08.02.2018

Subject I

**Define 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, explain where are they used.**

Subject II

a) The minimum # of bits

i. 61 = 0011 1101 = 11 1101 = 3Dh => 6 bits

ii. -62 = 1100 0010 = C2h => 8 bits

62 = 0011 1110 = 3Eh

iii. 130 = 1000 0010 = 82h => 8 bits

iv. -129 = 0111 1111 = 111 1111 = 7Fh => 7 bits

129 = 1000 0001 = 81h

b)

xor ah, ah ah = 0

cwde ax -> eax, the high word of eax = the sign bit of ax = ah = 0 (eax = 00 00 00 al)

add ebx, eax

mov al, [ebx]

**mov al, [ebx + al]**

Subject III

Overflow

a)

mov ax, 1000h AX = 1000h

mov bl, 1000b+10b BL = 8+2 = 10 = 0Ah

div bl AH = 6 = 06h AL = 409 = 199h => AL = 99h (doesn’t really do this instruction)

**integer overflow**

b)

mov ah, 0bch AH = BCh

mov al, 0deh AL = DEh

add ah, al AH = 11\*16 + 12 + 13\*16 + 14 = 410 = 19Ah => AH = 9Ah **overflow (CF = 1)**

c)

mov ax, 1001h AX = 1001h AL = 01h

mov bx, 1111b BX = 000Fh BL = 0Fh

imul bl AX = AL\*BL = 1\*15 = 15 = 000Fh

**no overflow**

d)

mov dh, 62h DH = 62h

mov ch, 200 CH = C8h

sub dh, ch DH = 98 – 200 = -102 = 9Ah

**overflow (CF = OF = 1)**