LESSON 4



ASM

mov & Addressing

Learn it all @MyWebUniversity



A FREE MYWEBUNIVERSITY COURSE

ASSEMBLY LANGUAGE

LESSON 4

Data, mov and Addressing Modes

db, dw, dd, dq, resb, mov, size specifiers, addressing modes and lea

Defining Data

data.asm

```
section .data
    age     db  21
    year    dw  2026
    count   dd  100000
    big     dq  9000000000
    scores  dd  90, 85, 77, 92        ; array of 4 dwords
    name    db  "Mariam", 0            ; C-style string

section .bss
    buffer  resb 64                    ; 64 empty bytes
```

- db defines bytes, dw words, dd double words and dq quad words. A list separated by commas ...
- In the bss section, resb, resw, resd and resq reserve empty space, for example a buffer for ...

01

mov

mov.asm

```
mov rax, 42             ; immediate → register
mov rbx, rax            ; register → register
mov eax, [count]        ; memory → register
mov [count], eax        ; register → memory
; mov [a], [b]          ; ✗ not allowed
mov ecx, [a]
mov [b], ecx            ; ✓ two steps
```

- mov destination, source copies a value. The source can be a number, a register or memory; ...
- You cannot move memory to memory in one instruction. Load into a register first.

02

Size Specifiers

ASSEMBLY LANGUAGE · 4

sizes.asm

```
mov byte [age], 22              ; 1 byte
mov dword [count], 0            ; 4 bytes
; mov [count], 0                ; ✗ error: operation size not specified

movzx eax, byte [age]           ; zero-extend 8 → 32 bits
movsx rax, dword [temp]         ; sign-extend 32 → 64 bits
```

- When NASM cannot tell the size from a register, you must say it: byte, word, dword or qword …
- movzx and movsx load a small value into a bigger register, filling with zeros or with the …

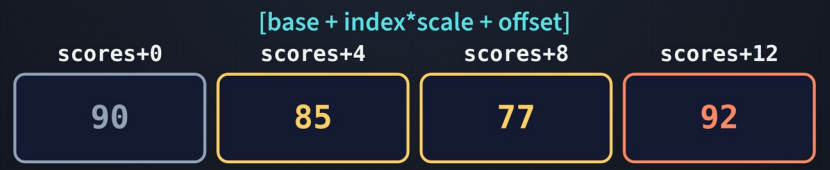
03

Addressing Modes

ASSEMBLY LANGUAGE · 4

hello.asm

```
mov rcx, 2
mov eax, [scores + rcx*4]   ; scores[2]
```



- Inside square brackets, x86 can combine a base register, an index register times one, two, ...
- This is exactly how array indexing works: base address plus index times element size.

04

lea: Load Effective Address



- lea computes the address inside the brackets but does not read memory. It puts the address …
- Compilers also use lea as a fast calculator: lea rax, rbx plus rbx times four computes rbx …

05

Lesson Map


1
Defining Data
2
mov
3
Size Specifiers
4
Addressing Modes
5
lea: Load Effective Address

Sources and Further Reading





Thank you for watching

MyWebUniversity.com

Free technical education since 2000 — English and Dari

@MyWebUniversity · @MyWebUniversity-Dari