## 1993 Paper 2 Question 10

What is an operating system *kernel*, and what runtime services does an executable binary program expect from a typical time-sharing kernel?

How may transfer of control (program flow) between the loaded application and the resident kernel be achieved?

## 1993 Paper 2 Question 11

Describe and compare the following two programming approaches:

($a$) execution of a compiled and linked set of modules;

($b$) using an interpreter.

Include examples of each, showing when one method is preferable to the other. Is there an absolute distinction between the two methods or is there a spectrum of approaches?

# 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 2 Question 11

You are in charge of the design of both hardware and software for a new (but fairly conventional) workstation which will have its peripherals (for example a disc drive and a printer) directly connected to it. Your workstation will be required to support an operating system that allows its user(s) to run several independent programs at once.

Explain and justify the method that you will provide for an application program to gain access to kernel services and physical input/output devices. [12 marks]

For both a printer and a disc drive, indicate additional functions (beyond those provided by the physical device itself) that an operating system typically provides in software. [8 marks]

## 1994 Paper 2 Question 12

If a program is written cautiously and in a suitable high level language then it can be "portable", and one set of source files can be used with little or no change on a wide variety of computers and with many different operating systems. Explain the steps that are taken to turn the source version of such a program into a runnable version of the application that it represents. Indicate all the places where programs or pieces of code not derived from the portable sources are involved. [4 marks]

Identify which parts of the software preparation path (if any) will need to be altered in each of the following cases, commenting on just where programs and code (not being directly part of the portable source) can be used unaltered and where different versions are called for:

($a$) The program is to be run on different hardware configurations of the various models of the same computer; [4 marks]

($b$) The program is to be run on computers which share the same hardware design and have the same processor, but under different operating systems (for example some PCs will run MSDOS, Unix and several other operating systems); [4 marks]

($c$) The program is to be run on two machines that share a common operating system (such as Unix) but which have different processor designs. [4 marks]

Suppose the portable programming language involved is implemented using an interpreter rather than a compiler, and the language is already implemented on all the computer systems involved. How does this affect the amount of work and the number of changes needed when building an executable version of a program for use in many different environments? [4 marks]

# 1994 Paper 2 Question 7

A small electronic mail system has been installed in your organisation, allowing the dozen or so workers in your group (all users of a single central server computer) to send each other messages about their work. There is now talk of extending this mail system to provide contact with several hundred other sites, each of which has from a dozen to a few hundred workers. For political reasons it will be necessary to construct a new mail system for this enlarged group, rather than buy in services from some commercial organisation that provides existing hardware or software.

Discuss some of the problems that will have to be considered in the design of the new mail system. Concentrate on ways that electronic mail will interact with the body of users rather than on any fine technical details of a design. In particular consider:

($a$) Mail redirection when a user moves to a new location; [5 marks]

($b$) Circulation lists and multiple copies of mail; [5 marks]

($c$) Recovery of the cost of setting up and running the mail service; [5 marks]

($d$) The possible alternative of using existing telephone lines and installing a large number of FAX machines. [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 5

## UNIX Case Study

For the UNIX operating system, define

($a$) process [2 marks]

($b$) the execution environment of a process [3 marks]

($c$) the `fork` system call [3 marks]

Explain how a command line of the form given below is implemented:

```
command <argument-list> &
```
[9 marks]

Outline the essential difference in the implementation of command lines of the form given below:

```
command1 <arg1-list> & command2 <arg2-list>
command1 <arg1-list> | command2 <arg2-list>
```
[3 marks]

# 1995 Paper 1 Question 11

**System Design**

An application program in wide use contains a set of printable strings which are error messages and other helpful messages to the user. The program prints a selection of these messages each time it is run. Describe the route these messages have typically taken from the point where the programmer textually enters them, to the time they are displayed by the running application on an output device.

[20 marks]

# 1995 Paper 1 Question 4

**System Design**

Describe a typical address map for a Von Neuman computer. Explain how a group of four adjacent bytes are typically used to represent

($a$) a positive integer;

($b$) a signed integer;

($c$) a floating point number.

[10 marks]

# 1995 Paper 2 Question 28

**Unix Case Study**

What is an *i-node* and what information is contained in it?

Describe how named files are mapped to i-nodes.

How is the information associating disc blocks with i-nodes represented?

What restrictions are placed on name to i-node links to simplify file system recovery?

[20 marks]

# 1995 Paper 3 Question 8

**Unix Case Study**

Describe the segments that compose a Unix process and how the `exec` functions create these segments from the sections of an executable binary file. [8 marks]

Using as an example the command line `ls | wc`, describe the sequence of operations and system calls that a shell must perform to execute the commands to the point at which an `exec` function is invoked on each of `ls` and `wc`. [12 marks]

# 1995 Paper 11 Question 7

**Operating System Foundations**

Consider the operation of a scheduler in a system where there are system level and user level processes. User processes may be IO bound or CPU bound and may have user controlled (negative) priority.

Describe the data structures that the scheduler might use, including parts of process descriptors that the scheduler would operate on. [10 marks]

Describe in detail the circumstances under which the scheduler would be entered and for each different circumstance outline a scheduling algorithm that might be used. [10 marks]

# 1996 Paper 1 Question 11

**System Design**

(*a*) How does a loaded program in a general purpose timesharing computer gain access to input and output streams provided by the operating system? [5 marks]

(*b*) What happens when the program is ready to receive the next data from a keyboard input stream and no key press is ready? [5 marks]

(*c*) A CPU intensive program will tend to run for long periods of time without making requests of the operating system. What happens to other users programs when someone runs a CPU intensive program? [5 marks]

(*d*) What can be done for a shared output device such as a printer or plotter to ensure civilised results when two or more users try to use it at once? [5 marks]

# 1996 Paper 1 Question 4

**System Design**

(*a*) What are the advantages of using a high-level language for writing programs?
[7 marks]

(*b*) Why must some parts of the code which makes up an executable binary program typically have to be written in assembly language ? Who typically writes these parts and where are they normally stored in the system?
[3 marks]

# 1996 Paper 2 Question 9

## Unix Case Study

(*a*) Draw a diagram of the address space of a Unix process. Indicate which parts may be shared with other processes. [5 marks]

(*b*) Explain the Unix system call mechanism. [5 marks]

(*c*) Describe how the fork system call is implemented. Discuss the mechanism from the viewpoint of efficiency and support for sharing between families of processes. How may a parent synchronise with the termination of a child? [6 marks]

(*d*) Contrast the execution by the shell of the commands

```
c1 > file1
c2 < file1 > file2
c3 < file2
```

with the execution of the single command

```
c1 | c2 | c3
```

[4 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]

# 1996 Paper 10 Question 12

**Unix Case Study**

For a Unix filing system:

($a$) Describe the organization of the metadata for a filing system held on a single device or partition of a device. Outline any advantages and disadvantages of the approach. [4 marks]

($b$) Explain how the online filing system is constructed from the separate filing systems you have described. [2 marks]

($c$) Describe how hard links and soft (symbolic) links are implemented. Mention any restrictions on their use and any implications of their existence on other file operations. [6 marks]

($d$) Describe how the disc blocks used to store a file or directory are recorded. Discuss whether this method is suitable for a large video file. [2 marks]

($e$) Describe how access control policy is recorded and how access by program, as well as by user, is supported. Discuss this latter mechanism. [6 marks]

# 1997 Paper 1 Question 11

**Operating Systems**

Why does the widespread use of graphical user interfaces (GUIs) make explicit the need for the underlying operating system to support concurrent processes and threads? [2 marks]

Outline the data structures that might be held by an operating system to support the entities that are scheduled (processes or threads). [8 marks]

Describe *one* scheduling algorithm and explain how it would be implemented, based on the data structures you have described above. [7 marks]

What are the implications for scheduling of the need to support new media types such as voice and video? [3 marks]

# 1997 Paper 1 Question 4

**Operating Systems**

Describe the various functions involved in interrupt handling. Indicate the hardware and software that might be involved in their implementation. [7 marks]

Discuss the interaction of interrupt-driven software and process scheduling in an operating system. [3 marks]

# 1997 Paper 2 Question 10

**System Architecture**

A hard disc and a floppy disc have many common features. Describe the operations which they support in common, as viewed by the filing system software which organises them into files and directories. How are they different? [10 marks]

Describe how a keyboard (or similar device) on a PC or workstation might typically be connected to the central microprocessor and the typical sequence of events when someone presses a key. [10 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]

# 1997 Paper 10 Question 12

**Unix Case Study**

Explain how the buffer-cache is used by the Unix file system. [10 marks]

Indicate why this approach might cause problems for applications which are concerned with the management of persistent data. [1 mark]

Give examples of when a process needs to *sleep* while accessing the buffer-cache and in each case indicate the event that causes the corresponding *wakeup* to be executed. [6 marks]

Explain the interaction of wakeup and scheduling. [3 marks]

# 1998 Paper 1 Question 11

**Operating Systems**

As well as storage and retrieval of data, the functions of a filing system include:

- naming and name resolution
- access control
- existence control
- concurrency control

For *each* of the above

(*a*) Briefly define the function. [1 mark each]

(*b*) Discuss how, and in which filing system component, the function may be provided. Use examples from one or more real operating systems to illustrate your answer. [4 marks each]

# 1998 Paper 1 Question 12

**Operating Systems**

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

List *five* techniques which the operating system can use to enhance the efficiency of demand paging. [5 marks]

Suppose the page table for the currently executing process is as follows. All values are decimal, and indexes are numbered from zero. Addresses are memory byte addresses. The page offset is 10 bits.

| Virtual Page # | Valid bit | Reference bit | Modify bit | Page Frame # |
|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 4 |
| 1 | 1 | 1 | 1 | 7 |
| 2 | 0 | 0 | 0 | – |
| 3 | 1 | 0 | 0 | 2 |
| 4 | 0 | 0 | 0 | – |
| 5 | 1 | 0 | 1 | 0 |

Describe exactly how, in general, a virtual address generated by the CPU is translated into a physical address, with the aid of a diagram. [4 marks]

To what physical address (if any) would the following virtual addresses correspond?

($a$) 1052

($b$) 2221

($c$) 5499

[6 marks]

# 1998 Paper 1 Question 4

**Operating Systems**

Typical memory architectures combine both paging and segmentation for the management of virtual memory. Contrast the two approaches with regard to their support for dynamic memory allocation, efficiency of memory use, support for code/data sharing, and data protection. [10 marks]

# 1999 Paper 1 Question 11

## Operating Systems

Most operating systems provide each process with its own *address space* by providing a level of indirection between virtual and physical addresses.

Give *three* benefits of this approach. [6 marks]

Are there any drawbacks? Justify your answer. [2 marks]

A processor may support a *paged* or a *segmented* virtual address space.

(*a*) Sketch the format of a virtual address in each of these cases, and explain using a diagram how this address is translated to a physical one. [8 marks]

(*b*) In which case is physical memory allocation easier? Justify your answer. [2 marks]

(*c*) Give *two* benefits of the segmented approach. [2 marks]

# 1999 Paper 1 Question 12

## Operating Systems

File systems comprise a *directory service* and a *storage service*.

What are the two main functions of the directory service? [2 marks]

What is a directory *hierarchy*? Explain your answer with the aid of a diagram. [2 marks]

What information is held in file *meta-data*? [4 marks]

What is a *hard link*? Does file system support for hard links place any restrictions on the location of file meta-data? [2 marks]

What is a *soft* (or *symbolic*) link? Does file system support for soft links place any restrictions on the location of file meta-data? [2 marks]

Describe with the aid of a diagram a Unix *inode*. You should explain the motivation behind *indirect blocks*, and how they are used when accessing a file. [8 marks]

# 1999 Paper 1 Question 4

**Operating Systems**

An operating system uses a single queue round-robin scheduling algorithm for all processes. You are told that a *quantum* of three time units is used.

What can you infer about the scheduling algorithm? [1 mark]

Why is this sort of algorithm suitable for a multi-user operating system? [1 mark]

The following processes are to be scheduled by the operating system.

| Process | Creation Time | Required Computing Time |
|---|---|---|
| $P_1$ | 0 | 9 |
| $P_2$ | 1 | 4 |
| $P_3$ | 7 | 2 |

None of the processes ever blocks. New processes are added to the tail of the queue and do not disrupt the currently running process. Assuming context switches are instantaneous, determine the *response time* for each process. [6 marks]

Give one advantage and one disadvantage of using a small quantum. [2 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]

# 2000 Paper 1 Question 11

**Operating Systems**

Describe with the aid of a diagram the life-cycle of a process. You should describe each of the states that it can be in, and the reasons why it moves between these states. [4 marks]

What information does the operating system keep in the process control block? [4 marks]

What information do the shortest job first (SJF) and shortest remaining time first (SRTF) algorithms require about each job or process? How can this information be obtained? [2 marks]

Give *one* advantage and *one* disadvantage of non-preemptive scheduling. [2 marks]

What steps does the operating system take when an interrupt occurs? Consider both the programmed I/O and DMA cases, and the interaction with the CPU scheduler. [4 marks]

What problems could occur if a system experienced a very high interrupt load? What if the device(s) in question were DMA-capable? [4 marks]

# 2000 Paper 1 Question 12

**Operating Systems**

In the context of memory management, under which circumstances do *external* and *internal* fragmentation occur? How can each be handled? [4 marks]

What is the purpose of a page table? What sort of information might it contain? How does it interact with a TLB? [3 marks]

Describe with the aid of a diagram a two-level page table. Explain the motivation behind the structure and how it operates. [5 marks]

What pieces of information make up the *meta-data* of a file? [4 marks]

Describe the basic access control scheme used in the Unix filing system. How does Unix support more advanced access control policies? [4 marks]

# 2000 Paper 1 Question 3

**Operating Systems**

Operating systems need to be able to prevent applications from crashing or locking up the system, or from interfering with other applications. Which *three* kinds of hardware support do we require to accomplish this? Justify your answer. [6 marks]

How do applications request that the operating system performs tasks on their behalf? [2 marks]

What could we do if we did not have the requisite hardware support? [2 marks]

# 2001 Paper 1 Question 11

**Operating Systems**

(*a*) Describe how the CPU is allocated to processes if static priority scheduling is used. Be sure to consider the various possibilities available in the case of a tie. [4 marks]

(*b*) "All scheduling algorithms are essentially priority scheduling algorithms."

Discuss this statement with reference to the first-come first-served (FCFS), shortest job first (SJF), shortest remaining time first (SRTF) and round-robin (RR) scheduling algorithms. [4 marks]

(*c*) What is the major problem with static priority scheduling and how may it be addressed? [4 marks]

(*d*) Why do many CPU scheduling algorithms try to favour I/O intensive jobs? [2 marks]

(*e*) Describe how this is achieved in the (*i*) UNIX and (*ii*) Windows NT operating systems. [3 marks in each case]

# 2001 Paper 1 Question 12

**Operating Systems**

(*a*) From the point of view of the device driver, data may be read from an I/O device using *polling*, *interrupt-driven programmed I/O*, or *direct memory access* (DMA). Briefly explain each of these terms, and in each case outline using pseudo-code (or a flow chart) the flow of control in the device driver when reading data from the device. [14 marks]

(*b*) From the point of view of the application programmer, data may be read from a device in a *blocking*, *non-blocking* or *asynchronous* fashion. Using a keyboard as an example device, describe the expected behaviour in each case. [6 marks]

# 2001 Paper 1 Question 4

**Operating Systems**

For *each* of the following, indicate whether the statement is true or false, and explain why this is the case (no marks will be awarded for an answer with no explanation).

(*a*) Round-robin scheduling can suffer from the so-called "convoy effect".

(*b*) System calls are an optional extra in modern operating systems like Windows 2000.

(*c*) A paged virtual memory is smaller than a segmented one.

(*d*) In UNIX, hard-links cannot span mount points.

(*e*) Direct memory access (DMA) makes devices go faster.

[2 marks each]

# 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]

.

# 2002 Paper 1 Question 11

**Operating Systems**

(*a*) Explain briefly the memory-management scheme of *paging*. [4 marks]

(*b*) Give *two* disadvantages of paging. [2 marks]

(*c*) A translation look-aside buffer (TLB) is sometimes used to optimise paging systems. Explain carefully how a TLB can be used in this way, and how it can optimise a paging system. [3 marks]

(*d*) The fictional Letni 2P chip uses (single-level) paging and has a memory access time of 8 nanoseconds and a TLB search time of 2 nanoseconds. What hit ratio (the probability that an item is in the TLB) must be achieved if we require an average (paged) memory access time of 12 nanoseconds? [4 marks]

(*e*) The management of the Letni Corporation wish you to design and evaluate a *multi-level* paging system for their new 64-bit processor, the 3P, which has 4K-sized pages.

(*i*) Give details of your proposed multi-level paging system. [5 marks]

(*ii*) State, and justify briefly, whether you think this proposal is realistic. [2 marks]

# 2002 Paper 1 Question 12

**Operating Systems**

Two important facilities provided by a conventional operating system (OS) are

- protection between different parts of the system,
- convenient programming interfaces by abstraction from the basic facilities exposed by the hardware.

Illustrate how these facilities are provided for the following four resources. In each case you should describe how protection is enforced and outline the interface that the hardware provides to the OS and an interface that the OS may provide to an application:

(*a*) CPU processing time; [5 marks]

(*b*) access to a device, such as a serial or parallel data port; [5 marks]

(*c*) storage of data and code in memory; [5 marks]

(*d*) storage of files on disk. [5 marks]

# 2002 Paper 1 Question 4

**Operating Systems**

(*a*) What is an *interrupt*? [1 mark]

(*b*) A hardware device wishes to transfer information to the main memory of the computer for access by an application. The following three mechanisms are available:

(*i*) polled mode operation;

(*ii*) programmed I/O;

(*iii*) direct memory access (DMA).

For each one, summarise its operation and suggest an advantage it holds over the others.

[3 marks each]

# 2002 Paper 10 Question 7

**Operating System Foundations**

(*a*) Explain briefly the memory-management scheme of *paging*. [4 marks]

(*b*) Give *two* disadvantages of paging. [2 marks]

(*c*) A translation look-aside buffer (TLB) is sometimes used to optimise paging systems. Explain carefully how a TLB can be used in this way, and how it can optimise a paging system. [3 marks]

(*d*) The fictional Letni 2P chip uses (single-level) paging and has a memory access time of 8 nanoseconds and a TLB search time of 2 nanoseconds. What hit ratio (the probability that an item is in the TLB) must be achieved if we require an average (paged) memory access time of 12 nanoseconds? [4 marks]

(*e*) The management of the Letni Corporation wish you to design and evaluate a *multi-level* paging system for their new 64-bit processor, the 3P, which has 4K-sized pages.

(*i*) Give details of your proposed multi-level paging system. [5 marks]

(*ii*) State, and justify briefly, whether you think this proposal is realistic. [2 marks]

# 2002 Paper 11 Question 5

## Operating System Foundations

(*a*) Write brief notes on the following:

(*i*) Interrupt-based I/O. [3 marks]

(*ii*) Direct Memory Access. [3 marks]

(*iii*) File access control in Unix. [3 marks]

(*iv*) Mounting file systems in Unix. [3 marks]

(*b*) You are one of the design team of the new Xinu file system. As part of this design you have proposed an inode structure similar to that in "traditional" Unix. (Blocks are assumed to be 4KB, and file pointers are 4 bytes.)

(*i*) Give details of your proposal for the inode structure. [6 marks]

(*ii*) Calculate the maximum file size, in blocks. [2 marks]

# 2003 Paper 1 Question 11

## Operating Systems

(*a*) Describe, with the aid of diagrams where appropriate, how Unix implements and manages:

(*i*) a hierarchical name space for files; [2 marks]

(*ii*) allocation of storage on disk; [2 marks]

(*iii*) file-system and file meta-data; [8 marks]

(*iv*) pipes. [4 marks]

(*b*) A system administrator decides to make a 'versioned' file-system in which there are a number of directories called `/root-dd-mm-yyyy`, each of which holds a copy of the file-system on day `dd`, month `mm` and year `yyyy`. The idea is that at any particular time only the most recent snapshot will be used as the 'real' filesystem root, but that all previous snapshots will be available by explicitly accessing the directory in question. In this way the system administrator hopes to allow resilience to mistaken edits or unintentional deletions by users, or to hardware problems such as a disk head crash.

To implement this, the system administrator arranges for a program to run every morning at 01:00 which recursively 'copies' the current snapshot to the new one. However to save disk space, hardlinks are used in place of actual copies. Once the 'copy' is complete, the new snapshot is used as the new root.

To what extent will this scheme provide the functionality the system administrator hopes for? What advantages and disadvantages does it have? [4 marks]

# 2003 Paper 1 Question 12

## Operating Systems

(*a*) Describe with the aid of a diagram how a simple computer executes a program in terms of the *fetch–execute cycle*, including the ways in which arithmetic instructions, memory accesses and control flow instructions are handled.

[12 marks]

(*b*) Explain how a program accesses I/O devices when:

(*i*) it is running in supervisor-mode; [5 marks]

(*ii*) it is running in user-mode. [3 marks]

# 2003 Paper 1 Question 4

**Operating Systems**

For each of the following, indicate whether the statement is true or false, and explain why this is the case (no marks will be awarded for an answer with no explanation).

(*a*) Preemptive schedulers require hardware support.

(*b*) The Unix shell supports redirection to the buffer cache.

(*c*) A context switch can be implemented by a flip-flop stored in the translation lookaside buffer (TLB).

(*d*) Non-blocking I/O is possible even when using a block device.

(*e*) Shortest job first (SJF) is an optimal scheduling algorithm.

[2 marks each]

# 2004 Paper 1 Question 11

## Operating Systems

(*a*) What was the key innovation of the *von Neumann Architecture*? [2 marks]

(*b*) Describe (with the aid of a diagram where appropriate) the representation in main memory of:

(*i*) an unsigned integer; [2 marks]

(*ii*) a signed integer; [2 marks]

(*iii*) a floating point number; [4 marks]

(*iv*) an instruction. [4 marks]

(*c*) Does an operating system need to know whether the contents of a particular register represent a signed or unsigned integer? Justify your answer. [2 marks]

(*d*) Describe what occurs during a *context switch*. [4 marks]

# 2004 Paper 1 Question 12

## Operating Systems

(*a*) *System calls* are part of most modern operating systems.

(*i*) What is the purpose of a system call? [2 marks]

(*ii*) What mechanism is typically used to implement system calls? [2 marks]

(*b*) Process scheduling can be *preemptive* or *non-preemptive*. Compare and contrast these approaches, commenting on issues of simplicity, fairness, performance and required hardware support. [8 marks]

(*c*) Briefly compare and contrast the notion of *process* in the Windows XP and Unix operating systems. Describe the scheduling algorithms used in each case. [8 marks]

# 2004 Paper 1 Question 4

## Operating Systems

($a$) Modern computers store data in a variety of "memories", each with differing size and access speeds. Briefly describe each of the following:

($i$) cache memory; [2 marks]

($ii$) main memory; [2 marks]

($iii$) registers. [2 marks]

($b$) Give an example situation in which operating systems effectively consider disk storage to be a fourth type of "memory". [2 marks]

($c$) A researcher proposes using fast non-volatile memory for *all* data, rather than using the four separate kinds of "memory" mentioned above. Comment on the pros and cons of this approach. [2 marks]

# 2004 Paper 10 Question 6

**Operating System Foundations**

(*a*) Describe a scheduling algorithm with the following properties:

- favours I/O-intensive processes
- responds dynamically when processes change their behaviour: e.g. enter a compute-bound or I/O-intensive phase
- has acceptable context switching overhead
- avoids indefinite overlook (starvation) of a process

[7 marks]

(*b*) In order to carry out its functions, a filing system holds metadata on each stored object.

(*i*) What is this metadata likely to comprise? [6 marks]

(*ii*) Describe the directory service functions of a filing system, including how the metadata is used. [7 marks]

# 2004 Paper 11 Question 6

**Operating System Foundations**

Two operating systems OS-A and OS-B offer only synchronous system calls, for example, for I/O. In addition, OS-A supports only one process per user-level address-space whereas OS-B supports multi-threaded applications.

($a$) ($i$) Explain how an application-level runtime system or library running on OS-A can provide the user threads needed by concurrent programs. [8 marks]

($ii$) Discuss any disadvantages of supporting a concurrent programming language in this way. [3 marks]

($iii$) Are there any advantages of having only user threads? [1 mark]

($b$) ($i$) Explain the *differences* from the runtime described for OS-A of a runtime for OS-B which maps user threads to kernel threads. [5 marks]

($ii$) Are the disadvantages you discussed in part ($a$)($ii$) overcome? Explain. [2 marks]

($iii$) Have any problems been introduced by the use of kernel threads? [1 mark]

# 2005 Paper 1 Question 11

**Operating Systems**

(*a*) Describe with the aid of a diagram the on-disk layout of a UNIX V7 filesystem. Include in your description the role of the *superblock*, and the way in which free inodes and data blocks are managed. [6 marks]

(*b*) Describe with the aid of a diagram a UNIX V7 *inode*. [6 marks]

(*c*) Estimate the largest file size supported by a UNIX V7 filesystem. [2 marks]

(*d*) Suggest *one* reliability enhancement and *two* performance enhancements which could be made to the UNIX V7 filesystem. [2 marks each]

# 2005 Paper 1 Question 12

**Operating Systems**

(*a*) Describe with the aid of a diagram the structure of the Windows XP operating system. Sketch the functions of each component, and clearly indicate which parts execute in kernel mode and which in user mode. [6 marks]

(*b*) Compare and contrast the *object namespace* in Windows XP with the *directory namespace* of UNIX. [4 marks]

(*c*) Compare and contrast *blocking*, *non-blocking* and *asynchronous* I/O. [6 marks]

(*d*) Give *four* techniques which can improve I/O performance. [1 mark each]

# 2005 Paper 1 Question 4

## Operating Systems

(*a*) What is the *address binding* problem? [1 mark]

(*b*) The address binding problem can be solved at compile time, load time or run time. For *each* case, explain what form the solution takes, and give *one* advantage and *one* disadvantage. [3 marks each]

# 2005 Paper 11 Question 6

## Operating System Foundations

($a$) A system has paging hardware but no segmentation hardware. Discuss the likely structure of a process page table. What information would you expect to be held in a page table entry? [5 marks]

($b$) Describe the operation of the following hardware support options for paging:

($i$) A pair of processor registers.
PTBR holds the address of the base of the page-table of the current process.
PTLR holds the size of the page table in bytes. [5 marks]

($ii$) A TLB (Translation Lookaside Buffer). [10 marks]

# 2006 Paper 1 Question 2

## Operating Systems

For *each* of the following, indicate whether the statement is true or false, and explain why this is the case (no marks will be awarded for an answer with no explanation).

($a$) The Windows XP Executive is mostly implemented in user mode.

($b$) Floating-point hardware can be used to invert the access matrix.

($c$) Polled-mode I/O is sometimes preferable to interrupt-driven I/O.

($d$) Microkernel operating systems are faster than monolithic systems.

($e$) Windows XP is architecturally more secure than Unix.

[2 marks each]

# 2006 Paper 1 Question 7

**Operating Systems**

(*a*) FIFO, LRU, and CLOCK are three page replacement algorithms.

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

(*ii*) 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]

(*iii*) 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]

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

(*c*) Which buffer-cache replacement strategy would you choose to use within an operating system? Justify your answer. [4 marks]

# 2006 Paper 1 Question 8

## Operating Systems

(*a*) Devices are ultimately connected to the CPU via a *bus*.

(*i*) What are the main components of a bus? [3 marks]

(*ii*) Describe how the CPU uses a bus to communicate with a device. [3 marks]

(*iii*) How does the situation become more difficult when we have DMA-capable devices? [2 marks]

(*iv*) Why does a typical computer have more than one bus? [2 marks]

(*b*) A programmer at MegaCorp is given the task of optimising a program for which no source code exists; all that is available is an executable file.

(*i*) How could the programmer modify the operating system to work out which parts of the program are executed frequently, and thus might be candidates for optimisation? [4 marks]

(*ii*) The programmer determines that the slowest parts of the program involve loops which perform repeated integer multiplications, and where the multiplicand is always either 8 or 15. How could the program be modified to use faster ALU operations instead? [6 marks]

# 2006 Paper 10 Question 7

## Operating System Foundations

(*a*) Describe the steps involved in resolving any component of a file pathname. [6 marks]

(*b*) (*i*) Describe the steps involved in

- creating a file;
- deleting a file. [8 marks]

(*ii*) Discuss the possible effects of a crash, causing loss of main memory, at various points during file creation and deletion. [6 marks]

# 2007 Paper 1 Question 2

**Operating Systems**

(*a*) In relation to scheduling of processes, describe the concept of a working set and briefly outline how it can be used within an operating system. [3 marks]

(*b*) Briefly explain why context switching between processes is inherently more costly than switching between threads of a process. [3 marks]

(*c*) Give *two* reasons why operating system designers often choose to make code in the kernel non-preemptive. [2 marks]

(*d*) Why would it be bad for a Unix file owned by `root` with the `setuid` bit set also to have read, write and execute access permissions granted to all users? [2 marks]

# 2007 Paper 1 Question 7

## Operating Systems

(*a*) Describe carefully how a 32-bit virtual address could be translated to a physical address during the execution of a memory reference instruction on a typical modern CPU that supports paged virtual memory. You should assume that the page size is 4096 bytes and that the system uses two-level paging with page tables at both levels holding 1024 entries. [5 marks]

(*b*) List the protection bits that you would expect to find in a page-table entry and briefly explain how they are used. [5 marks]

(*c*) Outline the main differences between paging and segmentation, and show how a segmentation scheme can be implemented with reasonable efficiency in a system that supports paging. [4 marks]

(*d*) Outline how you would implement, on a machine with 64-bit virtual addresses, the MULTICS-like view of files in which open files are mapped onto positions in virtual memory. [6 marks]

# 2007 Paper 1 Question 8

**Operating Systems**

(*a*) Most modern processors have a status register which include bits that specify whether it is running in supervisor or user state, and whether interrupts are enabled. Explain why this information is useful and what effect they have on instruction execution. [4 marks]

(*b*) Suggest *three* situations that cause the settings of these status bits to change. [3 marks]

(*c*) Discuss whether it is ever useful to run

(*i*) in supervisor state with interrupts enabled;

(*ii*) in user state with interrupts disabled. [4 marks]

(*d*) Carefully describe the structure of a Unix *inode*. Assuming that the block size is 4096 bytes in a Unix file system and that indirection blocks can hold 512 entries, calculate the size in bytes of the largest file that does not need to use triple indirection. You may give your answer as a formula. [5 marks]

(*e*) State *four* reasons why the Windows NTFS file system is superior to the FAT32 file system. [4 marks]

# 2007 Paper 10 Question 12

**Operating System Foundations**

Operating systems are integrated closely with the hardware on which they run.

($a$) Distinguish the hardware and software involved in interrupt-driven I/O. [4 marks]

($b$) Describe *three* uses of the interrupt mechanism in addition to device I/O. [3 marks]

($c$) Give examples of how the privilege state bit in a CPU's status register is used. [2 marks]

($d$) ($i$) What could go wrong in a system that does not make use of a timing device in process scheduling? [1 mark]

($ii$) What class of scheduling algorithms would be impossible to implement without a timing device? [2 marks]

($e$) What is the main advantage of using half the virtual address space of a process for the operating system and half for applications? [1 mark]

($f$) Explain memory-mapped I/O. [2 marks]

($g$) In a paging system, would you expect every page of the operating system address space to be:

($i$) mappable in the Translation Lookaside Buffer (TLB)? [2 marks]

($ii$) capable of being cached? [1 mark]

Explain your answers.

($h$) How do DMA (Direct Memory Access) devices operate? [2 marks]

# 2007 Paper 11 Question 12

**Operating System Foundations**

(*a*) Each object named in a filing system has associated with it a fixed-length metadata record (file control block). Describe *three* ways in which the disk blocks allocated to a file have been recorded, as part of its metadata, in various filing systems. State the good and bad properties of these approaches. [9 marks]

(*b*) (*i*) What is meant by a *hard link*? [2 marks]

(*ii*) Can a file or directory have no hard links? Explain. [1 mark]

(*c*) Filing systems make use of an in-memory cache of disk blocks. This is a shared data structure, accessed by device drivers and processes carrying out application-level requests for I/O.

(*i*) What are the advantages and disadvantages of not writing synchronously to disk? [2 marks]

(*ii*) Discuss why both mutual exclusion and condition synchronisation are needed for this data structure. [4 marks]

(*d*) One filing system names the blocks in its disk-block cache as filing-system-ID, block-number. In another, the blocks are offsets within a file. What are the implications of these approaches? [2 marks]

# 2008 Paper 1 Question 2

## Operating Systems

Let $N$ be the 16-bit value $1001\,0101\,0000\,0000_2$.

($a$) What is the value of $N$ when interpreted as:

($i$) An unsigned integer? [1 mark]

($ii$) A sign-and-magnitude format integer? [1 mark]

($iii$) A 2's complement integer? [2 marks]

($iv$) A floating-point number with a 5-bit bias-15 exponent and a normalised mantissa? [State any assumptions you make.] [4 marks]

($b$) Imagine $N$ has been loaded into the 16-bit register `r1`. Explain what the values of the $C$ (carry) and $V$ (overflow) flags would be after the CPU executes the instruction `add r0` $\leftarrow$ `r1, r1`. [2 marks]

# 2008 Paper 1 Question 7

**Operating Systems**

(*a*) What is *file metadata*? [2 marks]

(*b*) Explain with the aid of a diagram how file metadata is managed in:

(*i*) the Unix file-system; [4 marks]

(*ii*) the FAT32 file-system; [4 marks]

(*iii*) the NTFS file-system. [4 marks]

(*c*) A researcher suggests using non-volatile flash memory to store the NTFS log file. He believes this will *improve performance*, *reduce power consumption* and make the system *more resilient to failure*. Is he right? Briefly justify your answer in each case. [2 marks each]

# 2008 Paper 1 Question 8

## Operating Systems

(*a*) In general we can consider a process to move between five *process states* during its existence. Discuss, with the aid of a diagram, the circumstances in which a process will enter or leave each of these states. [6 marks]

(*b*) Both Unix and Windows NT use *dynamic priority scheduling*. Compare and contrast their scheduling algorithms, with a particular focus on how dynamic priorities are managed. [8 marks]

(*c*) Which data structures would you use if implementing a dynamic priority scheduling algorithm? Justify your answer. [2 marks]

(*d*) Some industry predictions suggest that in less than five years we shall have chips with hundreds or even thousands of CPUs on them. What kinds of scheduling algorithms do you think will be appropriate for such systems? What problems do you foresee? [4 marks]

# 2009 Paper 2 Question 3

## Operating Systems

(*a*) Operating systems typically provide each process with a *virtual address space*.

(*i*) Give *three* advantages of this. [3 marks]

(*ii*) In which circumstances does *external fragmentation* occur? How can it be managed? [2 marks]

(*iii*) In which circumstances does *internal fragmentation* occur? [1 mark]

(*iv*) Design a multi-level page table for a computer with a 48-bit virtual address space, 48-bit physical address space, and a 4K page size. You should explain its operation, and justify your design decisions. [6 marks]

(*b*) In the context of the UNIX operating system:

(*i*) What is a *pipe*? What is it used for? How does it operate? Use a diagram to illustrate your answer. [4 marks]

(*ii*) What is the *shell*? Describe its operation in pseudo-code, giving special emphasis to any system calls invoked. [4 marks]

# 2009 Paper 2 Question 4

## Operating Systems

(*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*) Buses are used to connect devices to the processor.

(*i*) Describe with the aid of a diagram the operation of a *synchronous* bus. [4 marks]

(*ii*) In what ways does an *asynchronous* bus differ? [2 marks]

# 2010 Paper 2 Question 3

## Operating Systems

(*a*) Shortest Job First (SJF), Shortest Remaining Time First (SRTF) and Round Robin (RR) are CPU scheduling algorithms. Briefly describe each.
[1 mark each]

(*b*) Consider the following set of processes.

| Process | Creation Time | Required Computing Time |
|---|---|---|
| $P_1$ | 0 | 25 |
| $P_2$ | 5 | 15 |
| $P_3$ | 10 | 5 |
| $P_4$ | 15 | 5 |

(*i*) Draw a diagram showing how SJF would schedule the processes. What is the average waiting time? [3 marks]

(*ii*) Draw a diagram showing how SRTF would schedule the processes. What is the average waiting time in this case? [3 marks]

(*iii*) Assuming a quantum of 10 time units, draw a diagram showing how RR would schedule the processes. What is the average waiting time in this case? [3 marks]

(*iv*) A student suggests that RR could be improved by reducing the quantum to 1 time unit. What are the advantages and disadvantages of this proposal? [2 marks]

(*c*) Compare and contrast the Unix *buffer cache* and the Windows NT *cache manager*. [6 marks]

# 2010 Paper 2 Question 4

## Operating Systems

(*a*) The virtual address space of a Unix V7 process contains a *text segment*, a *data segment* and a *stack segment*.

(*i*) What is contained in the text segment? How does this change as the process executes? [2 marks]

(*ii*) What is contained in the data segment? How does this change as the process executes? [2 marks]

(*iii*) What is contained in the stack segment? How does this change as the process executes? [2 marks]

(*b*) The Unix kernel is also present in the virtual address space of every process. Describe how the operating system can ensure that this memory region is protected from access by an executing process. Under what circumstances can a process gain access to this region of virtual memory? [2 marks]

(*c*) Compare and contrast *blocking*, *non-blocking* and *asynchronous* I/O. [2 marks each]

(*d*) You are asked to write a device driver for a hard-disk drive.

(*i*) Under what circumstances will you issue requests to the drive? [2 marks]

(*ii*) What steps will you need to take when an interrupt occurs? [2 marks]

(*iii*) Given that the hard-disk drive is not really a random access device, what steps could you take to improve performance? [2 marks]