Computer Organization I, exam   20.12.2017

This is the course exam for lecture course. Answer all questions if you have not taken the miniexams. Question 4 is also miniexam 4. If you already took miniexams 1-3, you need to answer only this question. Questions 1-3 are also make-up exams for miniexams 1-3.

Please write the answer for each problem on its **own sheet** and return it in **its own pile**!
It is sufficient to give 1-2 page answer to each question.
Write on **each** answer sheat your name and signature, student id number, and course name.

Please notice, that the exam paper is 2-sided.

1. [10 p] System structure, CPU, bus
   a. [2 p] How does the processor execute machine instructions?
   b. [2 p] What are interrupts? Give two different types of interrupts and explain when they occur.
   c. [2 p] When and how are interrupts processed in the processor?
   d. [2 p] An ordinary program can only reference its own memory areas. How is this implemented in the processor.
   e. [2 p] An ordinary program is not allowed to modify the values of special registers (e.g., BASE) or to clear the cache, but operating systems programs can. How is this implemented in the processor.

2. [10 p] Data representation and correctness, memory
   a. [2 p] What is the 16-bit two's complement little-endian representation of -23?
   What is the 16-bit one's complement big-endian representation of -23?
   b. [2 p] What is the 32-bit little-endian IEEE floating point standard representation for 4.25?
   c. [2 p] How do you represent strings in memory? Use string "String" as your example.
   d. [4 p] Assume that 64-bit data in memory is protected with Hamming code. How many (Hamming code) parity bits is needed? Explain. When do you compute new values to the parity bits? How is Hamming code used in practice?

3. [10 p] Operating systems, external memory, I/O
   a. [2 p] What are the different process states (in simple 5-state model)? When is a process in those states?

   b. [4 p] What are the possible state transitions in the preceding 5-state model? Explain briefly for each possible state transition, under which circumstances that state transition can occur and how is the state transition implemented?

   c. [2 p] A device driver (DD) running on CPU reads pressed keyes from the keyboard, one key at a time. The device driver does this in co-operation with the device controller process (DCP) running on the device controller for that device.

   How is pressing one key handled in this system, when I/O is implemented with indirect I/O (interrupt driven I/O)?
   Mention especially, when data is copied from some place to another, what DD and DCP will do at any given time, and in what process state(s) the DD is in during the I/O.

   d. [2 p] Why are hard disks usually implemented with DMA I/O and not with programmed I/O or interrupt driven I/O (indirect I/O)?

TURN

4. [10 p] Compilation, linking, loading, interpretation, emulation
   a. [2 p] When would you use macroes in the your program instead of subroutines? What advantages do macroes provide as compared to subroutines? What disadvantages?
   b. [2 p] How does the (assembly language) compiler give values to symbols? As examples, use variable X defined at main program level (X DC 65), and address ELSE in a forward jump instruction (JNE ELSE). Where do you store those symbol values?
   c. [2 p] What is the representation of Java program at execution time when it is executed by interpretation?
   d. [2 p] What is the representation of Java program at execution time when it is executed by compilation?
      What advantage does it have as compared to execution by interpretation?
      What disadvantage?
   e. [2 p] What is the representation of Java program at execution time when it is executed by JIT-compilation?
      What advantage does it have as compared to execution by compilation?
      What disadvantage?