**Student Name:** Tanaka Chitete

**Student ID:** 20169321

*Please do NOT change the order of Question/Answer. If you are unable to answer any question, do not delete the Question/Answer number.*

**QUESTION ONE (total: 20 marks): Deadlock**

**Q1 a) (Total: 6 marks).**

Q1 a) (i) (2 marks).

**Answer:**

One possible scenario would be when every process requests a resource at the same

time. In this situation, each process would own a resource but would be waiting on

another process to relinquish their own in order to acquire a second resource and

begin execution.

Q1 a) (ii) (4 marks).

**Answer:**

A simple rule should be that at most n/2 processes own an instance of a resource at any given time. In this way, hold-and-wait is prevented due to the fact that by only allowing n/2 processes to own a resource, we *guarantee* that all of the n/2 processes will be able to secure another resource as n resources divided by n/2 processes equals 2, meaning each process will be able to own the 2 resources required for execution.

**Q1 b) (Total: 4 marks).**

Q1 b) (i) (2 marks).

**Answer:**

Yes, as process A and process B may both acquire a subset of the resources. Since each process requires all four resources for execution, they will deadlock as there are not enough resources to satisfy both requests (i.e. 8) and neither process will relinquish their resources.

Q1 b) (ii) (2 marks).

**Answer:**

Yes, as process A and B may still both acquire a subset of the resources. Since each process requires all four resources for execution, they will deadlock as there are not enough resources to satisfy both requests (i.e. 8) and neither process will relinquish their resources.

**Q1 c) (4 marks).**

**Answer:**

Firstly, it makes the most optimistic assumption about a running process. Namely, that the process will return all of its resources and terminate normally.

Secondly, it also makes the most pessimistic assumption about a running process. Specifically, that the process immediately asks for all the resources it can.

**Q1 d) (Total: 6 marks).**

Q1 d) (i) (4 marks).

**Answer:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***Need*** | | | |
| **A** | **B** | **C** | **D** |
| **P0** | 2 | 2 | 2 | 2 |
| **P1** | 3 | 2 | 0 | 0 |
| **P2** | 0 | 3 | 2 | 4 |
| **P3** | 2 | 5 | 0 | 2 |
| **P4** | 2 | 0 | 0 | 1 |

Q1 d) (ii) (2 marks).

**Answer:**



**END OF QUESTION ONE**

### QUESTION TWO (total: 40 marks): Memory Management

**Q2 a) (4 marks).**

**Answer:**

Logical address and physical addresses must be the same in compile-time and load-time address binding schemes because...

Logical and physical addresses can differ in the execution-time address binding scheme because, at runtime, mapping from virtual to physical address is done by the MMU. Therefore, depending on the scheme (for example, a scheme using a relocation register), the addresses may differ.

**Q2 b) (3 marks).**

**Answer:**

* Paging out (and, in turn, paging in) frequently used pages may increase thrashing
* Paging out (and, in turn, paging in) frequently used pages may complicate memory management
* Paging out (and, in turn, paging in) frequently used pages may result in increased internal fragmentation

**Q2 c) (3 marks).**

**Answer:**

True, recall that internal fragmentation represents the wasted (unused) space within a page. Therefore, by increasing page size, we increase the amount of memory that could potentially be wasted after each allocation which, in turn, increases internal fragmentation.

**Q2 d) (Total: 6 marks).**

Q2 d)(i) (3 marks).

**Answer:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 8 | 2 | 3 | 8 | 1 | 8 | 7 | 8 | 7 | 8 | 3 | 0 | 9 | 0 | 1 | 2 | 8 | 1 | 2 |
| 7 | 7 | 7 | 7 | 7 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 1 | 1 | 1 | 1 | 1 |
|  |  | 2 | 2 | 2 | 2 | 2 | 7 | 7 | 7 | 7 | 7 | 7 | 9 | 9 | 9 | 9 | 8 | 8 | 8 |
|  |  |  | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 |

Using the above table, we see that there are 11 page faults.

Q2 d)(ii) (3 marks).

**Answer:**

False, as it results in the removal and addition of pages which are used later. For example, page 7 is removed from memory and then re-added later, resulting in a higher number of page faults than necessary.

**Q2 e) (4 marks).**

**Answer:**

|  |  |  |
| --- | --- | --- |
| T | = | (20 + (2 x 100)) x (1 - 0.95) + (20 + 100) x 0.95 |
|  | = | (20 + 200) x 0.05 + 120 x 0.95 |
|  | = | 220 x 0.05 + 120 x 0.95 |
|  | = | 125 ns |

**Q2 f) (6 marks).**

**Answer:**

Since the page size is 2^2 bytes (as 4 = 2^2), it takes 2 bits to distinguish individual addresses between 0 and 3.   
  
Recall that an address space is divided into the first-level page table, the second-level page table and the page offset. Since the second-level page table has 2^14 entries, 14 bits are required to represent the entire set of those entries. Furthermore, since the page offset has to be capable of indexing 4 bytes, it is, therefore, 2 bits in size (as 4 = 2^2). We can calculate the size of the first-level page table as follows:

38 = x + 14 + 2

38 = x + 26

x = 12

Therefore, the first-level page table needs 12 bits.

**Q2 g). (3 marks).**

**Answer:**

False, as FIFO replacement suffers from Belady’s anomaly; more frames result in more page faults.

**Q2 h). (4 marks).**

**Answer:**

**0, 600**

P = 100 + 600 = 700

**1, 0**

P = 2300 + 0 = 2300

**2, 500**

The offset in the logical address is larger than the length of the segment corresponding to that logical address (500 > 400). Therefore, this memory access is illegal.

**3, 500**

P = 500 + 500 = 1000

**Q2 i). (4 marks).**

**Answer:**

|  |  |  |
| --- | --- | --- |
| 250 ns | = | (1 - P) x 100 ns + 0.3P x 5000 ns + 0.7P x 12000 ns |
| 250 | = | 100 - 100P + 1500P + 8400P |
| 150 | = | -100P + 1500P + 8400P |
|  | = | 9800P |
| P | = | ~0.0153 |
|  | = | 1.53% |

**Q2 j). (3 marks).**

**Answer:**

False, the working-set model seeks to model the *total* number of frames process i requires. Recall that by computing the working-set size for a process, WSSi, we can then consider that D is the total demand for frames. Since each process is actively using the pages in its working set, we can then conclude that process i needs WSSi frames.



**END OF QUESTION TWO**

### QUESTION THREE (total: 28 marks): File, I/O, and disk

**Q3 a) (4 marks).**

**Answer:**

False, ...

**Q3 b) (3 marks).**

**Answer:**

True, as different components need to produce and/or consume I/O information but may have different transfer speeds. Therefore, caches are used to solve this transfer problem; they, in turn, remedy the producer–consumer problem as well.

**Q3 c**) **(3 marks).**

**Answer:**

False, data striping involves the division of disks into blocks–distributing user and system data across all disks present in the array. The goal of such an operation is to increase effective transfer rate *not* reliability.

**Q3 d**) **(3 marks).**

**Answer:**

True, as disk space is always allocated in fixed-sized blocks. Therefore, whenever a

file is written to disk, it may not fully occupy all of the blocks which were allocated

for its storage and thus, the last block may not be completely full which, of cause,

represents internal fragmentation.

**Q3 e**) **(3 marks).**

**Answer:**

False, The rotation time is ~11.11 ms (as 1 / 5400 = 1 minute / 5400 rotations = 1 second / 90 rotations = 11.11). However, the average rotation time is actually ~5.6 ms (as 11.11 / 2 = 5.6).

**Q3 f**) **(4 marks).**

**Answer:**

The schedule is 12, 2, 40, 38, 22, 10, 6. Therefore, the seek table is given by:

|  |  |
| --- | --- |
| **Head Position** | **Seek Distance** |
| 12 | - |
| 2 | | 2 - 12 | = 10 |
| 40 | | 40 - 2 | = 38 |
| 38 | | 38 - 40 | = 2 |
| 22 | | 22 - 38 | = 16 |
| 10 | | 10 - 22 | = 12 |
| 6 | | 6 - 10 | = 4 |

Since a seek takes 5 ms per cylinder moved and the seek distance is 82, the seek time is 410 ms (as 5 x 82 = 410).

**Q3 g) (4 marks).**

**Answer:**   
Since 512 bytes = 2^9 bytes and 4 bytes = 2^2 bytes, the indirect block holds 2^7 block numbers (as 2^9 / 2^2 = 2^7). Therefore, the maximum file size is 73,728 bytes (as 2^7 x 2^9 + (16 x 2^9) = 73,728).

* The **single indirect**2^7 x 2^9 = 65,536 bytes
* The **16 direct blocks**(16 x 2^9) = 8,192 bytes

**Q3 h) (Total: 4 marks).**

Q3 h) (i) (2 marks).

**Answer:**

This write operation consists of two different write operations. Firstly, we must read 199 blocks to find the block before the last, then we must write the first of the new blocks somewhere with the next block pointing to the block after the 199th block (i.e. the last block); representing a total of 201 operations. Secondly, we must now read 200 blocks to find the block before the last, then we must write the first of the new blocks somewhere with the next block pointing to the block after the 200th block (i.e. the last block); representing a total of 202 operations. Therefore, the total number of operations is 403.

Q3 h) (ii) (2 marks).

**Answer:**

This operation consists of 2 operations, as we read from index 9 directly then we write to index 5 directly.



**END OF QUESTION THREE**

**QUESTION FOUR (total: 12 marks): Protection and Security**

**Q4 a) (3 marks).**

**Answer:**

True, as it may result in a process having too many rights at its different phases. For example, in one phase a process may need read access to an object and in another it may need write access. By using static allocation, the process will receive both read and write access *for its entire lifetime*. Since the principle of least privilege requires that a process should operate using the least amount of privilege necessary to complete the *current* job, using static allocation will lead to the violation of this principle.

**Q4 b) (4 marks).**

**Answer:**

Yes, as Anna can print and all accounts are able to switch to Anna’s account. John is able to switch directly to Anna’s account; Students are able to switch to John’s account, then to Anna’s account; and Staff are able to switch to Student’s account, then to John’s, then to Anna’s. Therefore, all users can use the printer.

**Q4 c) (5 marks).**

**Answer:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | fileA | fileB | fileC | fileD |
| anna |  |  |  |  |
| john |  |  |  |  |
| staff |  |  |  |  |
| OS |  |  |  |  |
| student |  |  |  |  |



**END OF EXAM PAPER**