

Introduction

x86-64

In 64 bit processor, **address size is 64 bit**

Sizes

Byte	8 bits
Word	16 bits
Double Word (DWORD) a.k.a Long Word	32 bits
Quadruple Word (QWORD)	64 bits

Endianness (Order of storing values)

For the Value: 0x01234567

	0x100	0x101	0x102	0x103
Big Endian		01	23	45 67
Little Endian	67	45	23	01

Big Endian: The most significant value in the sequence is stored first, at the lowest storage address.

Little Endian: The least significant value in the sequence is stored first, at the lowest storage address.

In x86-64, **Little Endian is used.**

Virtual Memory Layout

General Purpose Registers

63	31	15	8	7	0	
%rax	%eax	%ax	%ah	%al		Return value
%rbx	%ebx	%bx	%bh	%bl		Callee saved
%rcx	%ecx	%cx	%ch	%cl		4th argument
%rdx	%edx	%dx	%dh	%dl		3rd argument
%rsi	%esi	%si			%sil	2nd argument
%rdi	%edi	%di			%dil	1st argument
%rbp	%ebp	%bp			%bpl	Callee saved
%rsp	%esp	%sp			%spl	Stack pointer
%r8	%r8d	%r8w			%r8b	5th argument
%r9	%r9d	%r9w			%r9b	6th argument
%r10	%r10d	%r10w			%r10b	Caller saved
%r11	%r11d	%r11w			%r11b	Caller saved
%r12	%r12d	%r12w			%r12b	Callee saved
%r13	%r13d	%r13w			%r13b	Callee saved
%r14	%r14d	%r14w			%r14b	Callee saved
%r15	%r15d	%r15w			%r15b	Callee saved

Special Registers

RIP (Instruction Pointer)	Pointer to the next instruction to run
RFLAGS	Saves the current state of the processor.

Both registers are 64 bits registers.

Assembly File Structure

```
.global _start # entry point

.section .data
index: .int 1

.section .bss
.lcomm final, 4

.section .text
_start:
    movl (index), %eax
    addl $5, %eax
    movl %eax, (final)
```

The assembly file is **divided into sections** using "Section Directives".

Moreover, the file can contain Global and External declarations, Directives, Labels, Comments, and of course, the assembly Instructions.

