

Le parti scritte in 0xYY sono in esadecimale (hexadecimal), le parti prima dei due punti sono riferite agli indirizzi di memoria in cui è conservata l'istruzione.

0x00001141 <+8>: mov EAX,0x20 ; copia il valore 0x20 (32 in decimale) nel registro EAX (Extended Accumulator)

0x00001148 <+15>: mov EDX,0x38 ; copia il valore 0x38 (56) nel registro EDX (Extended Data Register)

0x00001155 <+28>: add EAX,EDX ; aggiunge il contenuto di EDX, ad EAX

0x00001157 <+30>: mov EBP, EAX ; copia il contenuto di EAX, in EBP (Extended Base Pointer)

0x0000115a <+33>: cmp EBP,0xa ; compara due valori, in questo caso 0xa (10) e ciò contenuto in EBP. Poi assegna delle flags in base al risultato della comparativa, più precisamente il ZF (Zero Flag) e il CF (Carry Flag). Queste flags definiscono il comportamento delle successive parti di codice (come ad esempio per degli Jump).

0x0000115e <+37>: jge 0x1176 <main+61> ; salta alla locazione specificata (0x1176/4470) se la destinazione è maggiore o uguale alla sorgente nella funzione "cmp" (quella precedente è un'ottimo esempio).

0x0000116a <+49>: mov EAX,0x0 ; in questo caso, copia il valore 0x0 (che sarebbe sempre 0) nel registro EAX, "resettandolo"

0x0000116f <+54>: call 0x1030 <printf@plt> ; il comando "call" fa una chiamata ad una funzione (in questo caso, printf) che si trova all'indirizzo 0x1030 (4144)