

Monash University

Semester Two 2007 Examination Period

Faculty of Information Technology

EXAM CODES: CSE/FIT

TITLE OF PAPER: FIT2022 Computer Systems II

EXAM DURATION: 3 hours writing time

READING TIME: 10 minutes

THIS PAPER IS FOR STUDENTS STUDYING AT: Clayton, Malaysia

During an exam, you must not have in your possession, a book, notes, paper, calculator, pencil case, mobile phone or other material/item which has not been authorised for the exam or specifically permitted as noted below. Any material or item on your desk, chair or person will be deemed to be in your possession. You are reminded that possession of unauthorised materials in an exam is a discipline offence under Monash Statute 4.1.

No examination papers are to be removed from the room.

AUTHORISED MATERIALS

CALCULATORS: NO

OPEN BOOK: NO

SPECIFICALLY PERMITTED ITEMS: NO

Candidates must complete this section if required to write answers within this paper

STUDENT ID

DESK NUMBER

SECTION A: Multiple Choice (20 marks total)

Each question in this section is worth *2 marks*. Circle the one alternative (a,b,c,d,e) that you think is the **best** answer to the question. If you make a mistake, draw a single stroke “/” through your incorrect answer, then circle your revised answer.

- 1) Operating systems are all about:
 - a) managing memory
 - b) managing processes
 - c) managing people
 - 2** d) managing computer resources
 - 1** e) all of the above

- 2) The kernel of the operating system:
 - a) is written in Python
 - 1** b) manages all interrupt activity in the system
 - 2** c) schedules I/O
 - d) keeps the memory partitioned
 - e) recovers the system from deadlock

- 3) A process scheduler:
 - 1** a) is a kernel process
 - b) is a user process
 - c) is an application process
 - 2** d) is an operating system process
 - e) is not a process at all, since it schedules them

- 4) Process synchronization involves:
 - a) deadlocks
 - b) graphical user interfaces
 - 1** c) non-atomic operations
 - d) all of the above
 - 2** e) none of the above

- 5) An interrupt
 - a) is a faster way of transferring large blocks of data
 - 2** b) improves the scheduling of waiting processes
 - c) is caused by an atomic instruction
 - d) is generated whenever deadlock occurs
 - e) none of the above

- 6) The limit register in memory protection systems
- defines the starting address of a process image
 - 2** b) controls the size of a process image
 - allows access to any location in memory
 - defines the address of a page table
 - causes an interrupt on page faults
- 7) Dispatch latency is
- the time a process waits before being executed by the dispatcher
 - the time it takes for the dispatcher to start a waiting process
 - the time a process executes before being pre-empted by the dispatcher
 - the time a process waits before being terminated by the dispatcher
 - 2** e) the time it takes for the dispatcher to switch between processes
- 8) Indexed allocation in file system organization is where
- a) each file is represented by an i-node
 - b) the file sectors are linked in a list
 - c) pointers to the file sectors are stored in an indexed hash table
 - 2** d) each file has its own block of pointers to the sectors of the file
 - e) none of the above
- 9) Starvation in disk scheduling may occur in the algorithms:
- FCFS (First Come First Served) and SCAN
 - FCFS and LOOK
 - SSTF (Shortest Seek Time First) and C-SCAN
 - SSTF and FCFS
 - 2** e) none of the above
- 10) Suppose $D_1 = \{ \langle O_1, rw \rangle, \langle O_2, rx \rangle \}$, $D_2 = \{ \langle O_1, rp \rangle, \langle O_2, w \rangle \}$ are two protection domains, where r=read right, w=write right, x=execute right and p=print right. Then
- In D_1 , O_1 can be executed
 - In D_1 , O_1 can be printed
 - 2** c) In D_2 , O_1 can be printed
 - In D_2 , O_2 can be read
 - In D_2 , O_2 can be executed

SECTION B: Answer any FOUR questions in this section (80 marks total)

1)

- a) What is the difference between multiprogramming and multiprocessing? [4 marks]

Answer: Multiprogramming allows multiple processes to run concurrently;
Multiprocessing allows multiple processors to run concurrently.

2 marks each for these statements or (succint) variants thereof. Lose 1 mark if they omit the word (or notion of) concurrency, 1 mark if distinction between processes and processors not made adequately.

- b) What are the advantages and disadvantages of multiprocessing? [4 marks]

Answer: Advantages:

- more efficient use of resources
- greater programmer/user productivity
- increased throughput of programs

Disadvantages:

- greater complexity
- increased overheads
- security concerns

1 mark for each valid point up to a total of 4

- c) What are the memory management implications of multiprocessing? In your answer, explain particularly how shared memory affects memory management in the context of multiprocessing. [4 marks]

Answer:

- multiple processes, so multiple address spaces
- address spaces need to be protected against each other
- shared memory requires lifting of protection mechanisms for shared areas
- usually require hardware assistance to maintain protect and separation

(1 mark each)

- d) Some computers have three levels of privileged instructions: user, supervisor, and kernel; as opposed to the usual two (user and supervisor). Explain why the kernel mode might be useful. [4 marks]

Answer: There are a number of areas where separation between kernel activities and more general operating system activities might be made: for example, interrupt device handling and (kernel) versus device buffering (supervisor); process switching (kernel) versus process scheduling (supervisor); etc.. (3 marks for identifying any separation issues, 1 for any additional examples)

- e) Which of the following instructions should be executable in user mode? (circle all correct answers)
- i) set value of timer
 - 1** ii) read the clock
 - 1** iii) turn off interrupts
 - 1** iv) subroutine call
 - 1** v) operating system (or supervisor) call
 - 1** vi) switch from user mode to kernel mode

Answer: (plus 1 if all 3 correct)

[4 marks]

2) **The Supermarket Checkout Problem:** A supermarket has a row of checkout stations, where customers can present their purchased goods for payment. Customers arrive at random times at the checkouts. There is a queue at each checkout (possibly empty), and when each customer arrives at the head of the queue, they deposit goods on the counter for checkout by the operator. The operator in turn picks up these goods, scans them, and packs them into bags for later collection by the customer. When all goods have been scanned, the customer pays the total amount of money due, and collects their packed goods.

You are to model this activity by writing two pieces of code, one for describing each customer process, and one for describing each checkout operator process. Pay particular attention to the synchronization of the two processes. Assume each customer process is initiated by arrival at the checkout queue, and terminates when goods are collected. There may be more than one of these processes at any one time. The checkout operator process is assumed to run indefinitely, and you need model only one such process. You may use any programming language you choose to write the code. (A bonus of 1 mark applies if you use Python!) Pseudo-code is acceptable. [20 marks]

Answer:

- a) **Identify processes** (2 marks)
- b) **Identify variables** (4 marks)
- c) **Identify events** (2 marks)
- d) **write checkout process** (4 marks)
- e) **write customer process** (4 marks)
- f) **add critical sections** (4 marks)

3)

- a) In Laboratory session 3, you saw how to implement a rudimentary file system. Identify what issues have to be addressed if the file system design allows for Unix-style *links* to be used in identifying files. Give your answer in point form. [8 marks]

Answer:

- what is a link? multiple directory references to the one i-node (2 marks)
- deleting a file just removes a link. Only when last link removed can file be deleted, therefore need reference count (2 marks)
- need mechanism for adding and removing links (2 marks)
- files manipulated via i-node references, since names are non-unique (2 marks)
- (other comments possible)

b) A disk drive has a total of 100 tracks, with a track-to-track seek time of 0.1mS, and a linear seek time proportional to the number of tracks traversed (so seeking from track 0 to track 100 takes 10mS). The head may be returned from track 100 to track 0 with a “head home” operation that takes only 0.2mS.

Given a set of 5 independently generated disk block addresses [85,60,72,36,91], arrange them in the order that gives the lowest seek time under

- i) shortest seek time first scheduling;
- ii) cyclic scanning, where the head starts at track 0 (the home track) and always moves upward, except when returning from the last track to the home track;

assuming that the head is currently positioned at track 67. You may ignore rotational latency, and assume that seek time is proportional to track distance. Which algorithm gives the better performance?

If the reference to track 85 is changed to track 83, what happens to the seek times and the relative performance?

[12 marks]

Answer:

| Start | Higher | Lower | DiffUp | DiffDown | Choose | Time |
|-------|--------|-------|--------|----------|--------------|------------|
| 67 | 72 | 60 | +5 | -7 | 72 | 0.5 |
| 72 | 85 | 60 | +13 | -12 | 60 | 1.2 |
| 60 | 85 | 36 | +25 | -24 | 36 | 2.4 |
| 36 | 85 | - | +45 | - | 85 | 4.5 |
| 85 | 91 | - | +6 | - | 91 | 0.6 |
| 91 | | | | | total | 9.6 |

Sequence is 72,60,36,85,91; total time 9.6mS (2 marks for sequence, 2 for time)

| Start | Next | Diff | Time |
|-----------|------|--------------|------------|
| 67 | 72 | +5 | 0.5 |
| 72 | 85 | +13 | 1.3 |
| 85 | 91 | +6 | 0.6 |
| Head Home | | | |
| 91 | 0 | - | 0.2 |
| 0 | 36 | +36 | 3.6 |
| 36 | 60 | +24 | 2.4 |
| | | total | 8.6 |

Sequence is 72,85,91,(H),36,60; total time 8.6mS (2 marks for sequence, 2 for time)

C-SCAN is the better algorithm (2 marks)

If ref 85 is changed to 83, the sequence becomes 72,83,91,60,36 time 7.9mS for SSTF, unchanged for C-SCAN, hence the relative performance is reversed! (2 marks)

4)

- a) Write a short program fragment (in any language) that defines the semantics (meaning) of the Test and Set instruction. [5 marks]

Answer:

```
def TestAndSet(v):  
    ret = v.value  
    v.value = 1  
    ret
```

(2 marks for returning **old** value, 1 mark for setting variable, 2 marks for identifying that parameter must be pass-by-reference)

- b) Why can't you use a test and set instruction in place of a binary semaphore? Explain what additional semantics are required. [5 marks]

Answer: A binary semaphore requires either a busy wait (deprecated) or a blocking wait, semantics not provided directly in the Test and Set. (The advantage of a binary semaphore is that it does not require an arbitrary length queue of processes waiting on the semaphore.)

(5 marks for blocking wait or explanation of why busy wait is deprecated, 3 marks for just busy wait. 1 bonus mark for explanation of binary semaphore.)

- c) Write a short program fragment (in any language) that defines the semantics of the blocking `wait(semaphore)` operation, and calls the test and set operation defined in part a) above. You may assume a primitive instruction "block". [5 marks]

Answer:

```
def wait(semaphore):  
    if TestAndSet(semaphore) == 0:  
        semaphore.process = this_process  
        block
```

(1 mark for the semaphore parameter, 2 marks for the `if` with `TestAndSet` conditional, 1 mark for identifying which process is blocked, 1 mark for the `block`.)

- d) What is the difference between deadlock prevention, deadlock avoidance, and deadlock detection? [5 marks]

Answer: prevention: Ensuring that at least one of the four conditions for deadlock never holds; (2 marks)

avoidance: Ensuring that processes never enter the "unsafe" regions of resource allocation; (2 marks)

detection: Allowing deadlock to occur, and then recovering from the deadlock. (1 mark)

5)

a) Consider a demand-paged computer system where the degree of multiprogramming is currently fixed at 4. The system was measured to determine utilization of the CPU and the paging disk. The results on three different occasions were

- i) CPU utilization 13%, disk utilization 97%
- ii) CPU utilization 87%, disk utilization 3%
- iii) CPU utilization 13%, disk utilization 3%

On each occasion, what was happening and why? Can the degree of multiprogramming be increased to improve the CPU utilization? Is the paging helping? [10 marks]

Answer:

- i) system is thrashing. No. No.
- ii) system is working well. Not really, CPU running close to maximum utilization. Yes.
- iii) system is largely idle. Yes. Not really.

1 mark for each correct point, and 1 for any explanations.

b) Explain the “working set model” of virtual memory. On a given system, the following measurements were made:

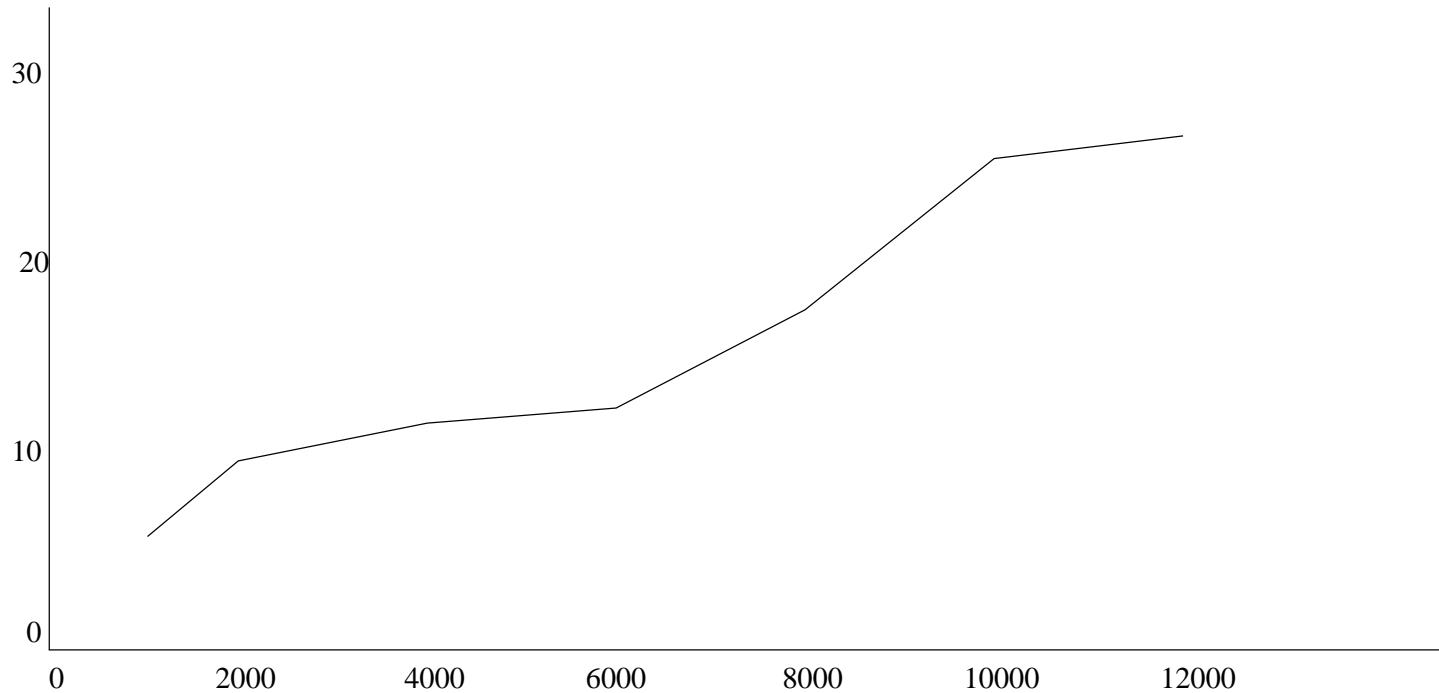
| Number of instructions | Working Set size (pages) |
|------------------------|--------------------------|
| 1000 | 6 |
| 2000 | 10 |
| 4000 | 12 |
| 6000 | 13 |
| 8000 | 18 |
| 10000 | 26 |
| 12000 | 27 |

Sketch the shape of this curve and explain what is happening as the number of instructions in the working set window increases

- i) from 2000 through to 6000
- ii) from 6000 to 10000
- iii) from 10000 to 12000

[10 marks]

Answer: The working set model identifies a set of pages that contains the most recent Δ references. It attempts to capture program locality by identifying those pages that make up the current set of pages in active use. As the range Δ increases, the working set increases, but on the assumption that it is less than linear. (2 marks)



- i) operating from within first level working set
- ii) filling second level working set
- iii) operating within second level working set

Note that from 0 to 2000 the first level working set is being filled
 2 marks for plotting curve (gift!), 2 marks for each of the above statements (1 bonus mark for a comment about 0 to 2000). If they have not got the three key statements above, but write something that indicates they might know what is going on, 1 mark (i.e., 5 marks max for Q)

- a) List the three characteristics that any solution to the critical section problem must address.

[6 marks]

Answer:

- i) mutual exclusion on critical section
- ii) bounded wait on any process wanting to enter critical section
- iii) all processes must make progress when entering a critical section

(2 marks each)

- b) Instead of a password consisting of a sequence of characters to be entered, it is proposed to use a sequence of arithmetic operators (+, −, ×, ÷). When the user wants to authenticate to a system, they type their username, and then the computer responds with a sequence of numbers. The user applies their operators to the digits, and then enters the result, which must match to authenticate.

For example, suppose your password is −, ×, +, +, and the computer tells you 23, 34, 9, 16, 12. You compute the expression $((((23 - 34) \times 9) + 16) + 12)$ and get −71, which you enter to authenticate yourself.

Discuss the advantages and disadvantages of such a scheme.

[14 marks]

Answer:

Advantages

- no obvious eavesdropping
- response is different each time
- (therefore) same 'password' can be used for many applications
- response can be sent as clear text

Disadvantages

- complex calculation for user - takes time and effort
- (therefore) not suitable for some users
- must balance size of generated numbers with complexity of algorithm
- algorithm must be non-trivial (e.g. +, +)
- some calculations may have to be excluded (e.g., integer division - what happens? generated numbers may have to be tailored)

(2 marks each correct answer, to max of 14)

END OF EXAMINATION QUESTIONS