference influence page replacement decisions? Illustrate your answer by briefly describing an appropriate page replacement algorithm or algorithms. [3 marks]

(*iv*) What is meant by *spatial locality of reference*? [2 marks]

(*v*) In what ways does the assumption of spatial locality of reference influence the design of the virtual memory system? [3 marks]

(*b*) A student suggests that the virtual memory system should really deal with "objects" or "procedures" rather than with pages. Make arguments both for and against this suggestion. [4 and 2 marks respectively]

.

2003 Paper 3 Question 5

Operating Systems II

Modern server-class machines often use a Redundant Array of Inexpensive Disks (RAID) to provide non-volatile storage.

(a) What is the basic motivation behind this? [2 marks]

(b) Describe RAID level 0. What are the benefits and drawbacks of this scheme? [3 marks]

(c) Describe RAID level 1. What are the benefits and drawbacks of this scheme? [3 marks]

(d) Compare and contrast RAID levels 3, 4 & 5. What problem(s) with the former pair does the last hope to avoid? [6 marks]

A server machine has k identical high-performance IDE disks attached to independent IDE controllers. You are asked to write operating system software to treat these disks as a RAID level 5 array containing a single file system. Your software will include routines to read file-system data, write file-system data, and to schedule these read and write requests. What difficulties arise here and how may they be addressed? [6 marks]

2003 Paper 4 Question 5

Operating Systems II

(*a*) Describe the basic operation of a *log-structured file system*. What are the potential benefits? What are the problems? [8 marks]

(*b*) Several modern file systems make use of *journalling*. Describe how a journal is used by the file system, and the situations in which it is beneficial. [6 marks]

(*c*) You are assigned the task of designing a tape backup strategy for an important file server. The goal is to maximise the time any file is held on backup while minimising the number of tapes required. Sketch your strategy, commenting on the support you require from the file system, and justifying your design decisions. [6 marks]

2004 Paper 3 Question 5

Operating Systems II

(*a*) What problem do real-time scheduling algorithms try to solve? [2 marks]

(*b*) Describe one *static priority* and one *dynamic priority* real-time scheduling algorithm. You should discuss the issue of admission control, and comment on the data structures that an implementation would need to maintain and on how these would be used to make scheduling decisions. [8 marks]

(*c*) A designer of a real-time system wishes to have concurrently executing tasks share a data structure protected by a mutual exclusion lock.

(*i*) What scheduling problem could arise here? [2 marks]

(*ii*) How could this problem be overcome? [2 marks]

(*d*) The designer also wishes the real-time system to use demand paged virtual memory for efficiency. What problems could arise here, and how could they be overcome? [6 marks]

2004 Paper 4 Question 5

Operating Systems II

(*a*) Most conventional hardware translates virtual addresses to physical addresses using *multi-level page tables* (MPTs):

(*i*) Describe with the aid of a diagram how translation is performed when using MPTs. [3 marks]

(*ii*) What problem(s) with MPTs do *linear page tables* attempt to overcome? How is this achieved? [3 marks]

(*iii*) What problems(s) with MPTs do *inverted page tables* (IPTs) attempt to overcome? How is this achieved? [3 marks]

(*iv*) What problems(s) with IPTs do *hashed page tables* attempt to overcome? How is this achieved? [3 marks]

(*b*) Operating systems often cache part of the contents of the disk(s) in main memory to speed up access. Compare and contrast the way in which this is achieved in (*i*) 4.3 BSD Unix and (*ii*) Windows 2000. [8 marks]

2005 Paper 3 Question 5

Operating Systems II

(*a*) Modern operating systems typically support both *threads* and *processes*. What is the basic difference between a thread and a process? Why do operating systems support both concepts? [2 marks]

(*b*) You get a summer job with a company which has an in-house operating system called sOs. sOs uses static priority scheduling, supports at most 32 concurrently-executing processes, and works only on uniprocessor machines. Describe with justification how you would modify sOs in order to:

(*i*) support up to 50000 concurrently executing processes; [2 marks]

(*ii*) reduce or eliminate the possibility of starvation; [3 marks]

(*iii*) efficiently schedule processes on an 8 CPU symmetric multiprocessor (SMP) machine; [5 marks]

(*iv*) support threads in addition to processes on SMP machines. [3 marks]

(*c*) How would you go about reducing the time taken to boot a modern operating system? [5 marks]

2005 Paper 4 Question 5

Operating Systems II

(*a*) Scheduling disk requests is important to reduce the average service time. Describe briefly how the SSTF, SCAN and C-SCAN scheduling algorithms work. [3 marks]

(*b*) Recently there have been proposals that *2-D disk scheduling* algorithms should be used. What is the basic idea behind 2-D disk scheduling? [2 marks]

(*c*) You are asked to develop support for 2-D disk scheduling in a commodity operating system.

(*i*) What would be the major difficulty that you face? [2 marks]

(*ii*) Sketch a design for how you would go about overcoming this difficulty, and comment on how well you think the resulting system would work. [8 marks]

(*d*) Several modern file-systems and databases make use of a *journal* to aid in crash recovery.

(*i*) Briefly describe how journalling helps crash recovery. [2 marks]

(*ii*) A researcher suggests adding 128 MB of NVRAM to a disk drive and using this to store the journal. Discuss the advantages and disadvantages of this approach. [3 marks]

2006 Paper 3 Question 5

Operating Systems II

Most modern computers include a *translation lookaside buffer* (TLB) to speed up address translation.

(*a*) Describe with the aid of a diagram the basic operation of a TLB. [2 marks]

(*b*) Some TLBs support *process tags* (sometimes called *address space numbers*). Explain how process tags could be used by an operating system, and what benefits might be expected. [2 marks]

(*c*) Some TLBs support *superpages* of one or more sizes.

(*i*) Explain how superpages could be used to reduce the TLB footprint for operating system kernels. [2 marks]

(*ii*) Explain how superpages could be used to reduce the TLB footprint for processes. What additional considerations need to be taken into account in this case? How well do you think your scheme would work in practice? [4 marks]

(*d*) Compare and contrast the way in which demand paging is performed in Unix and VMS. [10 marks]

2006 Paper 4 Question 5

Operating Systems II

In the early 1980s, a new file-system called the *fast file-system* (FFS) was designed for BSD Unix.

(*a*) What *two* problems with the original Unix file system was FFS designed to overcome? [2 marks]

(*b*) For *each* of these problems, describe the solution used, and comment on its effectiveness. [6 marks]

(*c*) From the modern perspective, how appropriate or effective do you believe *each* of these solutions is? Justify your answer. [2 marks]

(*d*) Sketch an efficient design for a file-system to support (soft) real-time access to large multi-media files. Include details of how you would lay out and access files and directories, and how you would perform integrity checking. Be sure to justify any key differences from conventional designs, and to state any assumptions that you make. [10 marks]