

Middleman 1

Name: Masruro Ahmed Saito id vom 1

Id : 1520469642 id vom 2

works entwickele ab und hat ent

zu abo Am. to the Que No. 1

LEA: LEA is an instruction that loads "offset variable" while adjusting the address between 16 and 32 bits as necessary. "LEA (16 bit reg), (32 bit address)" loads the lower 16 bits of the address into the register, and "LEA (32 bit reg), (16 bit address)" loads the 32 bit register with the address zero extended to 32 bits.

Offset: Offset is an assembler directive in x86 assembly language. It actually means "address" and is a way of handling the overloading of the "mov" instruction.

- 1) mov si, offset variable
2) mov si, variable

The first line loads SI with the address of variable. The second line loads SI with the ~~value stored at the address of variable~~.

(for -id si) A71". processor as did we bne al
slid al repeat addl 16h (marginally 4 id ss)
-id sc) A71" bne Arm to the Gun No. 2
register -id ss int abas "(register -id si) (for
~~DATA~~ Data segment is the starting point of the Data segment in a Program and ~~DATA~~

~~DATA~~ DATA is the name given to this segment and SEGMENT is the keyword for defining segments, where we can declare our variables. On the other hand, DATA is a library function.

Ans. to the Ques No. 3

Given, ~~Now answer about below~~
DS : DATA CS : CODE

[8b + I₂ + X₀] [I₂] [I₂ + X₀]

Here code use this code to assign

- The CS for the code and DS [I₃ + 98]

assigned for the data. [X₀] ... [I₄ + 98]

16 and 32 bits are memory. 1.1A (16 bits) p

[21b + I₂] [21b + I₂ + X₀] [8b + I₂]

d [21b + I₂] [21b + I₂ + X₀] [8b + I₂]

b [21b + 98] [21b + I₂ + 98] [8b + 98]

c [21b + X₀] [21b + I₂ + 98] [8b + X₀]

Ques. Ques. What will be the effect on 16
bit language if we apply more address
and memory by doubling the existing
16 bit memory?

Ans. 1. New 32 bit address

Ans. to the Que No. 4

To access memory, we use : BX, SI, DI, BP.

$[BX + SI]$ $[SI]$ $[BX + SI + d8]$

$[BX + DI]$ $[DI]$ $[BX + DI + d8]$

$[BP + SI]$ $d16$ $[BP + SI + d8]$

$[BP + DI]$ $[BX]$ $[BP + DI + d8]$

$[SI + d8]$ $[BX + SI + d16]$ $[SI + d16]$

$[DI + d8]$ $[BX + DI + d16]$ $[DI + d16]$

$[BP + d8]$ $[BP + SI + d16]$ $[BP + d16]$

$[BX + d8]$ $[BP + DI + d16]$ $[BX + d16]$