Reverse Engineering and Malware Analysis

X86 ARCHITECTURE & ASSEMBLY

Agenda

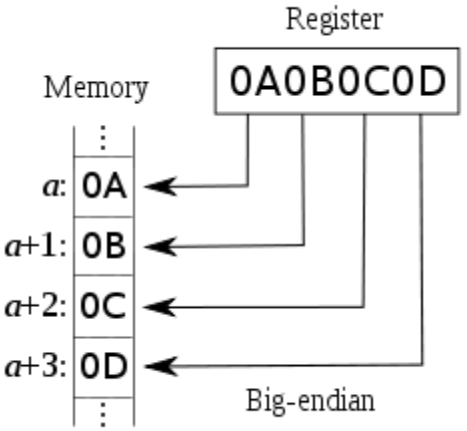
- x86 Architecture
  - Registers
  - Stack
  - Memory and Addressing
  - Memory Management
  - CPU Privilege Modes
- x86 Assembly
  - The Hello World
  - Basic x86 Programming

x86 Architecture

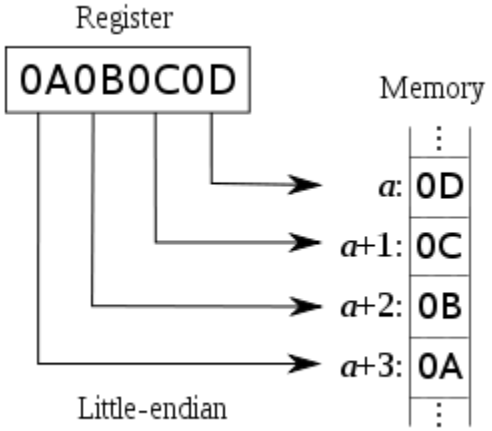
- Originally designed by Intel during 1978
- **CISC**: Complex Instruction Set Computer
  - Varying Length
  - Multi-Data Operation
- Little Endian
- Multi Tasking
- Protected Memory Model
  - Paging and Segmentation
- 16/32/64 bit Platform

Endianness

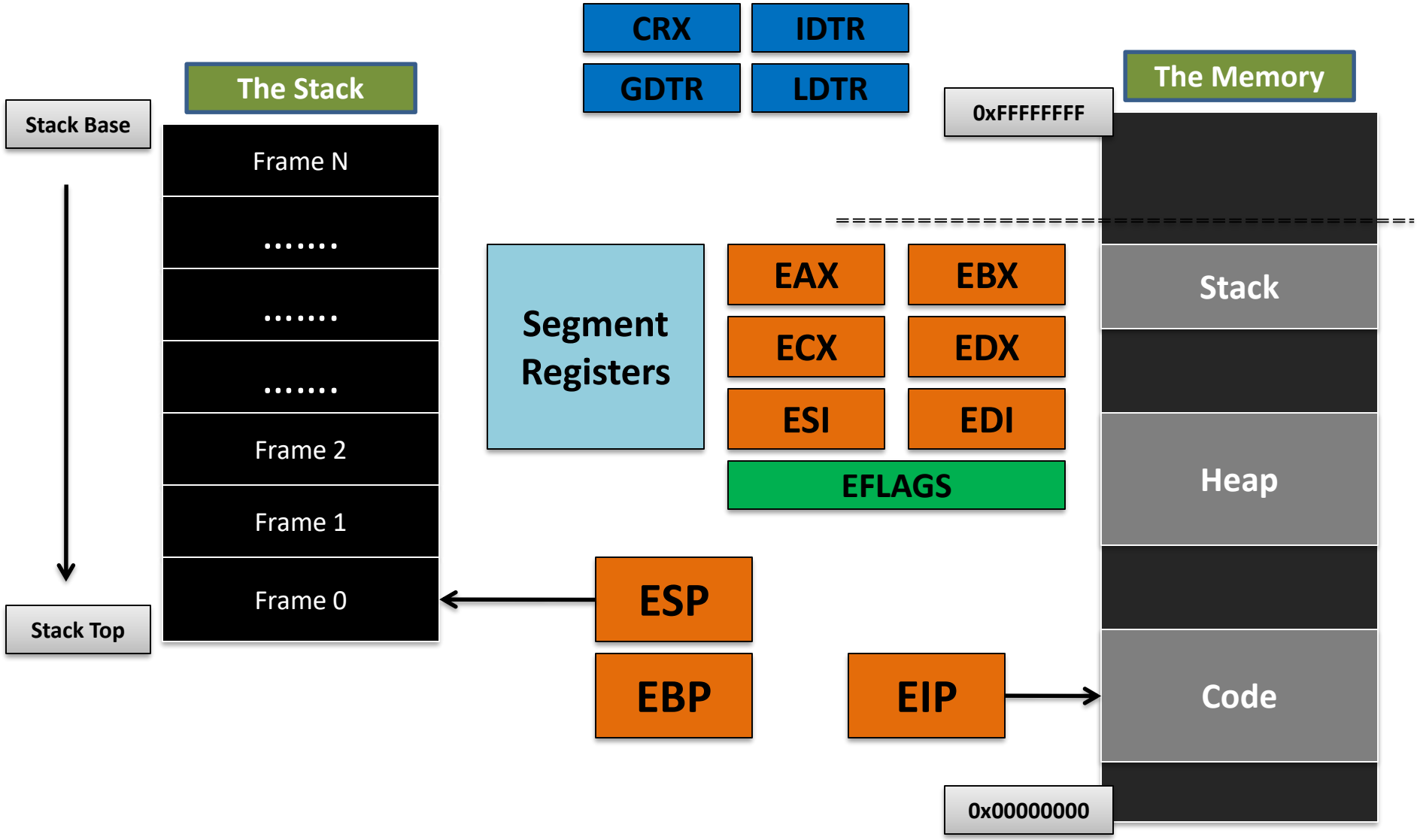
- **Big-Endian**
  - Most Significant Byte (MSB) First



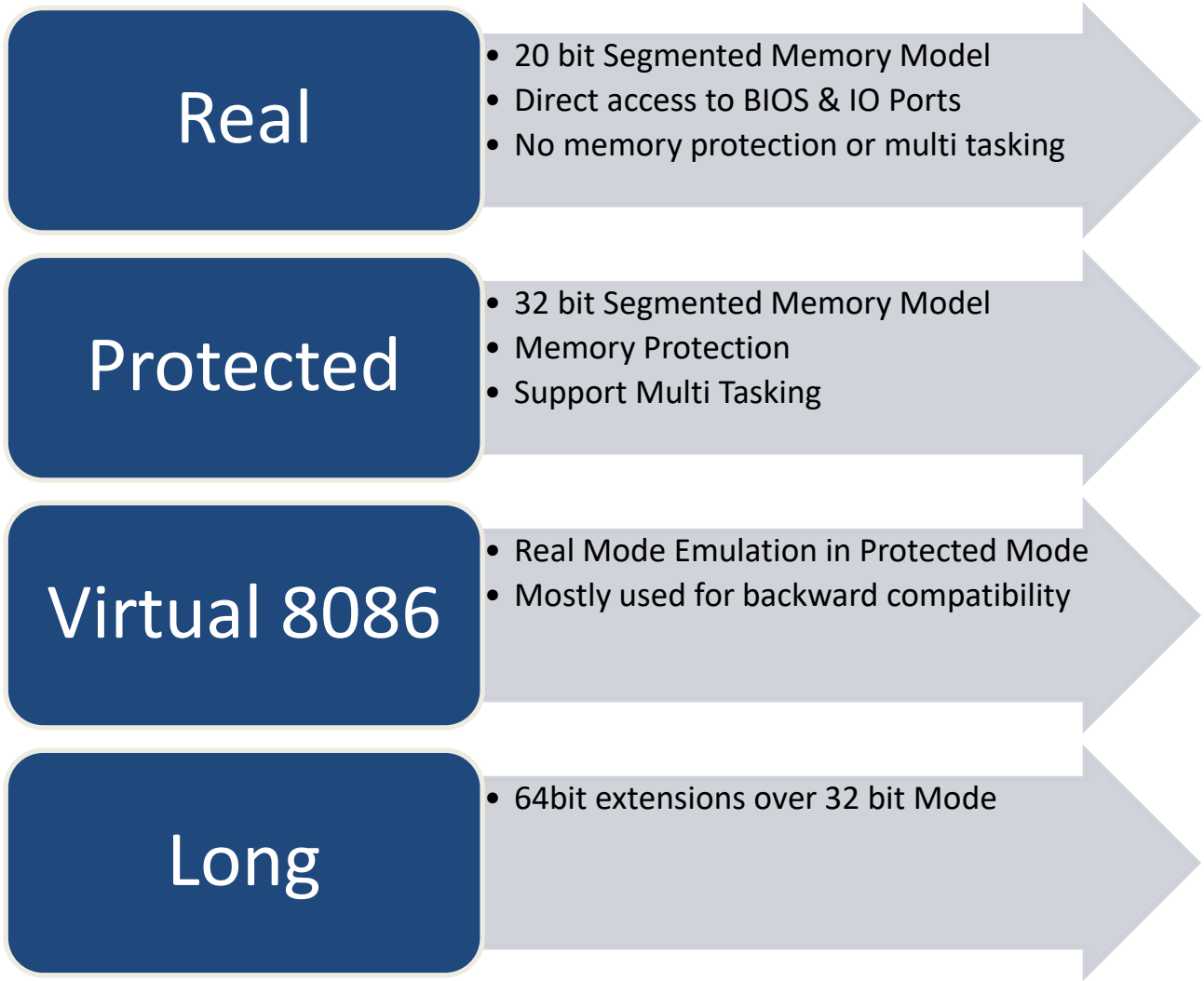
- **Little-Endian**
  - Least Significant Byte (LSB) First



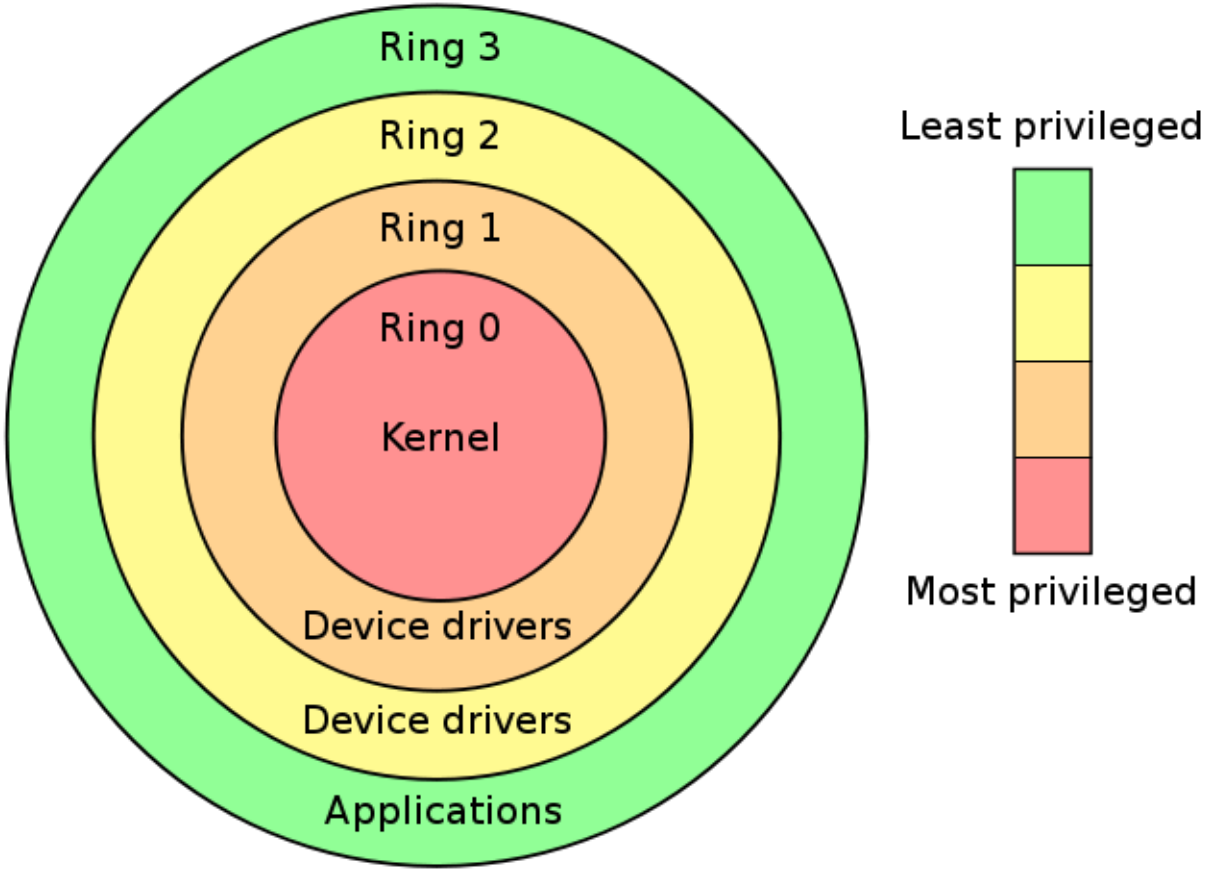
x86 Platform Components



x86 CPU Modes



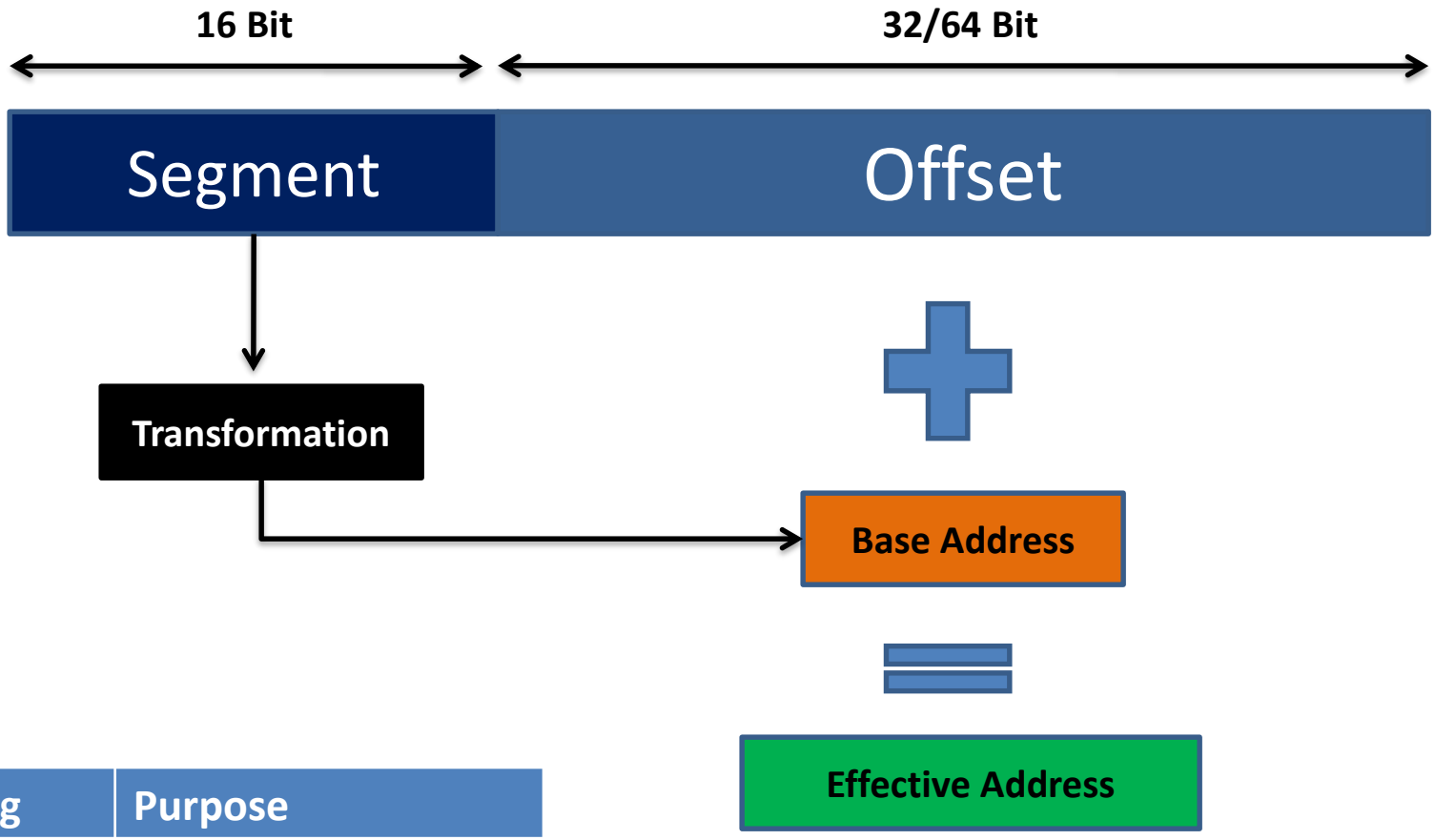
x86 CPU Privilege/Protection Levels



http://duartes.org/gustavo/blog/post/cpu-rings-privilege-and-protection
http://en.wikipedia.org/wiki/Ring_(computer_security)

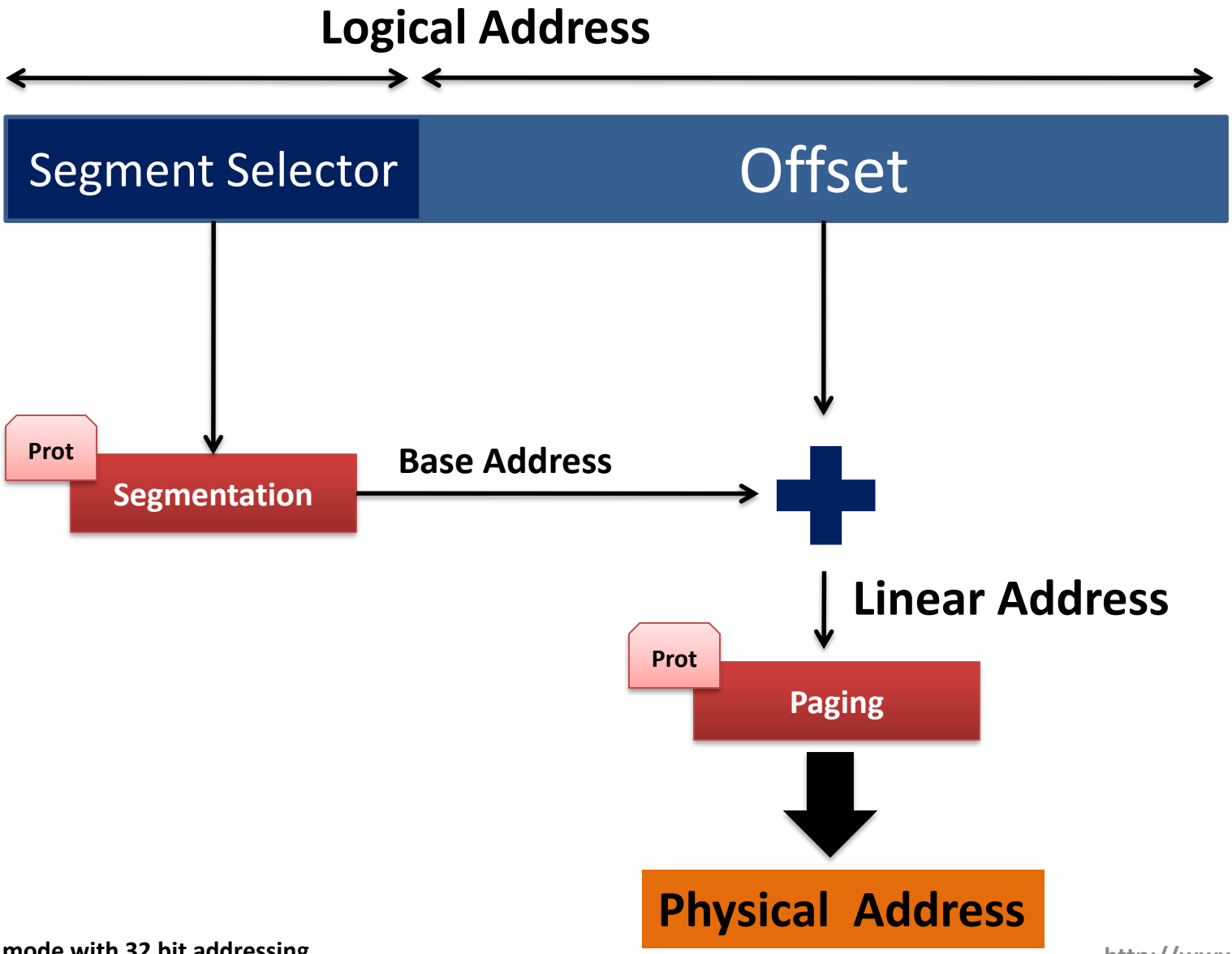
Intel Software Developers Manual

x86 Memory Addressing



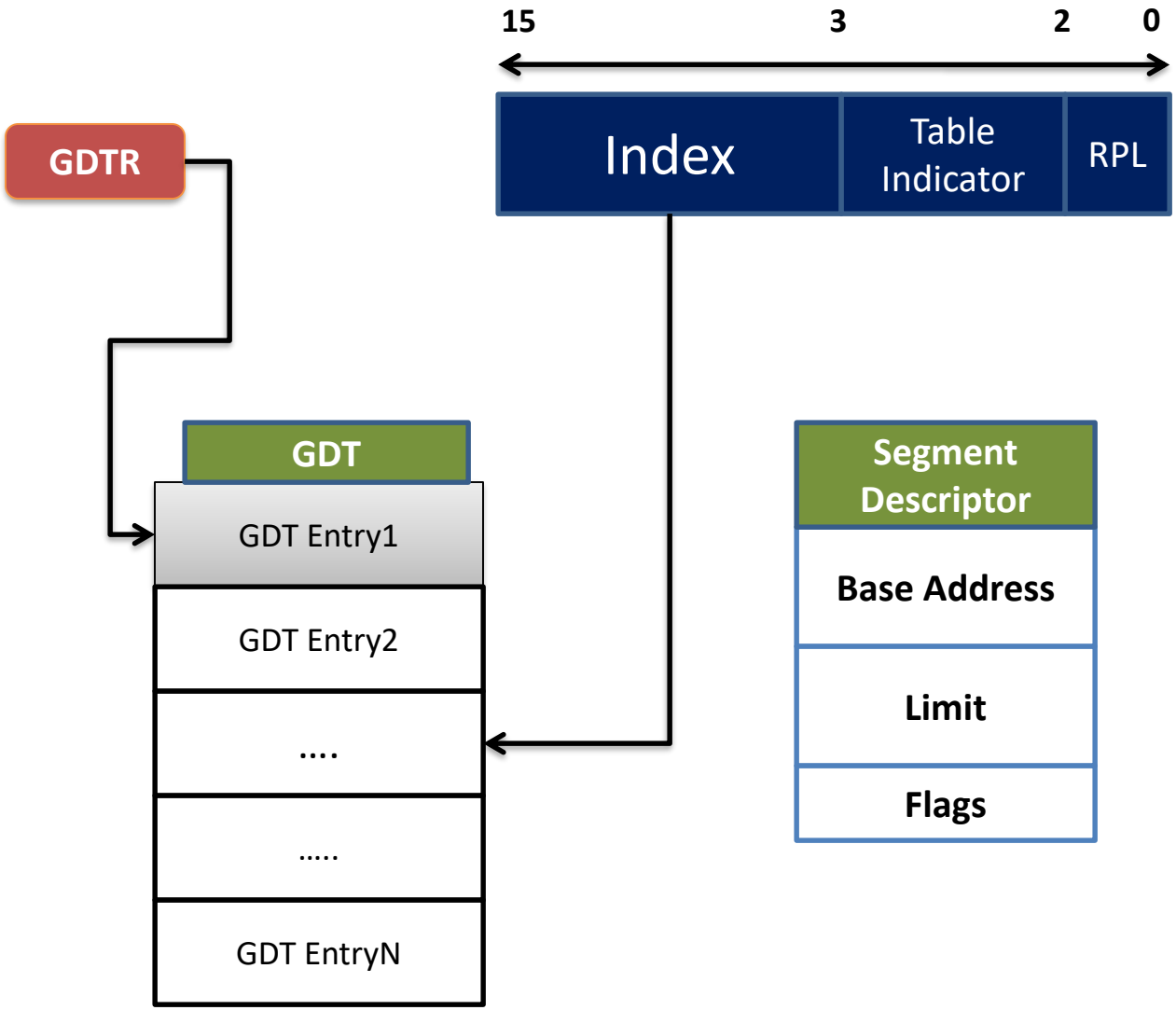
Reg	Purpose
CS	Code Segment
DS	Data Segment
SS	Stack Segment

x86 Memory Management

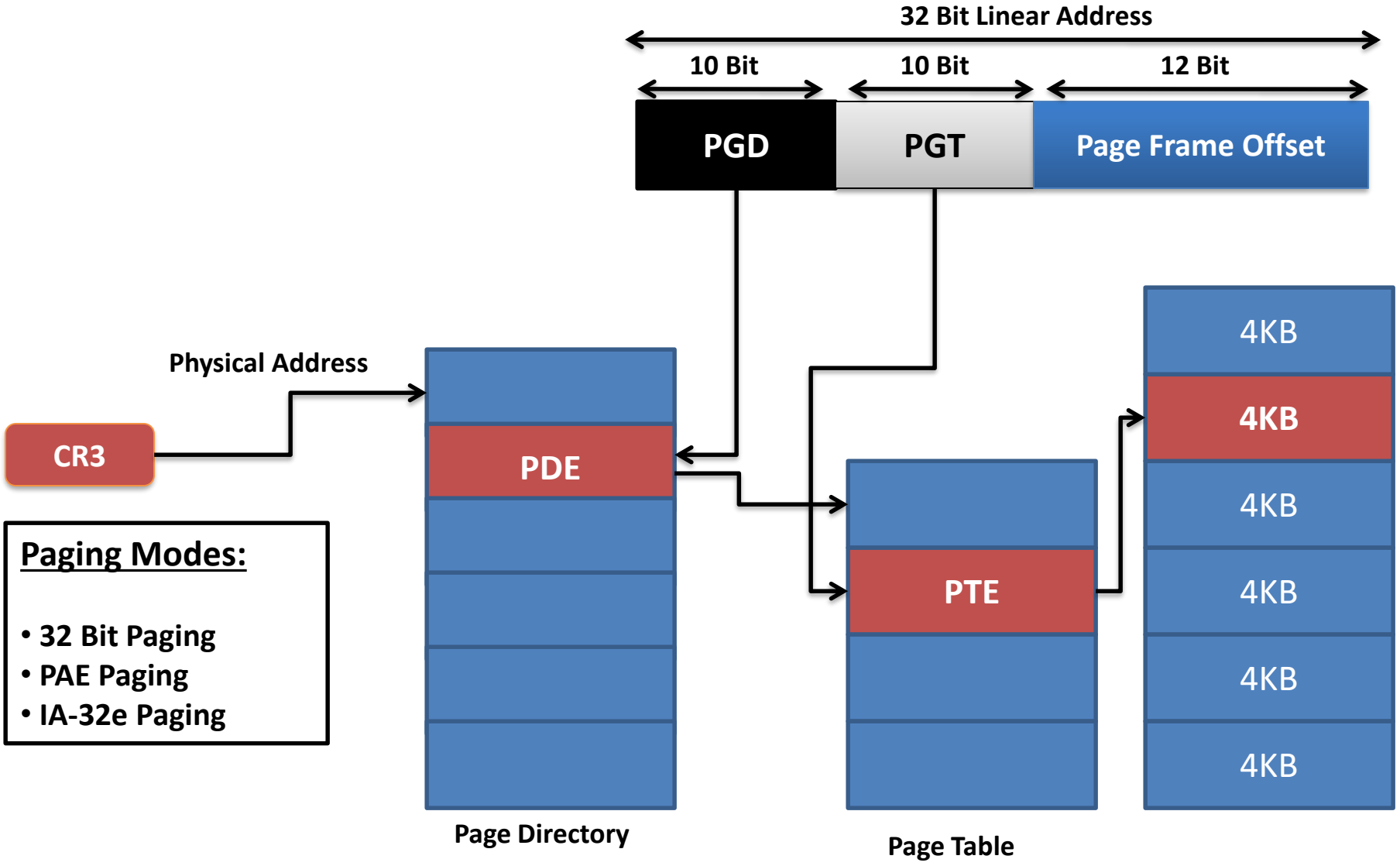


Protected mode with 32 bit addressing

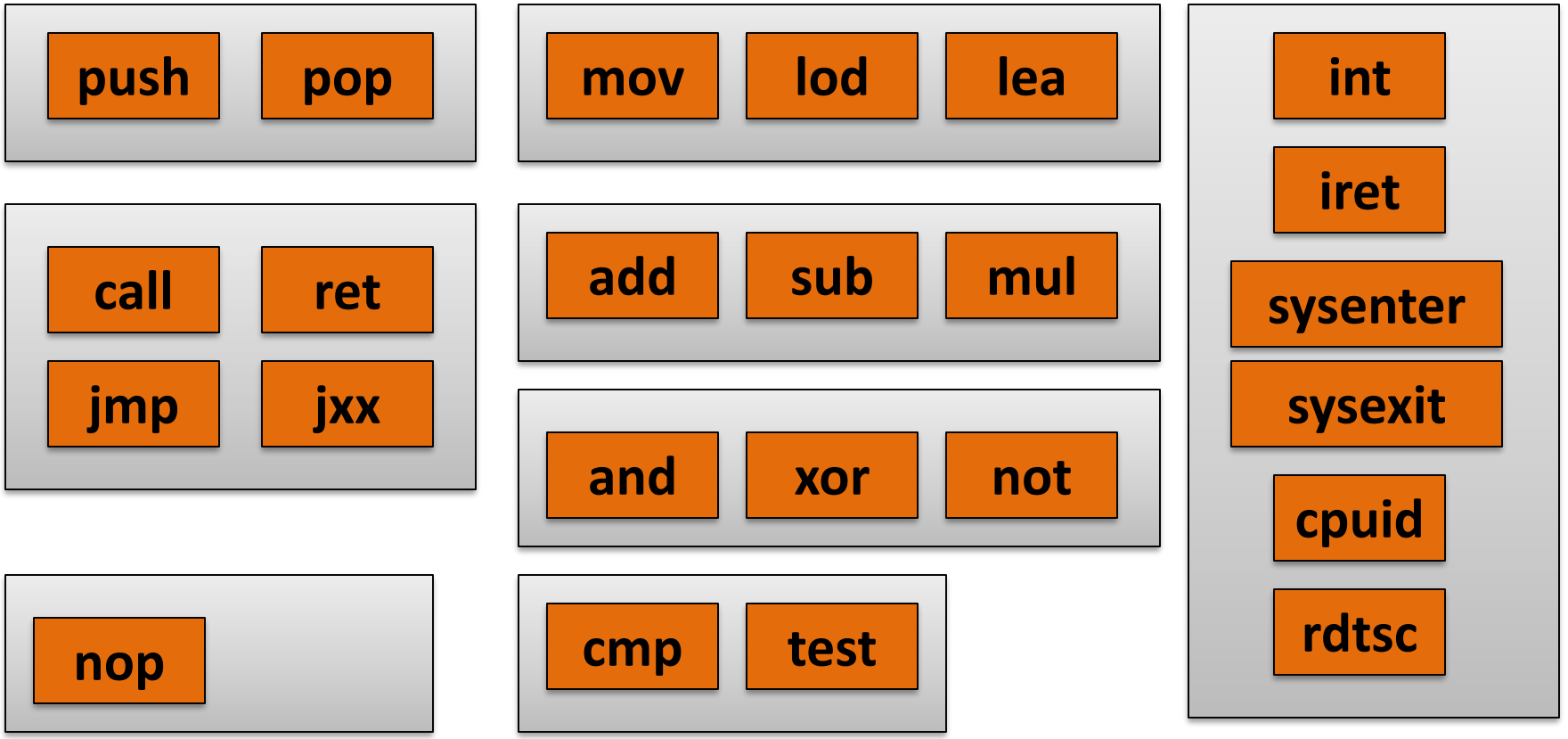
x86 MM: Segmentation



x86 MM: Paging



x86 Assembly: Common Instructions



http://en.wikipedia.org/wiki/X86_instruction_listings

x86 ASM Syntax

- **AT&T**
  - Operand Order: Source, Destination
  - Instructions are suffixed to indicate operand size
  - Used by default in Unix based systems
- **Intel**
  - Operand Order: Destination, Source
  - Register names indicate operand size
  - Used by default in MS-DOS, Windows systems.

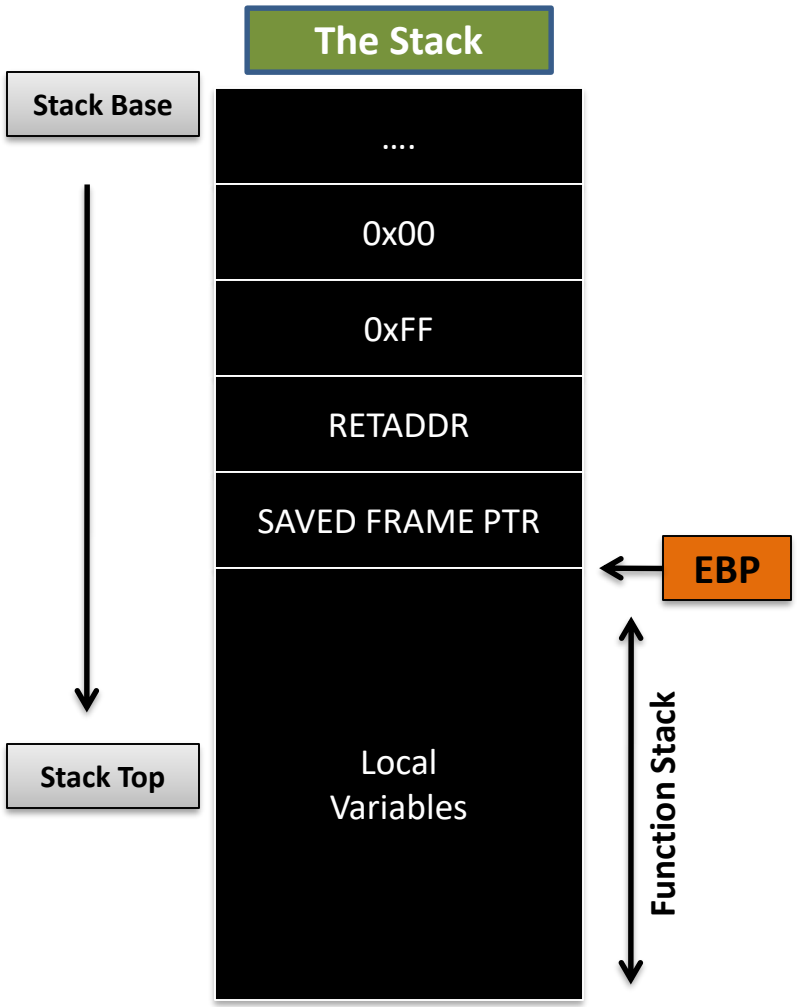
x86 ASM Hello World

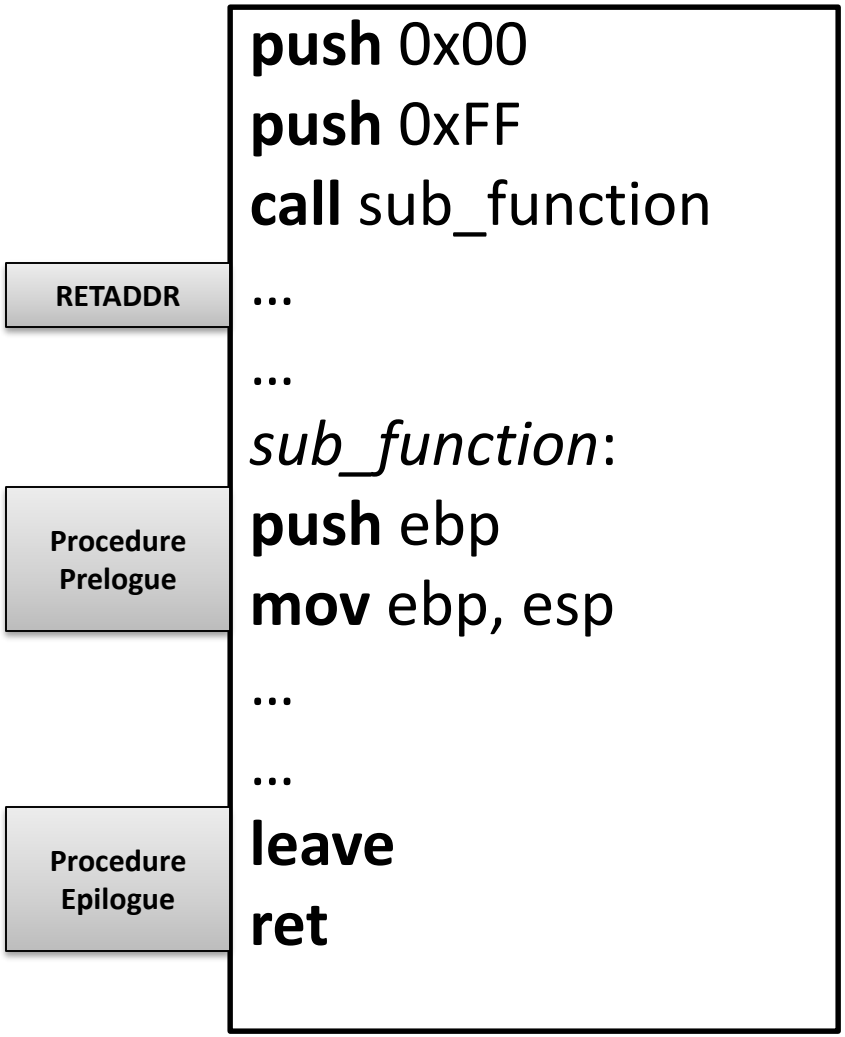
```
1  BITS 32
2
3  GLOBAL _start
4
5  _start:
6
7     call getString
8     db "Hello World",0x0a,0x00
9
10 getString:
11    pop ecx
12    mov edx, 12
13    mov ebx, 1
14    mov eax, 4
15    int 0x80
16
17    xor eax,eax
18    xor ebx, ebx
19    inc eax
20    int 0x80
```

Intel Syntax

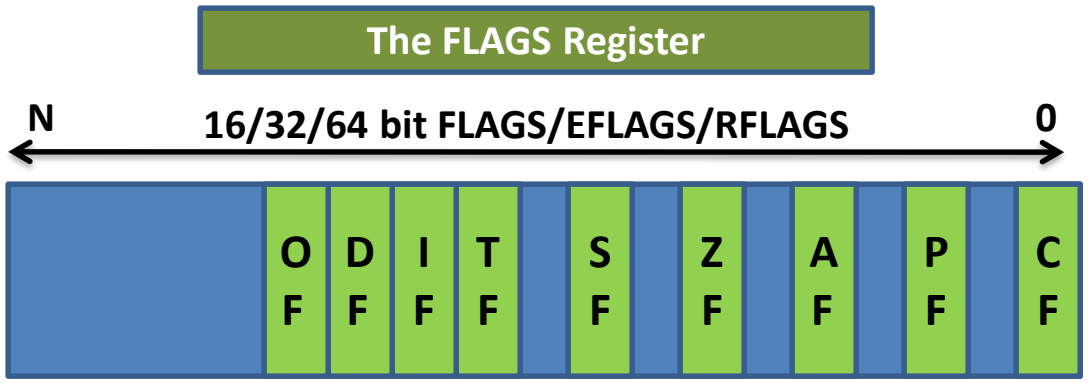
```
nasm –f elf –o hello hello.asm
ld –o hello hello.o
./hello
```

x86 ASM: Call Stack





x86 ASM: Conditions



CMP a, b := a – b

Set **ZF** if a = b

Set **SF** if b > a

…

CMP performs a subtraction and updates FLAGS register

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

mov eax, 1

mov ebx, 2

cmp eax, ebx

jz branch1

jnz branch1

jg branch1

branch1:

…

Serious Reading

- Plex86
  - x86 Virtual Machine
  - http://www.plex86.org/
- OSDever
  - http://www.osdever.net
- JamesM's Kernel Development Tut
  - http://www.jamesmolloy.co.uk/tutorial_html/
- Intel Architecture Software Developers Manual
  - Google…