Module 1: 32--Bit ASM on Linux

Data Types

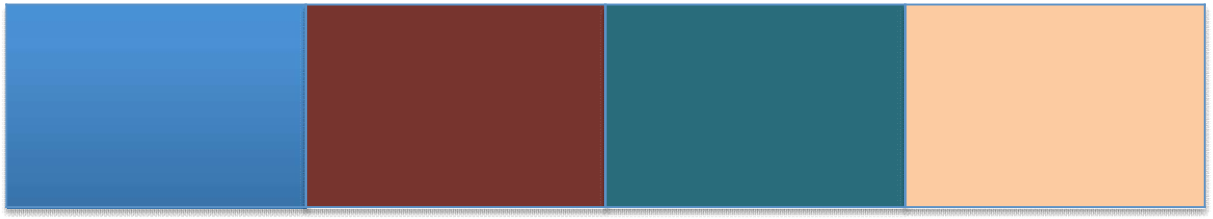
Fundamental Data Types

- Byte – 8 bits
- Word – 16 bits
- Double Word – 32 bits
- Quad Word – 64 bits
- Double Quad Word – 128 bits

Source: IA-32 Manual

Signed and Unsigned

Unsigned Double Word



31 0

Signed Double Word

31 30 0

Source: IA-32 Manual

- Case Sensitive syntax

- Accessing memory reference with []
  - message db 0xAA, 0xBB, 0xCC, 0xDD
  - mov eax, message ←← moves address into eax
  - move eax, [message] ←← moves value into eax

Defining Initialized Data in NASM

```
db      0x55                ; just the byte 0x55
db      0x55,0x56,0x57      ; three bytes in succession
db      'a',0x55            ; character constants are OK
db      'hello',13,10,'$'   ; so are string constants
dw      0x1234              ; 0x34 0x12
dw      'a'                 ; 0x61 0x00 (it's just a number)
dw      'ab'                ; 0x61 0x62 (character constant)
dw      'abc'               ; 0x61 0x62 0x63 0x00 (string)
dd      0x12345678          ; 0x78 0x56 0x34 0x12
dd      1.234567e20         ; floating-point constant
dq      0x123456789abcdef0  ; eight byte constant
dq      1.234567e20         ; double-precision float
dt      1.234567e20         ; extended-precision float
```

Declare Uninitialized Data


```
buffer:         resb    64              ; reserve 64 bytes
wordvar:        resw    1               ; reserve a word
```

Special Tokens

- $ -- evaluates to the current line
- $$ -- evaluates to the beginning of current section

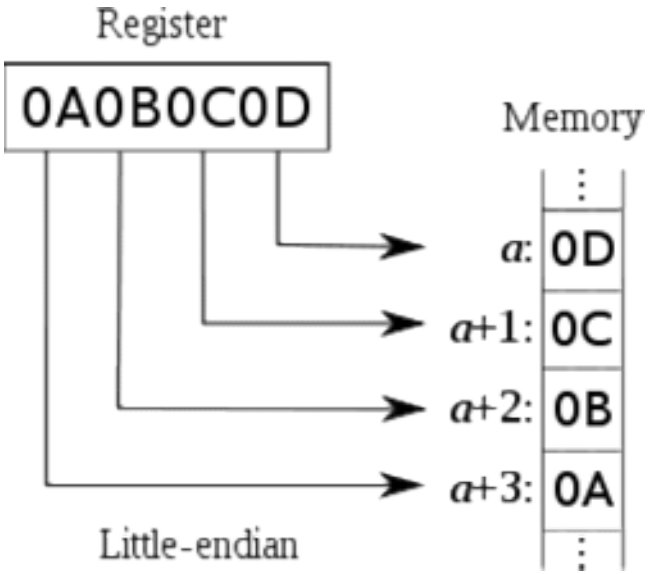
EQU and TIMES


```
message         db      'hello, world'
msglen          equ     $-message

        zerobuf:        times 64 db 0

        times 100 movsb
```

IA-32 uses Little Endian format



http://en.wikipedia.org/wiki/Endianness