

Codice Assembly

Ci viene chiesto di identificare lo scopo di ogni istruzione in codice assembly:

mov EAX, 0x20 => sposta il numero 32 nel registro EAX

mov EDX, 038 => sposta il numero 56 nel registro EDX

add EAX, EDX => fa una somma tra registro EAX ed EDX, quindi somma $32 + 56 = 88$

mov EBP, EAX => sposta il contenuto di EAX nel registro EBP

cmp EBP, 0xa => fa un confronto fra il valore del registro EBP (88) e il valore 10; queste comparazioni servono per impostare il valore dello **ZF**. In un secondo momento questo valore serve per capire quali salti condizionali compirà il codice.

jge 0x1176 => questo è un salto condizionale che verrà compiuto se la comparazione precedente abbia una destinazione maggiore della sorgente. Come nel nostro caso.

mov EAX 0x0 => spostare il valore 0 nel registro EAX.

call 0x1030 <printf@plt> => chiama la funzione printf simile a quella utilizzata in linguaggio C. @plt è utilizzato per rendere più facile il caricamento delle librerie.