Distance Mode Exam in TDIU11

Ahmed Rezine

2020-06-09 kl:14-18

Jour

Ahmed Rezine: email (`ahmed.rezine@liu.se`), phone (013-28 1938).

Instructions

- You may access your own lecture or lesson notes, books, or search the internet.
- No contacts, whether physical or virtual, are allowed during the duration of the exam with any person, whether the person is related to the course or not, except for contacting the examiner via email for clarifications if needed (write [TDIU11] in the subject).
- **Any suspected breach will be systematically reported to the disciplinary board**. We use Urkund (an automatic tool to combat plagiarism) after you submit via Lisam.
- Your **anonymous-id** can be found on the Lisam submission page next to the pdf and the deadline. The anonymous-id is of the form “A-XX..X” where “XX..X” is a sequence of digits. Examples of anonymous-ids include “A-3252” or “A-10564” or “A-97600”, etc. These are only examples. You will need to find your own (likely different) anonymous-id on the submission page. Ask the examiner by email if you do not find it.
- Several questions will refer to (some of) the last 4 digits “$D_1D_2D_3D_4$” of your anonymous-id. For example, if your anonymous-id is “A-3252”, then D_1=3, D_2=2, D_3=5 and D_4=2. If it is “A-10564”, then D_1=0, D_2=5, D_3=6 and D_4=4. If it is “A-97600”, then D_1=7, D_2=6, D_3=0 and D_4=0. These are only examples. You will need to find your own anonymous-id on the submission page and deduce your own digits D_1, D_2, D_3 and D_4. Ask the examiner by email if you do not find it.
- The questions use the modulo operator %. For instance, 8%3 = 2 and 8%4 = 0.
- The problems are not ordered according to difficulty. You are encouraged to read all problems carefully and completely before you begin. The answers need to be ordered (even if you leave some questions unanswered).
- You may answer in either English or Swedish.
- Do not take pictures of your solutions or draw them. We only accept PDF files obtained from a text editor or a word processor. You are free to use any text editor (examples include notepad, vim, gedit, emacs, etc) or other text-based office programs (Microsoft Word or Open/Libre Office).You can write `2^24` to mean 2 to the power of 24. We expect you to generate and to submit, via Lisam, a single PDF with ordered answers.

- Be precise in your statements and clearly motivate all statements and reasoning. Explain calculations and solutions. Ambiguous or unclear answers will give fewer to no points.
- If in doubt about a question, write down your interpretation and assumptions. Be coherent.
- Grading: U, 3, 4, 5. The preliminary grading thresholds for p points are: $0 \leq p < 20$: U, $20 \leq p \leq 30$: 3, $31 \leq p \leq 35$: 4 and $36 \leq p \leq 40$: 5.

Problem 1 (12pt)

Assume a tick is a time unit. Assume four tasks t_a, t_b, t_c and t_d. Suppose the execution time of t_a is 2 ticks, the one of t_b is 4 ticks, the one of t_c is 6 ticks and the execution time of t_d is 8 ticks. The indices a, b, c and d need to be deduced from your digits $D_1D_2D_3D_4$ (see instructions on page 1 about how to extract these digits from your anonymous-id) in the following way:

- the index a is such that D_a is the smallest value (and leftmost in case of ties) among D_1, D_2, D_3 and D_4,
- the index b is such that D_b is the smallest value (and leftmost in case of ties) among D_1, D_2, D_3 and D_4 that is different from the index a,
- the index c is such that D_c is the smallest value (and leftmost in case of ties) among D_1, D_2, D_3 and D_4 that is different from both indices a and b
- d is the remaining index, i.e. $d \in \{1, 2, 3, 4\} \setminus \{a, b, c\}$.

Here come some examples that associate digits $D_1D_2D_3D_4$ to values a, b, c, d (these are only examples, recall you need to extract your own $D_1D_2D_3D_4$ digits using the anonymous-id you find on the submission page on Lisam):

- if $D_1D_2D_3D_4 = 5070$ then $a = 2, b = 4, c = 1, d = 3$,
- if $D_1D_2D_3D_4 = 7050$ then $a = 2, b = 4, c = 3, d = 1$,
- if $D_1D_2D_3D_4 = 9631$ then $a = 4, b = 3, c = 2, d = 1$,
- if $D_1D_2D_3D_4 = 7777$ then $a = 1, b = 2, c = 3, d = 4$,
- if $D_1D_2D_3D_4 = 4554$ then $a = 1, b = 4, c = 2, d = 3$,
- if $D_1D_2D_3D_4 = 7667$ then $a = 2, b = 3, c = 1, d = 4$

Remember to:

- Clearly state the execution time of each of the tasks: t_1, t_2, t_3 and t_4. For this, you know the execution times of t_a, t_b, t_c and t_d and you need to associate a, b, c and d to different values in $\{1, 2, 3, 4\}$ depending on your $D_1D_2D_3D_4$ digits as described above. Ask the examiner by email if this is unclear.
- Write down your interpretation and assumptions and to be coherent.

Questions:

1. Assume Round Robin scheduling with a time quantum of 3 ticks.
   a) Fill in the table above. This corresponds to a Gantt diagram. Specify, at the start of each tick, which task is running and which are in the ready queue. (2pts)

event	tick	running	ready queue
-	0	none	none
t_1 arrives at start of	1		
t_2 arrives at start of	2		
t_3 arrives at start of	3		
t_4 arrives at start of	4		
-	5		
-	6		
-	7		
-	8		
-	9		
-	10		
-	11		
-	12		
-	13		
-	14		
-	15		
-	16		
-	17		
-	18		
-	19		
-	20		
-	21		
-	22		

b) What is the waiting time of each tasks. (2pts)

2. Assume a preemptive SJF scheduling algorithm.
   a) Fill in the table above. This corresponds to a Gantt diagram. Specify, at the beginning of each tick, which task is running and which are in the ready queue. (2pts)
   b) What is the waiting time of each task. (2pts)
3. Suppose tasks of different execution times are generated by an environment we do not control. Suppose the environment generates tasks at a rate (the actual value of the rate is not relevant for this question) for which the size of the waiting queue is never exceeded. Name and explain a problem that may occur during SJF scheduling but that cannot occur duing Round Robin. (2pts)
4. Explain how to remedy to this problem and describe how to combine this remedy with SJF scheduling. (2pts)

Problem 2 (13 pt)

Consider a filesystem on a disk with physical blocks of 512 bytes each. Assume, as usual, that logical blocks are concatenations of n contiguous physical blocks. Suppose the filesystem uses indexed allocation with, in each inode, 12 direct pointers, one indirect pointer, one double indirect pointer and one triply indirect pointer.

1. Let $n = 2 \times 2^{(D_4\%3)}$ where D_4 is the last digit of your anonymous-id. What is the size in bytes of a logical block? From now on, assume this value for the logical blocks. (2pts)

2. Assume block pointers are 4 bytes long. What is the maximum size of a hard drive if all logical blocks can be pointed to using a block pointer? (2pts)
3. What is the maximum size of a file? (2pts)
4. Is a double indirect indexing level sufficient to index 2^{11} logical bocks of data? how many logical blocks can a triple indirect indexing level index? Explain. (2pts)
5. Suppose your disk contains 5000 files of unknown sizes (the sizes of the files are not relevant for this question). Estimate, and clearly explain, the internal fragmentation in the data blocks. (2pts)
6. Assume 500 us (500×10^{-6}s) access time and all pointers are already loaded in memory. How long would it take to read a 4GiB image from the disk if half the logical blocks require a new seek (you can neglect transfer time)? Explain. (3pts)

Problem 3 (11pt)

Memory management and virual memory. Let $n = 2 + (D_4\%3)$ where D_4 is the last digit in your anonymous-id. Assume a paged virtual memory with 32 bits logical addresses and 32 bits physical addresses.

1. Assume n level (recall $n = 2 + (D_4\%3)$) paging. Assume page entries of 4 bytes each. Describe which bits of a logical address are associated to each level. Explain. (2pts)
2. What is the maximum number of frames your system can have? Explain. (1pt)
3. What is the size of the page table at each level? what is the smallest amount of memory taken by the page tables in case a process only uses one page of data. (2 pts)
4. Describe, by showing possible contents of page entries at each level, how logical address 0000 0000 0000 0000 0000 0000 0000 0000 can be translated into the physical address 1111 1111 1111 1111 1100 0000 0000 0000. (2pts)
5. Why do pages need to be replaced? explain why a fifo replacement mechanism is simpler to implement than an LRU mechansim. (2pts)
6. What would be the information kept in a TLB for your system? Why do some TLBs make the effort of storing, for each TLB entry, an identifier associated to a process? Give a scenario where this information is useful. (2pts)

Problem 4 (4pt)

A replay attack consists in an attacker listening and repeating valid data (e.g., passwords) in order to gain access to secret data. One mechanism to resist replay attacks in password authentication is to use one-time passwords. The idea is that the password is unusable by the time the attacker learned it by repeating it. For this, a list of n passwords is prepared and shared between the system and the user in a secure way. Passwords can only be used once.

Instead of sharing a big list of n passwords, a system and a user program can instead share a single “main password” pw. They can change pw after each n authentications (the mechanism for changing the main password in a secure manner is not relevant for this problem).

Assume $n = 2 + (D_4\%3)$ where D_4 is the last digit of your anonymous-id. Describe a mechanism that allows a user program and a system to have the same list of n passwords while sharing only one single “main password” pw. Explain how your mechanism can derive the i^{th} password (for each $i = 1, 2, \ldots n$) from the current "main password" pw. The mechanism can be public (everyone, including the attacker) knows how the mechanism works, but no attacker should be able to deduce the passwords unless it knows the main password. (hint: one way hash functions like those used to save hash functions of salted passwords)