Operating Systems (5 op), final exam/renewal exam 17.4.2019

Write in each answer sheet course name, date, your name, signature and student id.

No calculators. Exam time 3.5 h. Exam paper is 2-sided. For each question, it is sufficient to give a 1-2 page answer.
Please select one of the following and mark your selection in your exam paper. The default selection is (b).
(a) Replacement exam for Spring 2019 lecture course miniexams 1-4. Answer to 1-4 miniexams.
(b) This is ordinary final (separate) exam and covers the whole course. Answer all questions 1-4.

1. [9 p] **Concurrency**
    a. [3 p] Explain,what is the critical section problem, and why it is difficult to solve.

    b. [3 p] When and how can the critical section problem be solved purely with software (with shared memory, memory read and write instructions)?
    When and how can the critical section problem be solved with disabling interrupts?
    When and how can the critical section problem be solved with special instructions (e.g., Compare&Swap)?

    c. [3 p] Running track. Running track is 400m long. Ann and her friends Bill and Charlie come there often to run 4000m. Ann is social and waits after each round that both boys have caught up with her (at least the same number of rounds run). The boys are competitive, and do not wait for anybody.

    Solve the resulting synchronization problem with semaphores. Give your solution by modifying the runner pseudocodes below. Remember to define your semaphores with their initial values.

```
Ann                 Bill                Charlie
===                 ====                =======
for (i=1 to 10)     for  (i=1 to 10)    for  (i=1 to 10)
  <run one round>     <run one round>     <run one round>
  <synchronize>       <synchronize>       <synchronize>
```

2. [9 p] **Deadlock, memory management**
    a. [3 p] Operating system processes P and Q update shared data structures A and B. The updates for both A and B must be done atomically. A is protected by critical section cs-A, and B by critical section cs-B.
    How should you write the code for P and Q so that you can quarantee that deadlock will not happen? For performance reasons, P and Q are not allowed to always reserve cs-A and cs-B at the same, because most of the time P and Q use only one critical section at a time.
    Explain what problems there are in implementing the code for P and Q. Why is the solution not trivial?
    Explain why your solution will quearantee that deadlock will never happen.

    b. [3 p] Which memory management problem is solved with Buddy-system, and how does it work? What advantages does Buddy-system provide, as compared to other solutions to the same problem?

    c. [3 p] How can you know if some process does not have enough memory (page frames) in virtual memory system?
    Give some solution method, that would dynamically allocate more memory to such processes. Give the fundamentals for such solution method.

TURN

3. [9 p] **Scheduling**
   a. [1 p] On which data is (scheduling algorithm) FIFO based on and how is this data obtained?
   Is FIFO preemptive algorithm or not?

   b. [2 p] On which data is SPN (Shortest Process Next) based on and how is this data obtained?
   Is SPN preemptive algorithm or not?

   c. [2 p] On which basis is SPN better than FIFO?
   On which basis is FIFO better than SPN?

   d. [2 p] Real time systems scheduling is usually based on deadlines.
   Explain, how deadline scheduling works, in general.

   e. [2 p] RMS (Rate Monotonic Scheduling) avoids most problems in deadline scheduling.
   On which data is RMS based on and how is this data obtained?
   Why RMS can not be used for all real time systems scheduling?

4. [9 p] **Disk and file management, access control**
   a. [3 p] Which problem relating to hard disk (HDD) management is solved with C-SCAN algorithm, and how does that algorithm work in main principles?
   Which problem relating to C-SCAN algorithm is solved its variant Linus Deadline Scheduler, and how does that variant algorithm work in main principles?
   Which problem relating to C-SCAN algorithm is solved its variant Linus Anticipatory I/O Scheduler, and how does that variant algorithm work in main principles?

   b. [3 p] Indexed file.
   How many indexes does it have? How large is the index (indexes)?
   Give an example use case where indexed file would be the best file structure for your file.
   Why would it be sensible to implement the index in an indexed file with a B-tree, and not as ordinary sequential index?

   c. [3 p] Role-based access control (RBAC). As a professor, Jussi has the right to edit thesis database THESIS, and as staff member to read the internal bulletins in database BULLETINS.
   Show the RBAC access control matrix (or matrixes) with which these access rights are implemented in the system.