Operating Systems (5 cr), Exam 27.11.2019

Write in each answer sheet course name, date, your name, signature and student id. No calculators.
For each question, it is sufficient to give a 1-2 page answer.
This is ordinary final (separate) exam and covers the whole course. Answer all questions 1-4.

1. [9 p] **Concurrency control**
   Subroutine Sub1 can be used by many concurrent threads. It has the code segment

```
...
Count = Count+1;
LocalCount = Count;
if (LocalCount > 100)
   LocalCount = 100);
x = y+5;
...
```

   where Count is shared variable for all threads, and all other variables are local. Threads T1, T2, and T3 in process P all use Sub1 every now and then. Variable Count is not referenced elsewhere in shared code.
   a. [3 p] Give a scenario where P does not give correct result. Explain what is the problem in the code.
   b. [3 p] Can this concurrency control problem be solved with disabling interrupts? If it can not, explain why not. If it can, explain how.
   c. [3 p] Can this concurrency control problem be solved with semaphores? If it can not, explain why not. If it can, explain how.

2. [9 p] **Deadlock and synchronization**
   a. [3 p] Which four conditions must be met for it to be possible to have a deadlock?
   How are they met in the deadlocking solution to the Dining Philosophers problem?
   b. [3 p] What is the producer-consumer problem? Consider infinite buffer size case separately.
   c. [3 p] Give semaphore-based solution to the (finite buffer) producer-consumer problem.

3. [9 p] **Memory management, virtual memory**
   a. [3 p] How is the address translation for 1-level virtual memory done at hardware level and at software level? How long will the address translation take in different cases? Do not forget the TLB.
   b. [3 p] Which problem related to virtual memory is solved with Clock algorithm? How does the solution work in main principles?
   c. [3 p] Which problem related to memory management is solved with Page Fault Frequency (PFF) algorithm? How does the solution work in main principles?

4. [9 p] **Scheduling for real time systems** (RTS)
   a. [3 p] Why in RTS we cannot usually use the same scheduling methods (e.g., FIFO, or priority based solutions) as in ordinary systems? What is RTS scheduling usually based on?
   b. [3 p] However, Rate Monotonic Scheduling (RMS) is a priority based scheduling system especially for RTS. When can you use it and how does it work? How do you select the priorities?
   Why would it be better than other scheduling algorithms for RTS?
   c. [3 p] What is the priority inversion problem relating to RTS scheduling? How can you solve it?