Homework #1:

Due: Mar. 27 (to be collected in class)

Name:( ) SNU_ID:( )

Note:

Homework questions in this class are not difficult because we just want to make sure you learned basic stuff. We will grade your answers based on your understanding rather than the correctness of your answers. Therefore, we will give you credits for reasonably wrong answers as long as your story makes a good sense. Please try to do your best and explain your answers. It is important for use to see your reasoning as well as your answers.

Q1) What is 'programmer visible state?' It is also called as 'architectural state.'

(a) Explain in your own words from the perspective of operating system (OS) which does context switching of multiple programs in order.

(b) Please mark whether items below should be in architecture state or not. If not, explain why.

- Program Counter
- 4GB DRAM
- 256KB cache
- Memory-loaded address space
- stack frame
- heap
- ALU (calculator)
- register file

Q2)

(a) If you are an assembly programmer, which ISA would you prefer between RISC (reduced instruction set computer) and CISC (complex instruction set computer)? List key advantages and disadvantages of your computer against the other computer.

(b) What made RISC ISA as a popular choice for modern computer systems? Explain key reasons of such design trend.

(c) However, the most popular ISA, Intel X86, is CISC. Why didn't Intel change to RISC? What have they done to overcome well-known disadvantage of CISC ISA?

(d) What is 'orthogonal' ISA? Why was it popular in old days?

Q3) Consider the following types of instruction set architectures:

1. ***Accumulator:*** All operations occur between a single register and memory location.

   LOAD A – loads the accumulator with the contents of memory location A.

   STORE A – stores the contents of accumulator into memory location A, leaving the accumulator contents intact.

   ADD A – adds the contents of memory location A to the accumulator.

2. ***Stack:*** All operations occur on top of the stack. Push and pop are the only instructions that access memory. ALU operations implicitly use the top two elements of the stack and replace them with the result.

   PUSH A – Pushes the contents of the memory location A on the top of the stack

   POP A – Removes the contents at the top of the stack and stores it in the memory location A.

   ADD – Adds together the top two elements on the stack, removes both the elements from the stack, and stores the computed result on the top of the stack.

3. ***Load-store:*** All ALU operations occur in registers and register-to-register instructions have three register names per instruction. Load and Store are the only instructions that access memory directly.

LOAD R1, A – Loads the contents of memory location A into the register R1.

STORE A, R1 – Stores the contents of register R1 into the memory location A.

ADD R1, R2, R3 – Adds the contents of register R2 and R3 and stores the result in register R1.

We have the following high-level code sequence:

A = B + C

B = A + C

D = A + B

a) Write the assembly code sequence for each ISA above, using the assembly language defined above. Write your code in as few instructions as possible.

b) Assume that every access to memory for data reads/writes 4 bytes. How many bytes of data are transferred between memory and processor for ISA in our code? Please write the amount of data transfer per each instruction in your code (in bytes). Which architecture transfers the least amount of data between memory and processor?

c) Assume your instructions are originally in memory. If your instruction also comes from memory, they cause data movement between memory and CPU due to instructions (i.e., stored-program architecture). If your instruction is in 32-byte, what would be total amount of CPU-memory transfer due to both data and instruction for each ISA?

Q4~Q8) Exercise 2.1 ~ Exercise 2.5 (in P&H 5E)

Q9~Q10) Exercise 2.7 ~ Exercise 2.8 (in P&H 5E)

Q11) Exercise 2.12 (in P&H 5E)

Q12) Exercise 2.18 (in P&H 5E)

Q13~Q15) Exercise 2.26 ~ Exercise 2.28 (in P&H 5E)

Q16~Q21) Exercise 2.31 ~ Exercise 2.36 (in P&H 5E)