

ASM Cheat Sheet (SASM edition)

Istruzioni

Op Code	Scopo	Esempio
MOV	Assegna un valore ad un registro	mov eax, 15
ADD	Somma due valori e il risultato lo pone nel primo dei due registri. Il secondo parametro può essere un registro, un numero hex, dec o bin o un riferimento allo stack e un suo spazio	add eax, ebx add ecx, 23 add eax, [pippo]
SUB	Sottrae con stesse dinamiche della somma	sub eax, ecx sub eax, 1
AND	Operazione di AND logico. Utile per capire se un numero è pari	and eax, 1b
OR	Operazione di OR logico	or eax, ebx
NOT	Operazione di negazione NOT logica	not eax
XOR	Esegue lo XOR logico	xor eax, 101b
INC / DEC	Incrementa o decrementa di 1 come il classico i++	INC AX DEC BX
MUL / DIV	moltiplicazione o divisione con quello che è già in eax, * o / il parametro. Può essere un registro o numero. Porre a zero anche edx dove viene messo il riporto o resto	Es eax=5*2 mov eax, 5 mul 2
JMP	Esegue un salto di codice ad una etichetta assegnata	jmp etichetta

CMP Compara due elementi, il primo deve essere un registro, il secondo un registro o un numero o un riferimento [] cmp eax, ebx
cmp eax, 5
cmp ebx, [pippo]

PUSH Salva un registro nello stack. PUSHA tutti i registri in automatico. Utile quando hai i registri pieni ma devi riusarli per altre operazioni PUSH eax
PUSHA

POP Ripristina i dati nel registro parametro. Occhio, è un FIFO, da riprendere al specchio rispetto al PUSH POP rax
POPA

Salti condizionali

Instruction	Description
JG	Salta se >
JGE	Salta se >=
JL	Salta se <
JLE	Salta se <=
JE	Salta se ==
JZ	Sala se Zero

Registri

x86 32bit	x64
eax	rax
ebx	rbx
ecx	rcx
edx	rdx

MACRO LETTURA/SCRITTURA

Legge numero: GET_DEC 4, eax
Legge stringa: GET_STR 4, ebx
Legge carattere: GET_CHAR eax
Stampa numero PRINT_DEC eax
Stampa carattere: PRINT_CHAR "a"
Stampa stringa: PRINT_STRING "hello world"
A capo: NEWLINE