

# Esercitazione W23D1 - Pratica 2

## Assembly C

Fabio Benevento - 10/04/2024

### Traccia

Dato il seguente codice assembly, provare a ricostruire le istruzioni originali in C

```
push    %ebp
mov     %esp,%ebp
sub     $0x8,%esp
call    80483e9 <bar>
leave
ret
```

```
push    %ebp
mov     %esp,%ebp
sub     $0x8,%esp
call    80483fb <baz>
call    8048400 <quux>
leave
ret
```

```
push    %ebp
mov     %esp,%ebp
pop     %ebp
ret
```

```
push    %ebp
mov     %esp,%ebp
mov     $0x0,%eax
movl    $0x1, (%eax)
pop     %ebp
ret
```

```
push    %ebp
mov     %esp,%ebp
and     $0xfffffffff0,%esp
call    80483dc <foo>
mov     $0x0,%eax
leave
ret
```

## Svolgimento

Analizzando il codice, ognuno dei pezzi di codice è una funzione in quanto è possibile identificare la struttura tipica di inizializzazione dello stack

```
push %ebp
mov %esp, %ebp
```

L'ultima funzione è verosimilmente l'istruzione main in quanto contiene le istruzioni di pulizia dello stack prima dell'uscita del programma.

Molte delle funzioni chiamano ricorsivamente le altre.

E' possibile quindi ipotizzare la corrispondenza con il seguente programma C


```
void foo();
void bar();
void baz();
void quux();

void foo()
{
    bar();
}

void bar()
{
    baz();
    quux();
}

void baz()
{
    //do nothing
}

void quux()
{
    *(int*)(0) = 1
}
```



```
int main()
{
    foo();
    return 0;
}
```