漏洞银行逆向工程系列讲座（一） —— 暴力流学汇编

# 第八课：算术运算指令


## 主讲：K1ght（漏洞银行安全专家）


讲师互动丨课后交流丨资料共享 二进制学习圈：**542285506**

漏洞银行微信公众号：**BUG_BANK**

漏洞银行 | BUGBANK
www.bugbank.cn

# 算术运算指令

80x86指令包括加、减、乘、除四种基本算术运算操作及十进制算术运算调整指令。二进制加、减法指令，带符号操作数采用补码表示时，无符号数和带符号数据运算可以使用相同的指令。二进制乘、除法指令分带符号数和无符号数运算指令

- 加法指令、减法指令、加1减1指令
- 比较指令、交换相加指令、求补指令
- 乘法指令、除法指令

# 一、加法指令

**格式：**ADD　DEST，SRC

ADC　DEST，SRC

**功能：**ADD是将源操作数与目的操作数相加，结果传送到目的操作数。ADC是将源操作数与目的操作数以及CF(低位进位)值相加，结果传送到目的操作数。

源操作数可以是通用寄存器、存储器或立即数。目的操作数可以是通用寄存器或存储器操作数。

ADD，ADC指令影响标志位为OF，SF，ZF，AF，PF，CF。

MOV AX，9876H

ADD AH，AL；AX=0E76H　CF=1　SF=0O　F=0　ZF=0　AF=0　PF=0

ADC AH，AL；AX=8576H　CF=0　SF=1O　F=1　ZF=0　AF=1　PF=0

# 二、减法指令

**格式：**SUB DEST，SRC

SBB DEST，SRC

**功能：**SUB将目的操作数减源操作数，结果送目的操作数。SBB将目的操作数减源操作数，还要减CF(低位借位)值，结果送目的操作数。

源操作数可以是通用寄存器、存储器或立即数。目的操作数可以是通用寄存器或存储器操作数。

SUB，SBB指令影响标志位为OF，SF，ZF，AF，PF，CF。

MOV AX， 9966H；AX=9966H

SUB AL, 80H;AL=E6HCF=1SF=1OF=1ZF=0AF=0PF=0

SBB AH, 80H;AH=18HCF=0SF=0OF=0ZF=0AF=0PF=1

# 三、加1减1指令

**格式：**INC DEST

DEC DEST

**功能：**INC指令将目的操作数加1，结果送目的操作数。DEC指令将目的操作数减1，结果送目的操作数。目的操作数为通用寄存器或存储器操作数。

INC，DEC指令影响标志位为OF，SF，ZF，AF，PF。

INC BL；BL←BL+1

INC AX;AX←AX+1

INC WORDPTR [BX];存储器操作数加1

DEC BYTE PTR [SI];存储器操作数减1

DEC EAX；EAX←EAX-1

# 四、比较指令

**格式：**CMP DEST，SRC

**功能：**目的操作数减源操作数，结果不回送。源操作数为通用寄存器、存储器和立即数。目的操作数为通用寄存器、存储器操作数。

CMP指令影响标志位为OF，SF，ZF，AF，PF，CF。

CMP CX，3

CMP WORD PTR [SI]，3

CMP AX，BLOCK

# 四、比较指令-CMP指令对标志位的影响

执行比较指令后，对状态标志位影响见表。对于两个数的比较(AX-BX)有以下3种情况

两个正数比较，使用SF标志位判断。

SF=0，则AX≥BX，若ZF=1，则AX=BX

SF=1，则AX<BX

两个无符号数比较，使用CF标志位判断。

CF=0，则AX≥BX，若ZF=1，则AX=BX

CF=1，则AX<BX

两个负数比较，使用SF标志位判断。

SF=0，则AX≥BX，若ZF=1，则AX=BX

SF=1，则AX<BX

两个异符号数比较。

如果OF=0，仍可用SF标志判断大小。

如果OF=1，说明结果的符号位发生错误，所以

SF=0，则AX<BX

SF=1，则AX > BX

综上所述：两个异号数比较

当OF=0，SF=0，则AX > BX

SF=1，则AX<BX

当OF=1，SF=0，则AX<BX

SF=1,则AX > BX

用逻辑表达式表示为：

若OF∨-SF=0,则AX > BX

若OF∨-SF=1，则AX<BX

| 数据类型 | 关系 | CF | ZF | SF | OF |
|---|---|---|---|---|---|
| 带符号数 | Dest=Src | 0 | 1 | 0 | 0 |
| | Dest<Src | - | 0 | 1 | 0 |
| | | - | 0 | 0 | 1 |
| | Dest>Src | - | 0 | 0 | 0 |
| | | - | 0 | 1 | 1 |
| 无符号数 | Dest=Src | 0 | 1 | 0 | 0 |
| | Dest<Src | 1 | 0 | - | - |
| | Dest>Src | 0 | 0 | - | - |

# 五、交换相加指令

**格式：**XADD DEST，REG

**功能：**目的操作数加源操作数，结果送目的操作数。原目的操作数内容送源操作数。源操作数允许为通用寄存器。目的操作数允许为通用寄存器、存储器操作数。

XADD指令影响标志位为OF，SF，ZF，AF，PF，CF。

# 六、求补指令

**NEG DEST**

**功能：**对目的操作数求补，用零减去目的操作数，结果送目的操作数。目的操作数为通用寄存器、存储器操作数。

NEG指令影响标志位为OF，SF，ZF，AF，PF，CF。

# 七、乘法指令

**格式：**MUL　SRC

IMUL　SRC

**功能：**MUL为无符号数乘法指令，IMUL为带符号数乘法指令。源操作数为通用寄存器或存储器操作数。目的操作数缺省存放在ACC(AL，AX，EAX)中，乘积存AX，DX：AX，EDX：EAX中。

字节乘：AL SRC→AX

字乘：AX SRC→DX：AX

双字乘：EAX SRC→EDX：EAX

MUL，IMUL指令执行后，CF=OF=0，表示乘积高位无有效数据；CF=OF=1表示乘积高位含有效数据，对其它标志位无定义。

# 七、乘法指令

MUL BL；字节乘

MUL WORD PTR [SI]；字乘

IMUL BYTE PTR [DI]；字节乘

IMUL DWORD PTR [ECX]；双字乘

如果使用IMUL指令，积采用补码形式表示。

# 八、除法指令

**格式：**DIV SRC

IDIV SRC

**功能：**DIV为无符号数除法，IDIV为带符号数除法。源操作数作为除数，为通用寄存器或存储器操作数。被除数缺省在目的操作数AX，DX：AX，EDX：EAX中。

字节除法：AX/SRC商→AL，余数→AH

字除法：DX·AX/SRC商→AX，余数→DX

双字除法：EDX·EAX/SRC商→EAX，余数→EDX

# 八、除法指令

由于被除数必须是除数的双倍字长，一般应使用扩展指令进行高位扩展。当进行无符号数除法时，被除数高位按0扩展为双倍除数字长。当进行有符号数除法时，被除数以补码表示。可使用扩展指令CBW，CWD，CWDE，CDQ进行高位扩展。

例如：

```
MOV AX，BLOCK
CWD；被除数高位扩展
MOV BX，1000H
IDIV BX
```

对于带符号除法，其商和余数均采用补码形式表示，余数与被除数同符号。当除数为零或商超过了规定数据类型所能表示的范围时，将会出现溢出现象，产生一个中断类型码为“0”的中断。执行除法指令后标志位无定义。

加入漏洞银行二进制学习圈　**QQ群号：542285506**

获取**免费课件**｜结交**讲师伙伴**｜紧跟**后续课程**

**微信公众号：BUG_BANK**

漏洞银行 | BUGBANK
www.bugbank.cn