Questionnaire for Laboratory assignment 2
Operating Systems, EDA092, VT2009

The answers to this questionnaire need to be submitted to the *Fire sytem* in order to get the initialization command required for completion of lab 2 (implementation of CPU scheduling and memory management in OSP).
Please submit your answers (including your group number and name at the top of your document) in text (.txt) or pdf to
https://fire.cs.chalmers.se:8036/cgi/Fire-os

The following 16 questions are straight-forward to answer after careful reading of the "Introduktion till OSP", including sections 1.4.3 and 1.5 - 1.6 of the book "OSP: An Environment for Operating System Projects" (Part of the Course-Compendium, available at DC).

General questions about OSP

Q1: What is OSP, and what is the purpose of modules in OSP (what do modules represent)? Give a short answer - not more than 5 sentences.

Q2: Which pointer construction can you use to access the PCB (process control block) of the currently executing process in OSP? (How can the PCB of the running process be localized at any time?)

Q3: SIMCORE offers some help for debugging purposes. Which routines can be used to print the current system state?

Q4: How can you force OSP to print out all events occuring during the simulation? (Which is a very useful debugging feature!)

Q5: What is the purpose of the "-d" flag when executing OSP?

Questions about CPU Scheduling

Q6: Which procedure is responsible for placing a PCB in the queue of ready processes?

Q7: List two actions this procedure (answer of Q6) should perform before inserting a PCB into the queue.

Q8: What is the purpose of the procedure *dispatch()* in the module CPU?

Q9: List at least four actions which should be performed within the procedure *dispatch()*.

Q10: Which routine needs to be called in *dispatch()* in order to implement a preemptive round-robin scheduling (i.e. setting a time interval for the maximal-allowed execution time before the next interrupt)?

Questions about Memory Management

Q11: Which routine needs to be called by *pagefault_handler()* in module *PAGEINT*?

Q12: What are the two main tasks performed by the *get_page()* routine?

Q13: Describe the difference between 'virtual address space' and 'physical address space' and the relation between 'pages' and 'frames'. Give a short answer - not more than 5 sentences.

Q14: When does a 'page fault' occur and what needs to be done by the system?

Q15: List three steps necessary to be performed in the case of no available free frame when *get_page()* is called.

Q16: Which two mandatory actions need to be taken by *deallocate()* when deallocating a frame?