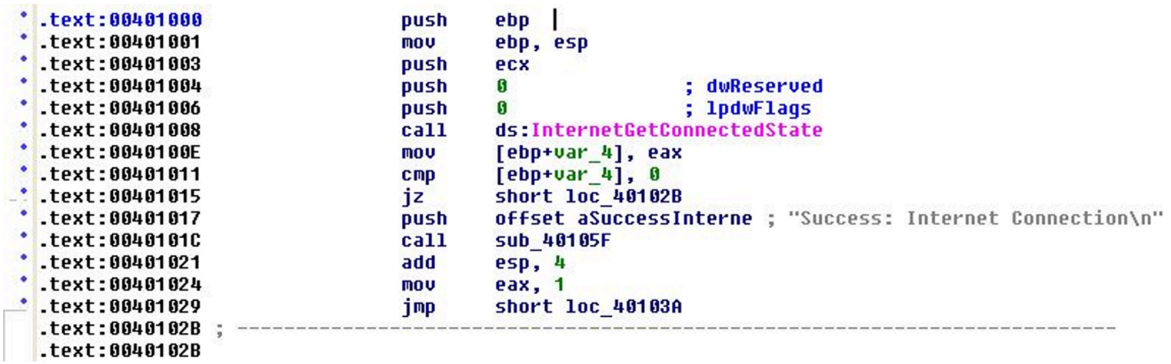
Linguaggio Assembly x86

Ci viene fornito il seguente codice Assembly, ci viene chiesto di identificare i costrutti noti.



```
.text:00401000                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
```

push ebp |
mov ebp, esp
Queste due funzioni servono per creare un nuovo stack, definendo l'inizio e la fine dello stack

push 0 ; dwReserved
push 0 ; lpdwFlags
Vengono inserite due variabili a valore 0 rispettivamente: un valore intero DWord riservato alla funzione che segue, si può presupporre che lp indica un puntatore alla DWord Flags (quindi punta a dei flags che influenzano la funzione che segue)
Entrambi queste variabili fanno riferimento alla funzione:

call ds:InternetGetConnectedState
Viene chiamata la funzione che sembra essere una verifica della connessione internet.
Quindi in base al valore delle due variabile precedenti si verifica o meno se c'è connessione ad internet, nello specifico se il valore delle variabili è diverso da quello impostato (0).

mov [ebp+var_4], eax
cmp [ebp+var_4], 0
jz short loc_40102B
Questo blocco salva come prima cosa il risultato nella funzione InternetGetConnectedState (si può interpretare eax come la variabile che rappresenta il risultato della funzione), in uno specifico spazio dello stack;
poi compara il risultato della funzione con il valore 0, se risulta uguale 0 salta alla porzione di memoria loc_40102B. Supponiamo che questa locazione di memoria dice che essendo 0 **NON** c'è connessione.

Viceversa se il confronto è diverso da 0 continua con l'esecuzione del codice:

```
push    offset aSeccessInternet
call    sub_40105F
```

Quindi carica la stringa (offset) di successo di connessione ad internet, chiama la subroutine 40105F, forse una serie di operazione quando avviene la connessione?

Infine:

```
add     esp, 4
mov     eax, 1
jmp     short loc_40103A
```

Viene aggiunto un valore per eliminare le operazioni precedenti, e conclude con nl salto ad un indirizzo specifico loc_40103A, penso che sia un salto di uscita dal programma in quanto la penultima riga, mov eax, 1, assegna un valore specifico forse ad indicare il completamento dell'esecuzione del programma???

Conclusione:

Ho presupposto che il programma serva a verificare se c'è o meno connesione ad internet, in caso di risposta positiva entra in esecuzione una subroutine, che una volta conclusa termina l'esecuzione del programma.

In linguaggio C potrebbe equivalere ad un costrutto if che identifica se c'è o meno connessione ad Internet.