Progetto S10-L3

Studente: Simone Mininni

Task: Fondamenti di Assemblyx86, commentare riga per riga il codice proposto in assembly.

Porzione di codice:
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>

Riga 1:
0x00001141 <+8>: mov EAX,0x20
L' istruzione mov copia il valore immediato 0x20(decimale = 32) nel registro EAX.

Riga 2:
0x00001148 <+15>: mov EDX,0x38
L' istruzione mov copia il valore immediato 0x38(decimale = 56) nel registro EDX.

Riga 3:
0x00001155 <+28>: add EAX,EDX
L'istruzione add somma il contenuto di EAX con quello di EDX e lo sovrascrive in EAX.
0x20+0x38 → 32 + 56 = 88.

Riga 4:
0x00001157 <+30>: mov EBP, EAX
L' istruzione mov copia il valore contenuto in EAX (88) nel registro EBP che punta alla base dello stack.

Riga 5:
0x0000115a <+33>: cmp EBP,0xa
L’ istruzione cmp confronta il valore immediato 0xa(decimale=10) effettuando una sottrazione e aggiorna il registro EFLAGS.
In questo caso si ha 88-10=78 quindi 88 è maggiore di 10 e la zero flag e la carry flag saranno impostate a 0.

Riga 6:
0x0000115e <+37>: jge 0x1176 <main+61>
Dopo il confronto con cmp, jge valuta il risultato del cmp vedendo il registro EFLAGS, e se il valore contenuto in EBP è maggiore o uguale di 0xa(decimale=10) fa un jump all'indirizzo 0x1176 dove parte la funzione <main+61>. Quindi si comporta come un costrutto iterativo(for, while).

Riga 7:
0x0000116a <+49>: mov EAX,0x0
Una volta uscito dal ciclo, il registro EAX viene inizializzato con il valore immediato 0x0(dediamale = 0).

Riga 8:
0x0000116f <+54>: call 0x1030 <printf@plt>
Infine call chiama la funzione printf.