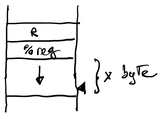
SC 31/3/2026

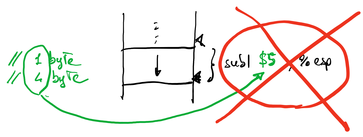
31/03/26 10:13

ALLINEAMENTO IN STACK

```
f:
    pushl %reg
    subl  $x, %esp
    ⋮
```



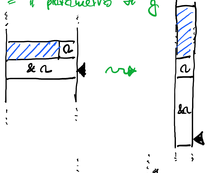

```
void g (char * c);
void f() {
    ⋮
    char a
    g(&a);
    ⋮
}
```



IMPORTANTE: In qualsiasi momento dell'esecuzione di qualsiasi funzione, lo stack pointer DEVE contenere un indirizzo multiplo di 4

(ATTENZIONE: il compilatore tipicamente allinea a 16 byte)

```
f:
    subl $4, %esp      # spazio per a
    subl $4, %esp      # = = il parametro di g
    ⋮
    leal ---
    movl ---, (%esp)
    call g
    ⋮
```

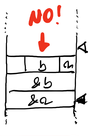



```
char a,b,c
g(&a, &b, &c)
```

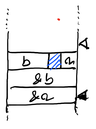
→ `# PROLOGO` `subl $16, %esp`


```
char a;
short b;
g(&a, &b);
```

→ `# PROLOGO` `subl $12, %esp`



IMPORTANTE: I dati di dimensione x nella stack sono allineati ad indirizzi multipli di x

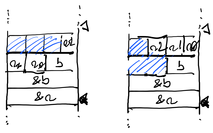



```
char [5]a;
short b;
g(&a, &b);
```

→ `#PROLOGO` `subl $16, %esp`


```
char [3]a;
short b;
g(&a, &b);
```

→ `#PROLOGO` `subl $16, %esp`



ALLINEAMENTO IN MEMORIA DI STRUCT

```
struct s {
    char x;
    short y;
    int z;
}
```

(1) Dove inizia in memoria ciascun campo della struct?

(2) Quanto spazio occupa complessivamente la struct in memoria?

L'allineamento in memoria delle struct segue le regole:

1. Un dato di dimensione x deve trovarsi allineato in memoria ad un indirizzo multiplo di x
2. La struct in memoria ha un allineamento complessivo che deriva dal campo di tipo più grande
3. La dimensione complessiva della struct in memoria è multipla della dimensione del suo campo più grande
4. Il compilatore NON riordina mai i campi di una struct

Esempi:

```
struct s {
    char x;
    int y;
}
```

base
+0 |x • • • |
+4 |x x x x|
sizeof s = 8

```
struct s {
    char x;
    int y;
    short z;
}
```

base
+0 |x • • • |
+4 |x x x x|
+8 |x x • • |
sizeof s = 12

```
struct s {
    char x;
    short z;
    int y;
}
```

base
+0 |x •
+2 x x|
+4 |x x x x|
sizeof s = 8

Esempio:

```
struct s {
    char x;    //+0
    int y;     //+4
}
void f(struct s * p, char a, int b) {
    p -> x = a;
    p -> y = b;
}
```

→

```
global f
f:
    movl 4(%esp), %eax
    movb 8(%esp), %cl
    movl 12(%esp), %edx
    movb %cl, (%eax)
    movl %edx, 4(%eax)
    ret
```