Fall 2018 CSC332 HW1 Due Mon 9/24/18

Q1. Suppose we have a correctly working user program. Part of the program reads as:

/\*The statement below computes the difference (value of R1)- (value of R2). Here R1 and R2 are two general purpose registers. Condition codes are set in PSW As usual. The two registers remain unchanged.

\*/

CMP R1, R2

//Jump to location L if the condition code N is set to 1.

JMPN L

Now suppose we replace the above statement “CMP R1, R2” by the following statement:

SYSCMP

The SYSCMP above is a system call instruction.

Below is a pseudo code for its service routine.

Pseudo code of the service routine for SYSCMP

---------------------------------------------

CMP R1, R2

RTI

Questions: Will the new user code necessarily work correctly?

Write “Yes” or “No” and give an explanation in less than 50 words.

Q2. Consider a service routine that responds to a timer interrupt.

Assume that the function of this service routine is simply

to increment the current value of ``clock''-- an integer

stored in absolute address CLOCK. Ignore any overflow

that may arise from this operation.

Below is a pseudo code for this service routine.

Pseudo code of the service routine

----------------------------------

Load CLOCK, R1 (\* loads the integer stored in address CLOCK to register R1 \*)

INCREMENT R1 by 1

STORE R1, CLOCK (\* stores the value in R1 to the memory location CLOCK \*)

RETURN (\* Pops the return address from control stack and loads it to PC \*)

Questions:

(a) Point out any problems with this code, in less than 30 words.

(b) Show a corrected pseudo code.

Q3.

Convert the following code fragment to assembly code fragment. Assume no overflow/underflow. Use these instructions:

Load X, R  loads contents of memory address X to register R

Store R, X stores contents of reg R to memory X

CMP R1, R2 computes (R1)-(R2) but does not change values of R1, R2. Sets condition codes

Jump, Jmpn, jmpp, jmpz as discussed in class

if X > Y then Y=X

else X=Y

Q4

.Suppose we have a system where user programs are loaded at address 0, all addresses generated in any program are treated as physical addresses (i.e., there is no translation from logical to physical address). No memory protection is provided. Assume that the user knows complete details of system software—content and addresses.

The rest of details are the same as discussed in class. Ex. We have dual mode of operation.

Suggest one simple way that the user can change its mode to supervisor mode.