

Relazione S10L3

L'esercizio di oggi prevede di identificare lo scopo di ogni istruzione in assembly:

mov EAX, 0x20 → sposta il numero 32 (notazione decimale) nel registro EAX.

mov EDX, 0x38 → sposta il numero 56 (notazione decimale) nel registro EDX.

add EAX, EDX → somma il valore presente nel registro EDX a quello presente nel registro EAX, in pratica $32 + 56 = 88$.

mov EBP, EAX → sposta il contenuto dal registro EAX a quello EBP

cmp EBP, 0xa → istruzione condizionale che controlla i valori di destinazione e sorgente, in questo caso 88 (perché l'abbiamo spostato nel comando precedente) e 10. Poiché destinazione è maggiore della sorgente avremo che i flag **ZF** e **CF** verranno entrambi impostati a 0.

jge 0x1176 <main+61> → quest'istruzione consiste in un salto condizionale a seconda del fatto che l'istruzione precedente abbia la destinazione maggiore della sorgente. Siccome nel nostro caso è vera allora salteremo all'indirizzo di memoria indicato.

mov EAX, 0x0 → sposta il valore 0 (notazione decimale) nel registro EAX.

call 0x1030 <printf@plt> → si utilizza call per chiamare la funzione printf di C, la parte @plt è una struttura che rende più facile il caricamento e il collegamento dinamico delle librerie. Utilizzando la funzione call viene creato uno stack per tenere in memoria i parametri della funzione.

Davide Lecci