

CONVERSIONE DI CODICE IN ASSEMBLY

0x00001141 <+8>: mov EAX,0x20 Il codice in questione assegna il valore esadecimale 0x20 (32 in decimale) al registro EAX.

0x00001148 <+15>: mov EDX,0x38 Questo codice assegna il valore esadecimale 0x38 (56 in decimale) al registro EDX.

0x00001155 <+28>: add EAX,EDX Questo codice esegue un'operazione di somma tra i valori dei registri EAX e EDX e il risultato viene memorizzato nel registro EAX. In altre parole, il valore corrente di EAX viene incrementato di 56 (il valore corrente di EDX).

0x00001157 <+30>: mov EBP, EAX Questo codice copia il valore del registro EAX nel registro EBP. In questo modo, il valore di EAX viene salvato in EBP per poter essere utilizzato in seguito nella funzione o nel programma.

0x0000115a <+33>: cmp EBP,0xa Questo codice confronta il valore del registro EBP con il valore esadecimale 0xa (10 in decimale). L'istruzione di confronto (cmp) sottrae 10 da EBP e imposta i flag di confronto appropriati, ma non modifica il valore del registro. Il confronto viene utilizzato spesso per controllare se un valore è maggiore o minore di un altro valore o se sono uguali.

0x0000115e <+37>: jge 0x1176 <main+61> Questo codice è un'istruzione di salto condizionale. Se il confronto precedente (cmp) ha impostato il flag di confronto "greater or equal" (cioè EBP è maggiore o uguale a 10), allora il salto condizionale (jge) eseguirà un salto alla posizione di memoria 0x1176 (il cui indirizzo è calcolato in base alla posizione corrente del codice e all'offset specificato). In caso contrario, il salto non viene eseguito e l'esecuzione continua alla successiva istruzione di codice. In pratica, questo codice controlla se il valore di EBP è maggiore o uguale a 10 e, se lo è, salta alla posizione di memoria 0x1176.

0x0000116a <+49>: mov eax,0x0 Questo codice sposta il valore costante esadecimale 0x0 (che corrisponde a 0 in decimale) nel registro EAX. L'istruzione di movimento (mov) è usata per spostare o copiare i dati da una locazione di memoria a un registro o viceversa. In questo caso, il valore 0x0 viene spostato nel registro EAX, sovrascrivendo il valore precedente contenuto in quel registro (se c'è).+

0x0000116f <+54>: call 0x1030 printf@plt Questo codice chiama la funzione printf().