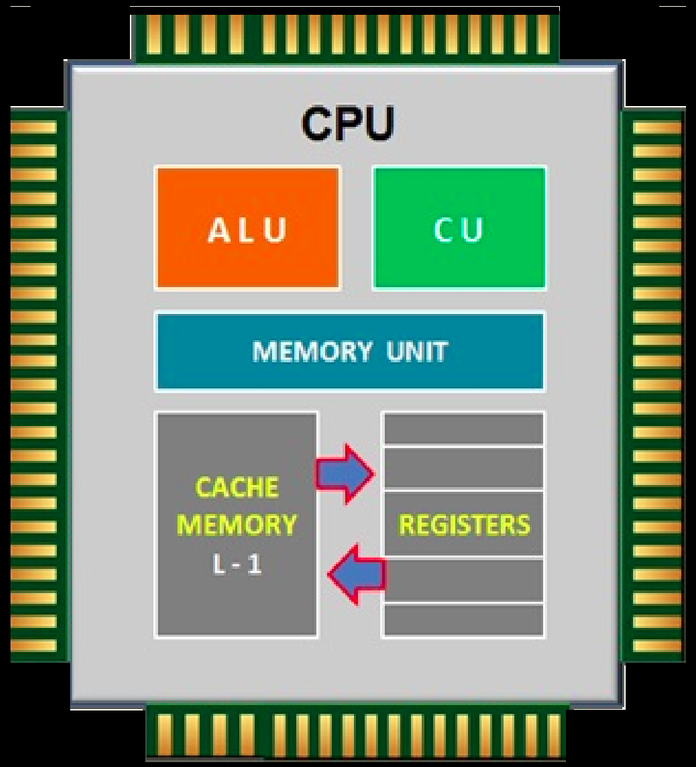
Pwn - Assembly Language

Registers

Internal Memory Locations to Store Data



64-bit register	Lower 32 bits	Lower 16 bits	Lower 8 bits
rax	eax	ax	al
rbx	ebx	bx	bl
rcx	ecx	cx	cl
rdx	edx	dx	dl
rsi	esi	si	sil
rdi	edi	di	dil
rbp	ebp	bp	bpl
rsp	esp	sp	spl
r8	r8d	r8w	r8b
r9	r9d	r9w	r9b
r10	r10d	r10w	r10b
r11	r11d	r11w	r11b
r12	r12d	r12w	r12b
r13	r13d	r13w	r13b
r14	r14d	r14w	r14b
r15	r15d	r15w	r15b

Registers come in all sizes !

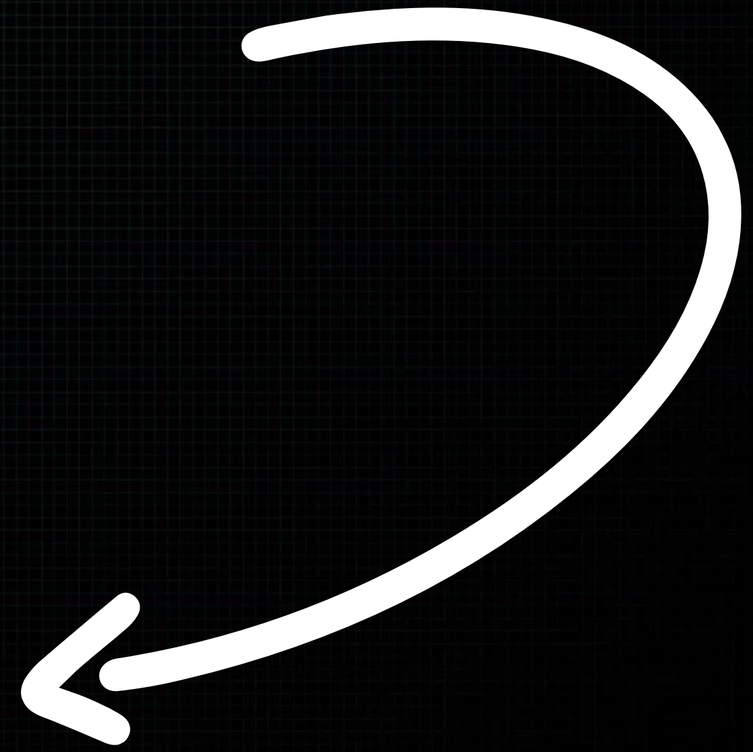
XL **L** **M** **S**

Registers


Reserved
x86
x64
EBP
RBP
EIP
RIP
ESP
RSP
General-Purpose
x86
x64
EAX
RAX
EBX
RBX
ECX
RCX

Disassembly

```
int ret_after_add_leetcode(int arg)
{
    return arg + 0x1337c0de;
}
```



```
gef➤  disass ret_after_add_leetcode
Dump of assembler code for function ret_after_add_leetcode:
   0x0000000000001149 <+0>:     endbr64
   0x000000000000114d <+4>:     push   rbp
   0x000000000000114e <+5>:     mov    rbp,rsp
   0x0000000000001151 <+8>:     mov    DWORD PTR [rbp-0x4],edi
   0x0000000000001154 <+11>:    mov    eax,DWORD PTR [rbp-0x4]
   0x0000000000001157 <+14>:    add    eax,0x1337c0de
   0x000000000000115c <+19>:    pop    rbp
   0x000000000000115d <+20>:    ret
End of assembler dump.
```

Instructions

<Instruction>

<Instruction> <Operand>

<Instruction> <Destination> <Source>

Arithmetic Instructions

Add - Adds destination from Source

```
add rax,0xdeadc0de
add rax,rdx
```

Sub - Subtracts destination from Source

```
sub rbx,0xcafed00d
sub rbx,rdx
```

Inc - Increment operand by 1

```
inc rcx
```

Dec-Decrement operand by 1

```
dec rcx
```

Transfer Instructions

Mov - copies destination to source

mov rax, 0xdecafbad

mov rax, rdx

mov rax, [rdx]

Push - Push onto Stack

push rbp

Pop - Pop from stack

pop rbp

Jump Instructions

Call - Call for a subroutine

```
call 0x1050 <printf@plt>
```

Jmp - Jump to Destination

```
jmp 0x11a4 <main+27>
```

Ret- Return from subroutine

```
ret
```

Assembly Language

```
gef➤  disass ret_after_add_leetcode
Dump of assembler code for function ret_after_add_leetcode:
   0x0000000000001149 <+0>:     endbr64
   0x000000000000114d <+4>:     push   rbp
   0x000000000000114e <+5>:     mov    rbp,rsp
   0x0000000000001151 <+8>:     mov    DWORD PTR [rbp-0x4],edi
   0x0000000000001154 <+11>:    mov    eax,DWORD PTR [rbp-0x4]
   0x0000000000001157 <+14>:    add    eax,0x1337c0de
   0x000000000000115c <+19>:    pop    rbp
   0x000000000000115d <+20>:    ret
End of assembler dump.
```

Function Prologue

Add Instruction

Function Epilogue


THANK YOU FOR WATCHING