Machine Code Explore

```
as -al MachineCode.s
```

```
    50                  push %eax
    51                  push %ecx
    52                  push %edx
    53                  push %ebx
    54                  push %esp
    55                  push %ebp
    56                  push %esi
    57                  push %edi
    58                  pop  %eax
    89 c0               mov  %eax, %eax
    89 c2               mov  %eax, %edx
    89 d0               mov  %edx, %eax
    89 d2               mov  %edx, %edx
    89 00               mov  %eax, (%eax)
    89 40 04            mov  %eax, 4(%eax)
    8b 00               mov  (%eax), %eax
    8b 40 04            mov  4(%eax), %eax
    b8 cd ab 34 12      mov  $0x1234abcd, %eax
    ba cd ab 34 12      mov  $0x1234abcd, %edx
    01 c0               add  %eax, %eax
    05 cd ab 34 12      add  $0x1234abcd, %eax
    81 c2 cd ab 34 12   add  $0x1234abcd, %edx
    29 c0               sub  %eax, %eax
    2d cd ab 34 12      sub  $0x1234abcd, %eax
    e8 b9 11 00 00      call printf
    b8 00 00 00 00      mov  $0x0, %eax
    31 c0               xor  %eax, %eax
```

http://git.io/MachineCode

ModRM Byte

m	m	r	r	r	b	b	b
mode		register			r/m field		

mm mode

`00` memory operand; address in register specified by bbb

`01` memory operand; address in register specified by bbb plus 8-bit offset

`10` memory operand; address in register specified by bbb plus 16-bit offset

`11` register operand; register specified by bbb

32-bit Registers

	Register Field			Name
0	0	0	0	eax
1	0	0	1	ecx
2	0	1	0	edx
3	0	1	1	ebx
4	1	0	0	esp*
5	1	0	1	ebp
6	1	1	0	esi
7	1	1	1	edi

* When used in the `r/m field` *and* is being used as a memory address (not mode `11`), this indicates the presense of a **SIB** byte, rather than the `esp` register.