MOLTIPLICAZIONE

Moltiplicazione senza segno

MUL S | D ← D * S

operando destinazione implicito

mull %edx:%eax %eax r32/imm32

Es.:

movl $2, %eax

mull $5

Moltiplicazione con segno

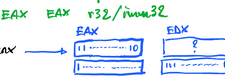
IMUL S | D ← D * S

EDX:EAX EAX r32/imm32

Es.:

movl $-2, %eax

imull $5



DIVISIONE

DIV S | D ← D / S

EDX:EAX

quoziente

idivl r32/imm32 | EAX ← EDX:EAX / imm32

EDX ← EDX:EAX % imm32

modulo/resto

idivw r16/imm16 | AX ← DX:AX / imm16

DX ← DX:AX % imm16

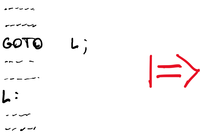
idivb r8/imm8 | AL ← AX / imm8

AH ← AX % imm8

GESTIONE DEL FLUSSO DI ESECUZIONE

SALTO INCONDIZIONATO

In C l'istruzione GOTO permette salti non controllati all'interno del codice sorgente



SALTO CONDIZIONATO

Tutte le istruzioni di salto condizionato in ASM si sviluppano in due fasi:

1. Esecuzione di una istruzione aritmetica/logica
2. Esecuzione di un salto nel codice condizionato in base ad una proprietà del risultato calcolato in 1

PROPRIETÀ DEL RISULTATO => CONDITION CODE (cc)

Es.: R = D - S , R = D + S , R = - D

CONDITION CODE (cc)	PROPRIETÀ DI R	
e / z	R == 0	
ne / nz	R != 0	
g	R > 0	con segno
l	R < 0	con segno
ge	R ≥ 0	con segno
le	R ≤ 0	con segno
a	R > 0	senza segno
b	R < 0	senza segno
ae	R ≥ 0	senza segno
be	R ≤ 0	senza segno

Il salto condizionato è implementato in ASM da

Jcc L

↳ CONDITION CODE

Es.: JZ , JGE , JB

Esempio.:

subl %eax , %ecx → R = ECX - EAX

je L

↳ salta ad L R == 0 <=> EAX == ECX

Esempio.:

subl %eax , %ecx → R = ECX - EAX

jge L

↳ salta se R ≥ 0 <=> ECX ≥ EAX

ECX - EAX ≥ 0

ISTRUZIONE IF IN ASM

CODICE C	CODICE C EQ. 1	CODICE C EQ. 2	CODICE ASM
if (c > a) ? istruz1 : istruz2 istruz 2	r = c - a if (r > 0) istruz1 istruz2	r = c - a if (r ≤ 0) GOTO L istruz 1 L: istruz2	subl %eax, %ecx jle L istruz 1 L: istruz 2

Per non "sporcare" un registro con SUB usiamo CMP

CMP S,D | D - S

Calcola il risultato di D-S ma non lo memorizza in alcun registro

Quasi tutte le istruzioni ASM alterano in diverso modo il registro EFLAGS