Studio Assembly

Traccia:
Identificare lo scopo di ognuna delle seguenti istruzioni:

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

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

0x00001155 <+28>: add   EAX,EDX

0x00001157 <+30>: mov   EBP, EAX

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

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

0x0000116a <+49>: mov   EAX,0x0

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

Analisi delle Istruzioni:

1. **0x00001141 <+8>: mov EAX,0x20**
   - L'operazione mov copia il valore esadecimale 0x20 (32 in decimale) nel registro EAX.
2. **0x00001148 <+15>: mov EDX,0x38**
   - L'operazione mov copia il valore esadecimale 0x38 (56 in decimale) nel registro EDX.
3. **0x00001155 <+28>: add EAX,EDX**
   - L'operazione add esegue una somma tra il valore contenuto nel registro EAX e il valore nel registro EDX, salvando il risultato nel registro EAX. Considerando i valori copiati dalle prime due istruzioni, dopo questa terza istruzione il registro EAX conterrà il valore 88 (decimale), ovvero 32 + 56, mentre EDX rimarrà invariato a 56 (decimale).
4. **0x00001157 <+30>: mov EBP, EAX**
   - L'operazione mov copia il valore contenuto in EAX nel registro EBP, quindi dopo questa istruzione i due registri conterranno lo stesso valore, 88.
5. **0x0000115a <+33>: cmp EBP,0xa**
   - L'operazione cmp effettua un confronto tra il valore contenuto nel registro EBP (destinazione) e il valore esadecimale 0xa (10 in decimale), sottraendo la sorgente dalla destinazione, quindi EBP - 0xa (88 - 10 = 78).
   - Le flag di stato ZF e CF saranno:
     - ZF = 0 perché il risultato non è 0 e quindi i due numeri non sono uguali.
     - CF = 0 perché 88 è maggiore di 10.
6. **0x0000115e <+37>: jge 0x1176 <main+61>**
   - Questa istruzione si riferisce al risultato di cmp dell'istruzione precedente. L'operazione jge salta al punto della memoria specificato (0x1176 <main+61>) se nell'istruzione cmp la destinazione è maggiore o uguale all'origine. In questo caso quindi il flusso di esecuzione verrà portato a 0x1176 <main+61>.
7. **0x0000116a <+49>: mov EAX,0x0**

- L'operazione mov copia il valore 0x0 (0 in decimale) nel registro EAX, che passerà quindi da contenere il valore 88 al valore 0.

8. **0x0000116f <+54>: call 0x1030 printf@plt**
   - L'operazione callesegue le seguenti operazioni:
     - Chiama la funzione 0x1030 <printf@plt>,
     - Salva l'indirizzo dell'istruzione successiva,
     - Esegue la funzione 0x1030 <printf@plt>,
     - Terminata la funzione, fa partire l'istruzione successiva (indirizzo salvato prima di eseguire la funzione).