Distance Mode Exam in TDIU11

Ahmed Rezine

2020-03-26 kl:14-19

Jour

Ahmed Rezine (013-28 1938).

Instructions

- Read instructions and all assignments carefully and completely before you begin.
- The exam is to be answered individually. Collaborating or discussing the questions or communicating answers or suggesting tentative solutions is strictly not allowed during the duration of the exam. We will use tools to control plagiarism and the disciplinary board will be notified if plagiarism is suspected.
- You can use any available literature (e.g., your notes from the course or the lessons or the course book) but it is strictly not allowed to work, discuss or communicate exam questions or answers with another person, related to the course or not, except for possibly asking for clarifications from the examiner during the duration of the exam.
- It should be enough to use a word editor or even a text editor for your answers. Write `2^{24}` to mean 2 to the power of 24.
- You can contact the examiner via mail during the exam (write [TDIU11] in the subject).
- You may answer in either English or Swedish.
- State your interpretation of the question and all assumptions you make.
- Be precise in your statements. Ambiguous formulations will lead to reduction of points.
- Motivate clearly and in depth all statements and reasoning. Explain calculations and solution procedures.
- The exam is 40 points and graded U, 3, 4, 5 (**preliminary** limits: 20pt, 31pt, 36pt).

event	tick	running	ready queue	waiting queue
-	0	none	empty	$\{P1, P2, P3, P4\}$
E4	1			
E1,E3	2			
E2	6			
	...			

Problem 1 (12pt)

Assume a drone where processes have the possibility to send radio communications. Assume an operating system with a preemptive and priority based scheduler. Suppose processes need to gain exclusive access to the radio before they can use it. For this, a process needs to first acquire the exclusive access to the radio (similar to a lock). It can then use it and release the exclusive access (similar to an unlock). When a process tries to acquire the radio while the radio is busy (no matter the priorities) the process is put on the waiting queue until the exclusive access to the radio is released by the process that held it. The requesting process is then put back to the ready queue where it can compete for the CPU with the other ready processes. A process can preempt a process with a lower priority even if the lower priority process is currently using the radio. At tick 0 (our time reference in this problem) the system has an empty ready queue and an idle CPU. The waiting queue contains the following jobs:

- P1 has priority 1 (highest) with an expected execution time of 2 ticks. It is waiting for event E1. As soon as it is put to run, Process P1 directly tries to acquire the radio and keeps it for 1 tick before releasing its exclusive access to it.
- P2 has priority 2 with an expected execution time of 2 ticks. It is waiting for event E2.
- P3 has priority 3 with an expected execution time of 5 ticks. It is waiting for event E3.
- P4 has priority 4 (lowest) with an expected execution time of 3 ticks. It is waiting for event E4. As soon as it is put to run, Process P4 directly tries to acquire the radio and uses it for 2 tick before releasing its exclusive access to it.

Questions:

1. Fill in the table above (you may need to insert more rows). This corresponds to a Gantt diagram. Specify, at the beginning of each tick, which (if any) process is running, which are in the ready queue and which are in the waiting queue. What is the waiting time of each process (6pt).
2. Between tick 1 and tick 2 some process is running on the CPU (not the kernel). This process could continue until completion, but instead another process with higher priority might get scheduled instead. Observe the kernel was not running at the preemption time. Explain how the kernel manages to regain control of the CPU in order to schedule other processes with higher priorities. (2pt)
3. What would the waiting time of P1 be if P3 was 10 ticks long (instead of 5 ticks as above)? P1 was assigned a high priority so it would not need to wait too long. In fact we do not want P1 to wait for more than 2 ticks (the longest time the radio is used by a job with a lower priority). What went wrong? Explain. (1pt)
4. How can you change the scheduling policy to ensure a higher priority job does not have to wait for lower priority jobs that do not use the radio. (1pt)

5. Adopting your proposed new policy, fill in the table again (there might be a different number of rows). What is the new waiting time of each process? (2pt)

Problem 2 (6 pt)

Assume a 4TiB (i.e., 2^{42} bytes) hard drive with 512 bytes logical blocks and a corresponding FAT32 file allocation table with 4 bytes for each table entry.

1. How large should logical blocks be in order for the table to be at most 128MiB large (2pt).
2. Assume two such disks are mounted. Describe the steps involved in copying a large file (say a 4GiB image) from one disk to the other. (2pt).
3. Assume 500 us (500×10^{-6}s) access time and the FAT tables have already been loaded in memory. How long would it take to copy the 4GiB image from one disk to the other if half the logical blocks require a new seek (you can neglect transfer time). Explain. (2pt)

Problem 3 (14pt)

Memory management and virual memory.

1. Assume a memory management system that uses segmentation. Give two advantages of using segmentation as compared to using contiguous allocation. Explain. (2pt)
2. What fragmentation problem does segmentation suffer from and that is solved by paging? Give an example, with a sequence of segments with concrete sizes, that shows how segmentation suffers from such a fragmentation problem and explain how paging completely avoids the problem. Explain. (4pt).

Assume a one level page table on a system with 32 bits logical addresses and 48 bits physical addresses. Page table entries are assumed to take 4 bytes each.

1. Describe two different possible choices of page sizes and explain, for each choice, how many bits are available in each page entry for information other than the corresponding frame number. (2pt)
2. Give an advantage and a disadvantage for each of the two choices. Explain. (2pt)
3. Pick one of your two choices. What information is cached in the TLB (translation look-aside buffer) of this system? Explain what for? (2pt)
4. Suppose access to main memory takes 10 nanoseconds (10×10^{-9}s). Address translation in a one level page table results in a performance degradation compared to directly accessing the target address. The TLB strives to diminish this degradation. What should the TLB hit ratio be in order to not suffer more than a 1% performance degradation due to address translation? Explain (You can neglect TLB access time). (2pt)

Problem 4 (8pt)

1. To be able to execute recently compiled executables in the current directory, a user sometimes uses the following search path ”PATH=.:/usr/local/bin:/usr/bin” (returned by ”echo $PATH”). Describe and explain what is the problem with this behaviour. (2pt)

2. How come salting a password protects the password even when the salt is stored in plaintext in the same file as the hashed salted password? (2pt)

3. One way to give a user a way to detect both malicious and unintentional modifications of (certain) files in a filesystem is to compute checksums for each such file. This helps detecting certain viruses that plant themselves in files or diagnosing hardware errors.

   a) A program is written to record checksums for files in the system. Any user should be able to use the program to verify the checksum of any file. Explain how you would use setuid to make sure the program can access any file. Explain how you adapt the implementation of the program to make your setuid solution as secure as possible according to the principles of least-privilege and need-to-know (2pt)

   b) The setuid program takes the path to some file and returns whether its checksum matches the stored checksum. Explain how such a program might allow a malicious user to execute arbitrary code as root? (2pt)