Basics of x86 assembly

What is x86 Architecture

- The x86 architecture is an instruction set architecture (ISA) series for computer processors.
- Developed by Intel Corporation, x86 architecture defines how a processor handles and executes different instructions passed from the operating system (OS) and software programs.
- The “x” in x86 denotes ISA version.

Registers

- A special high speed area in the processor
- General purpose registers
  - 16 bit: ax, bx, cx, dx
  - 32 bit: eax, ebx, ecx, edx
  - 64 bit: rax, rbx, rcx, rdx
- Pointer registers
  - *SP, *BP, *IP
- Index registers
  - *SI, *DI
- Flags

Instructions

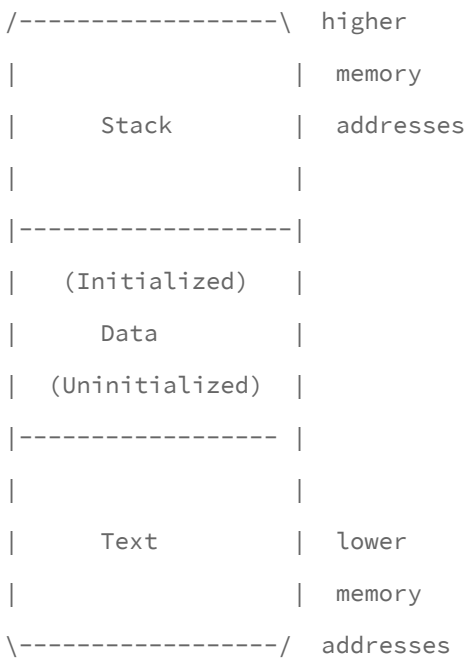
AT&T Syntax(src, dest)

- push/pop %rbp
- mov $0xa, %edi
- add/sub $0ax, %rax
- call 0x680
- jle/jne/jmp/je 0x7fa
- lea 0xe4(%rip), %rdi
- ret

Intel (dest, src)

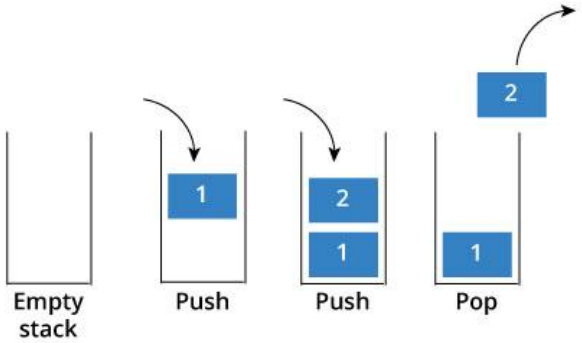
- push/pop rbp
- mov edi, 0xa
- add/sub rax, 0xa
- call 0x680
- jle/jne/jmp/je 0x7fa
- lea rdi, [rip +0xe3]
- ret

Process memory region



Stack

- An area memory that’s used during program execution
- Follows the “Stack” data structure.
- Last in first out (LIFO)



Stack pointers

- Stack pointer (SP)
- Frame pointer / local base pointer (FP/LBS)

How functions look in the stack

- Sample program

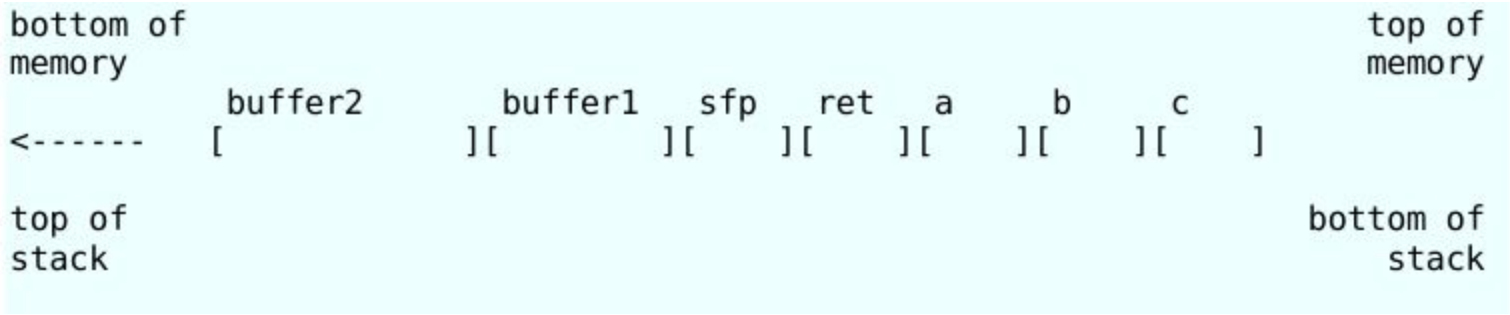
```
void function(int a, int b, int c) {
   char buffer1[5];
   char buffer2[10];
}

void main() {
  function(1,2,3);
}
```

-Disassembly of main function

```
pushl $3
pushl $2
pushl $1
call function
```

- Disassembly of function

```
pushl %ebp
movl %esp,%ebp
subl $20,%esp
```


bottom of memory
top of memory
buffer2
buffer1
sfp
ret
a
b
c
<------
top of stack
bottom of stack

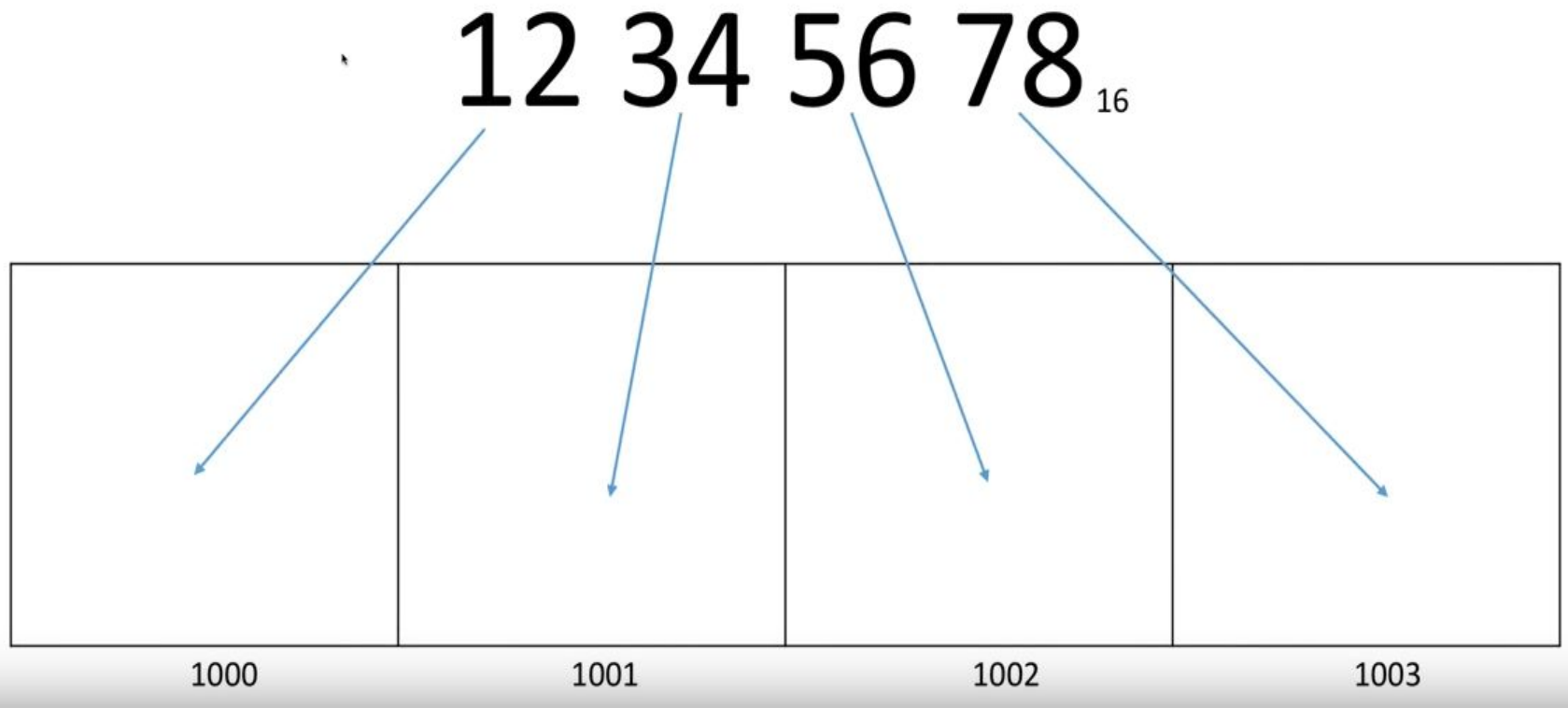
Endianness

Endianness refers to the order of bytes within a binary representation of a number.

Two types that we will discuss:

- Big endian
- Little endian

Big Endian



Little Endian