Operating Systems, course exam, 7.3.2018

Write in each answer sheet course name, date, your name, signature and student id.
Please notice that the exam paper is 2-sided. Please write the answer to each problem in its own answer sheet. For each problem, it is sufficient to give a 1-2 page answer.
Each problem is also the miniexam of the same number. Problem 4 is miniexam 4.

NOTE: Please return the answer for each problem in its correct pile!

1. [12 p] **Cache, Threads, Concurrency**
   a. [4 p] Assume that process P has three threads, and thread T will do a blocking operating system call. What happens to the execution of (other threads of) process P, if the threads are ULT? What happens if the threads are KLT (kernel level threads)? Which one (ULT or KLT) is better in this case and why?
   b. [4 p] Give a pseudocode example on a critical section problem where critical section is composed of two different code segments. Give a scenario, which has erroneus result. Show how this critical section problem is solved with monitor.
   c. [4 p] We have 80 character buffer B. Process P writes to B every now and then. If B is full, P must wait until there is room in B. Process Q reads from B every now and then. If B is empty, Q must wait until there are characters to read in B. Give the pseudocodelevel semaphore-based solution to the synchronization problem between P and Q.

2. [12 p] **Deadlock, memory management**
   a. [2 p] Which four conditions must be valid that a deadlock is possible. Explain what each of those four conditions mean.
   b. [2 p] Give an example on a situation where all four conditions discussed above are valid, and a deadlock has occurred. Explain, why deadlock occurred.
   c. [2 p] Give an example on a situation where all four conditions discussed above are valid, but there is still no deadlock. Explain, why deadlock did not occur this time.
   d. [2 p] Which memory management problem is solved with Buddy System? How does the solution work in main principles? Can there be internal/external fragmentation with Buddy System? Explain.

3. [12 p] **Virtual memory, Scheduling**
   a. [3 p] How does virtual memory address translation happen in practice when we have 2-level paging virtual memory, 32-bit byte addresses, 4KB page size, and referenced address 0x12345678.
   b. [3 p] The frame to be replaced can be located with Clock algorithm. In its basic form, the Clock algorithm has the problem that the selected frame may be dirty, and the page in it must be written to disk before the frame can be reused.
   How can this be avoided, and how does the modified algorithm work in main principles? What data does the solution need and how do you obtain it from running processes?
   c. [2 p] Why would Shortest Job First (SJF) scheduling be better than First Come First Served (FCFS)? What specific problem is there with SJN, and how can you solve it in practice?
   d. [4 p] A real-time system needs to handle concurrently (i) one voice phone call and (ii) one video stream. The phone call runs every 5 msec and consumes 0.5 msec of CPU time per burst. The video stream has 25 frames/sec, with each frame requiring 15 msec of CPU time. These processes are schedulable with RMS (rate monotonic scheduling). Explain why the processes are schedulable with RMS and what is the resulting schedule.

TURN

4. [12 p] **I/O, File mgt, Security**
   a. [3 p] Describe three problems with the SCAN (elevator) scheduling algorithm, and how they can be solved with?
   b. [3 p] We have 6 hard disks (D1-D6), each 1 TB, block size is 2 KB, and we use RAID-5.
   What is the total capacity of this RAID disk system?
   A small (764B) file FileA on disk D3 is (i) opened for editing, and finally (ii) written back.
   Which blocks are read from or written to each disk and when?
   (You may unrealistically assume, that no other process using the disks is concurrently in execution.)
   c. [3 p] Assume that system access control is role based and uses access control lists.
   What data structures are involved?
   Semaphore S use is limited to kernel level processes, which each belong to group KRNL. How do you check in practice that kernel process DD-DSK is allowed to perform s_wait operation to semaphore S? Who does the checking and when?
   d. [3 p] How does Windows 7 access control work in main principles?
   File MyPic is owned by teacher Hannu, and only he can write to it. File MyPic can be read by teachers Arto and Tiina, and by all students except Kalle.
   How do you implement this with Windows 7 entries in discretionary access control list (DACL)?
   The general format of one access control entry (ACE) is
   {allow/deny, read/write/exec/all, user/group/all}