

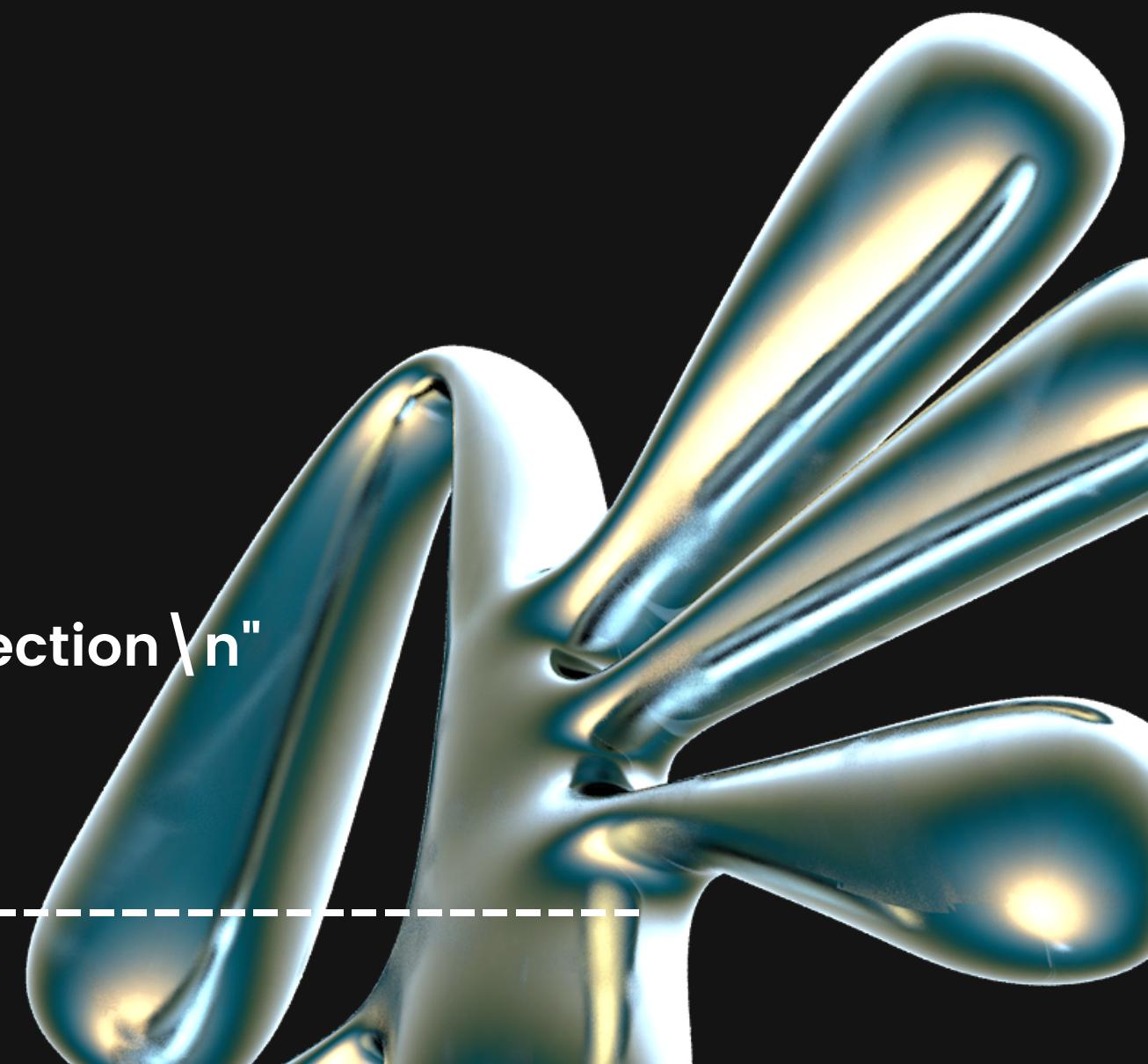
C# / Assembly x86

Ernesto Robles

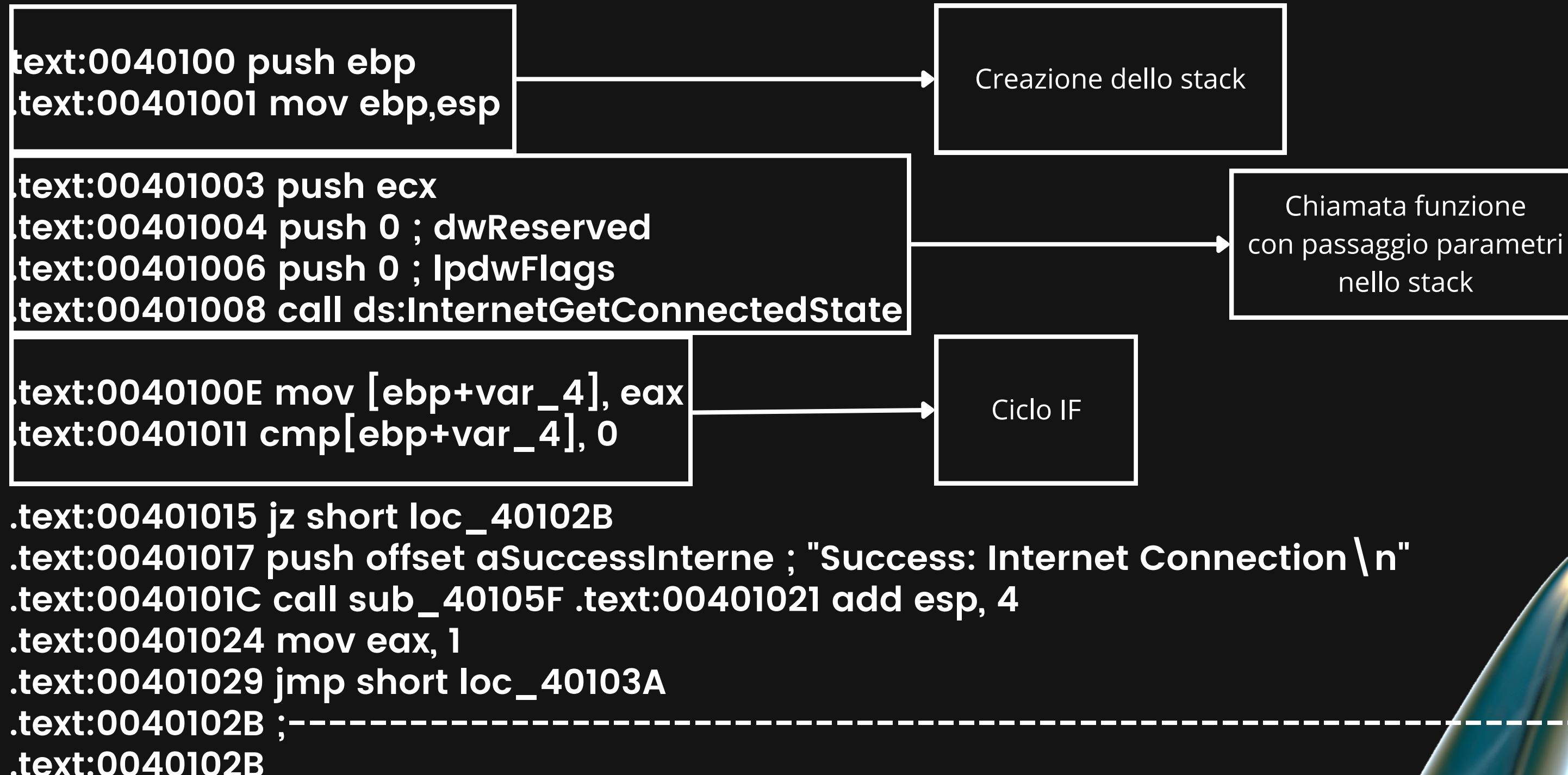
Assembly x86

Dato il seguente codice in Assembly per la CPU x86,
verranno identificati eventuali di programmazione
appartenenti al C i costrutti e lo scopo di ogni
istruzione.

```
text:0040100 push ebp
.text:00401001 mov ebp,esp
.text:00401003 push ecx
.text:00401004 push 0 ; dwReserved
.text:00401006 push 0 ; lpdwFlags
.text:00401008 call ds:InternetGetConnectedState
.text:0040100E mov [ebp+var_4], eax
.text:00401011 cmp[ebp+var_4], 0
.text:00401015 jz short loc_40102B
.text:00401017 push offset aSuccessInterne ; "Success: Internet Connection\n"
.text:0040101C call sub_40105F .text:00401021 add esp, 4
.text:00401024 mov eax, 1
.text:00401029 jmp short loc_40103A
.text:0040102B ;-----
.text:0040102B
```



Assembly x86



Assembly x86

Evidenziando il blocco:

```
.text:00401011 cmp[ebp+var_4], 0  
.text:00401015 jz short loc_40102B
```

Si può notare come avvenga un salto a seconda del risultato della comparazione [ebp+var_4] con zero, questo processo lo possiamo identificare come costrutto “if” guardando le istruzioni di confronto (cmp) e di salto condizionato (jz), oltre l’istruzione if non sembrano esserci altri rappresentazioni di costrutti tipici del linguaggio C come cicli for o while che potrebbero trovarsi in parti del codice qui non fornite.

Esempio rappresentativo codice C del blocco (cmp-jz)

```
if (var_4 == 0) {  
    // esecuzione codice se la condizione è vera  
} else {  
    // esecuzione codice se la condizione è falsa  
}
```

Lo scopo del codice Assembly è la gestione di uno scenario in cui la funzione InternetGetConnectedState restituisce un valore diverso da zero “connessione internet riucita” e in tal caso il programma stampa il messaggio “Success: Internet Connection \n” chiamando la subroutine sub_40105F.

Assembly x86

push ebp: Salva il valore del registro di base (ebp) nello stack.

mov ebp, esp: Imposta il registro di base (ebp) al valore corrente dello stack (esp).

push ecx: Salva il valore del registro ecx nello stack.

push 0: Mette zero nello stack (dwReserved).

push 0: Mette zero nello stack (lpdwFlags).

call ds:InternetGetConnectedState: Chiama la funzione InternetGetConnectedState per verificare lo stato della connessione a Internet. I parametri sono zero e zero (dwReserved e lpdwFlags).

mov [ebp+var_4], eax: Salva il risultato della chiamata di funzione in [ebp+var_4].

cmp [ebp+var_4], 0: Compara il valore salvato in [ebp+var_4] con zero.

Assembly x86

jz short loc_40102B: Salta a loc_40102B se il risultato della comparazione è zero (InternetGetConnectedState ha restituito 0, indicando che non c'è connessione).

push offset aSuccessInterne ; "Success: Internet Connection \n": Mette l'offset della stringa "Success: Internet Connection \n" nello stack.

call sub_40105F: Chiama una subroutine (sub_40105F) che probabilmente stampa il messaggio di successo.

add esp, 4: Libera spazio nello stack dopo la chiamata della funzione.

mov eax, 1: Imposta il registro eax a 1.

jmp short loc_40103A: Salta a loc_40103A.

loc_40102B: Etichetta del codice.