Nell’esercizio di oggi vedremo alcune istruzioni in linguaggio assembly. Assembly è un linguaggio di programmazione di basso livello, cioè molto vicino al linguaggio macchina. Esso è composto da istruzioni che sono divise in codice mnemonico, ovvero l’istruzione che esegue il processore, e operandi. Viene usato per programmare i processori e ne esistono diversi tipi proprio perché varia in base a l’architettura.

1) 0x00001141 <+8>: mov EAX,0x20
2) 0x00001148 <+15>: mov EDX,0x38
3) 0x00001155 <+28>: add EAX,EDX
4) 0x00001157 <+30>: mov EBP, EAX
5) 0x0000115a <+33>: cmp EBP,0xa
6) 0x0000115e <+37>: jge 0x1176 <main+61>
7) 0x0000116a <+49>: mov eax,0x0
8) 0x0000116f <+54>: call 0x1030 <printf@plt>

Nell’esercizio ci viene chiesto di identificare e spiegare le varie istruzioni elencate sopra:

1) “mov” è l’istruzione che permette di copiare un dato; l’istruzione copia il valore 20 (32 in decimale) nel registro EAX;
2) L’istruzione copia il valore 38 (56 in decimale) nel registro EDX;
3) L’istruzione “add” viene usata per sommare dati. Il valore nel registro EDX viene sommato a quello presente in EAX e il risultato salvato in quest’ultimo;
4) L’istruzione copia il valore del registro EAX nel registro EBP;
5) L’istruzione “cmp” permette di eseguire la sottrazione tra due valori come “sub”, ma a differenza di quest’ultimo cambia i flag zero e curry. Il primo diventa 1 se il risultato della sottrazione è 0, mentre il secondo diventa 1 se c’è un riporto;
6) L’istruzione “jge” esegue un salto alla locazione 4470<main+61>. La CPU eseguirà le istruzioni specificate su questo indirizzo;
7) L’istruzione copia il valore 0 nel registro EAX;
8) L’istruzione “call” serve per richiamare un sottoprogramma, nel nostro caso viene chiamata la procedura 4144<printf@plt>.