

ESERCIZIO M6 D3

1. 0x00001141 <+8>: mov EAX,0x20

Questa istruzione sposta il valore esadecimale 0x20 (decimale: 32) nel registro EAX. Ciò significa che il registro EAX contiene ora il valore 32.

2. 0x00001148 <+15>: mov EDX,0x38

Questa istruzione sposta il valore esadecimale 0x38 (decimale: 56) nel registro EDX. Ciò significa che il registro EDX contiene ora il valore 56.

3. 0x00001155 <+28>: add EAX,EDX

Questa istruzione somma i valori nei registri EAX e EDX e memorizza il risultato in EAX. Quindi, il valore contenuto in EAX (32) viene sommato con il valore contenuto in EDX (56) e il risultato (88) viene salvato di nuovo nel registro EAX.

4. 0x00001157 <+30>: mov EBP, EAX

Questa istruzione copia il valore del registro EAX nel registro EBP. Quindi, il valore 88 contenuto in EAX viene copiato nel registro EBP.

5. 0x0000115a <+33>: cmp EBP,0xa

Questa istruzione confronta il valore nel registro EBP (88) con il valore esadecimale 0xa (decimale: 10). La CPU esegue una sottrazione implicita tra EBP e 0xa e imposta i flag del processore in base al risultato.

6. 0x0000115e <+37>: jge 0x1176 <main+61>

Questa istruzione salta all'indirizzo di memoria 0x1176 (etichettato come "main+61") se il confronto precedente ha prodotto un risultato "maggiore o uguale" (jump if greater or equal). In altre parole, se il valore nel registro EBP è maggiore o uguale a 10, il programma salta all'indirizzo 0x1176 per eseguire le istruzioni successive.

7. 0x0000116a <+49>: mov eax,0x0

Questa istruzione sposta il valore 0 nel registro EAX. Quindi, il registro EAX viene impostato a 0.

8. 0x0000116f <+54>: call 0x1030 <printf@plt>

Questa istruzione chiama la funzione printf, che si trova all'indirizzo di memoria 0x1030. La funzione printf sarà responsabile di stampare un output a schermo, utilizzando i valori presenti nei registri o in memoria.