Operating Systems, 11.5.2017

Write in **each answer sheet** course name, date, your name, signature and student id.
For each question, it is sufficient to give a 1-2 page answer.
Please write the answer to each problem in **its own answer sheet**.
Each problem is also a miniexam of the same number. The replacement exams for miniexams 4-7 are problems 4-7. Miniexam 8 is problem 8. Term exam is problems 5-8.
NOTE: Please return the answer for each problem in the **correct pile**!

4. [6 p] **Virtual memory**
    a. [3 p] Page table has 1024 entries, page size in 4 KB, and the system uses byte addresses.
    How is the 2-level paged virtual memory address translation done at hardware and software level?
    How long will the address translation take in different cases? Do not forget TLB.
    As an example, use virtual address 0x00C0 400F.
    b. [3 p] How can you estimate in virtual memory system, how many page frames (how much memory) each process needs? Give one solution method and explain, how it works in main principles.
    Why is it bad, if some process is allocated far too few page frames?
    Why is it bad, if some process is allocated far too many page frames?

5. [6 p] **Process scheduling**
    a. [2 p] What would be good time slice length for Round Robin scheduling?
    Why is it bad, if the time slice is too short? Why is it bad, if the time slice is too long?
    b. [2 p] Which problem in usual Round Robin scheduling is solved with Virtual Round Robin, and how does the solution work?
    c. [2 p] What is the priority inversion problem and how is it related to scheduling? In what type of system can it occur? Why is it a problem? Give one method to avoid the priority inversion problem.

6. [6 p] **I/O management, file systems**
    a. [2 p] Describe three problems with the SCAN (elevator) scheduling algorithm, and how they can be solved with?
    b. [2 p] We have 6 hard disks, each 1 TB, block size is 2 KB, and we have RAID-5.
    What is the total capacity of this RAID disk system?
    What happens in this disk system, when one opens a small (764B) file for editing, and then writes it back after editing?
    c. [2 p] We have indexed file. How many indexes are needed?
    What record reference pattern would this file structure be exceptionally bad for?
    Why would it be useful to use B-tree for index implementation?
    How do you find the referenced record with a B-tree?

TURN

7. [6 p] **Embedded systems, distributed systems**
    a. [3 p] There are no monitors defined in eCos, but one can still write code that behaves very similarly to monitors. How does this work in main principles?
    How is the monitor mutex problem solved here?
    Does the eCos monitor-like solution follow signal and wait (Hoare), or signal and continue (Lampson and Redell) signalling semantics? Explain.
    b. [3 p] Remote procedure Call (RPC) is very handy to use remote services in a distributed system. However, in some environments the location for some given service changes often.
    How does the service user (client C) know the current network address for such service (server S)?

8. [6 p] **Data security**
    a. [3 p] How does Windows 7 access control work in main principles?
    As an example, use object Sem, that can be used by privileged kernel processes (e.g., P), but that can not be used by user level processes (e.g., U).
    b. [3 p] Why is metamorphic virus more advanced than polymorphic virus?
    Why is metamorphic virus more difficult to locate from a system than polymorphic virus?
    How can you locate a known metamorphic virus from your system?