

COMPUTER SYSTEMS ARCHITECTURE -

Written exam. February 8, 2018

Define and explain the concepts of call code, entry code and exit code in the general context of programming languages, after which analyze how are they reflected at the level of x86 assembly language. Why is needed the assembly language involvement in working with these concepts ? Explain, giving source code adequate schematic examples and following the structure of the run-time stack to clarify the involved responsibilities and needs. Who is responsible for generating these codes and when exactly ? Define the notion of stackframe and explain its role and structure. Describe the CDECL and STDCALL call conventions, explaining where are they used.

I a). Which is the MINIMUM number of bits necessary for representing: 0.61 -62 ) 130 M.-129 Justify and explain your answer by detailing the representation mechanisms of these values (Examples: the minimum number of bytes for representing number 5 is 3: 101b; the minimum number of bytes for representing number 16 is 5 10000b). For each of the above numbers give the representation in base 2 and then in base 16.

b). The following ASM sequence is given:

**xor ah, ah cwde : Write one single ASM instruction having the same effect as the given add ebx, eax ; sequence and explain/justify why the same effect is accomplished, mov al, febx] ; Detail the exact effects of every source line from the given sequence**

III. The following ASM sequences are given:

a), mov ax, 1000h

mov bl, 1000b+10b div bl

b) mov ah, Obch

mov al, Odeh

add ah,al

c), mov ax, 1001h

mov bx, 1111b imul bl

d). mov dh, 62h

một ch, 200 sub dh.ch

**Which is the result of the each of the above operations and instructions ? Detail the effect of every source line providing every involved value in each of the 10 and 16 nummeration bases, signed and unsigned. Define and explain the overflow concept, the way in which the x86 architecture reacts at such an event in the case of each of the basic arithmetic operations and specify if such a situation appears in one or more of the above cases, justifying its cause and effect.**

**IV. A string of doublewords is given (defined in module a asm). Build the string of byte ranks that have the maximum value from each doubleword (considering them unsigned) by calling a procedure from module b.asm. This procedure should also compute the sum of these bytes. Next, in the main module (a.asm) print this string of bytes on the screen (unsigned) and also print the sum of these bytes (signed). Explain the algorithm, justify and comment accordingly the source code and the involved mechanisms. When explaining, focus on problematic and difficult aspects from the given solution. Solve the problem using 2 separate object modules (separate compilation).**

**3421**

Ex: If the string of doublewords is:

**sir dd 1234A678h, 12345678h, 1AC3B47D, FEDC9876h the bytes of max value are respectively Abh. 78h, C3h and FEh, the corresponding string of bytes ranks being and the sum of these bytes being -33.**

**Working time: 2h 30'**

**Evaluation: 1-2.5p; II - 1.5p; III -2p; IV-3p; (+1p - by default)**