SC 18/3/2026

18/03/26 09:28

CHIAMATA A FUNZIONE CON PARAMETRI (CONTINUA...)

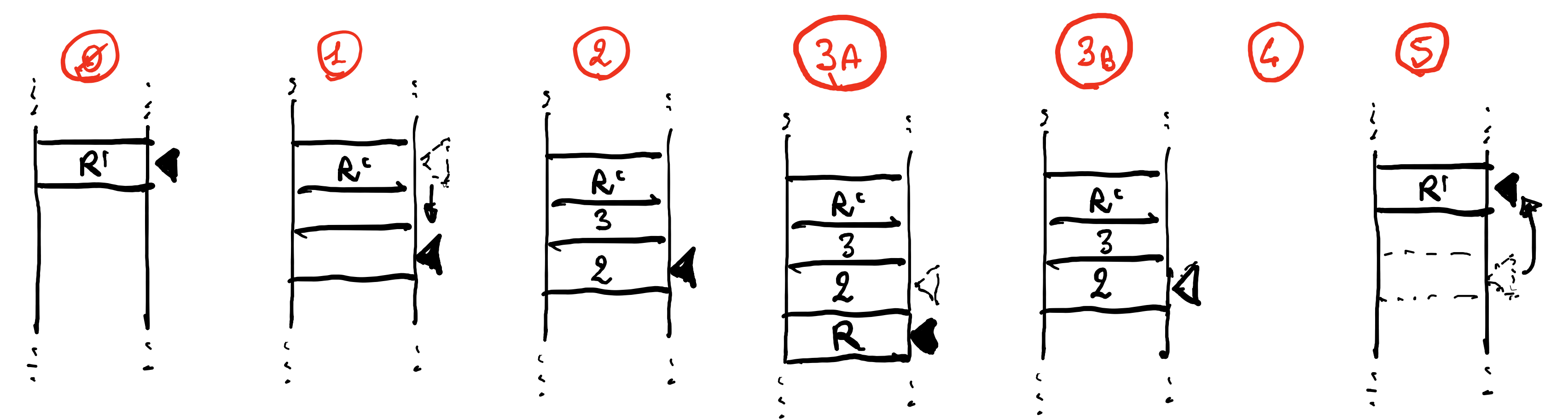
```
int f() {
    int x = g(2,3);
    x++;
    return x;
}
```

=>

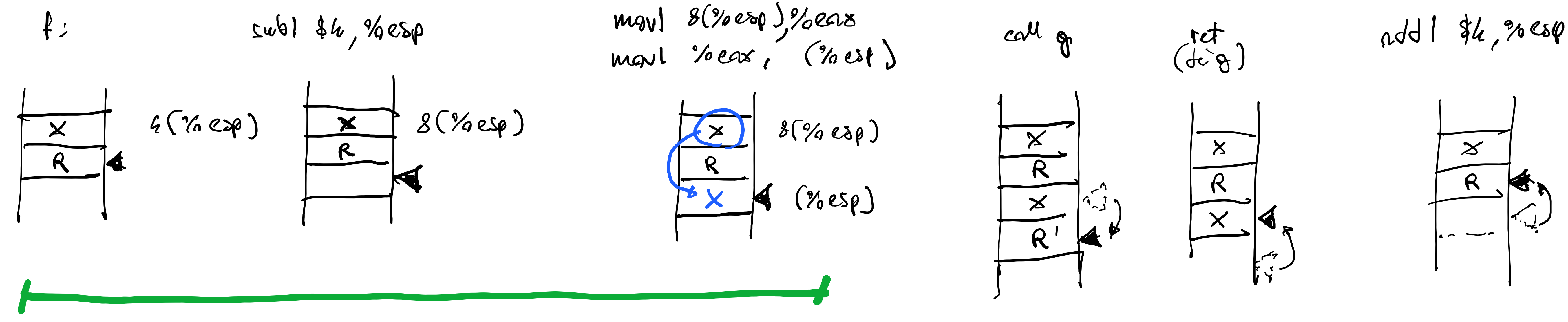
```
.globl f
f:
⓪[ subl $8, %esp          # riserviamo spazio
                           per parametri
②[ movl $3, 4(%esp)       # scrivo i parametri
 [ movl $2, (%esp)          attuali nello spazio che ho
                           riservato
③[ call g
④[ incl %eax
⑤[ addl $8, %esp
⑥[ ret
```

PROLOGO

EPILOGO



Esempio esecuzione "E3-arg-from-arg"



UTILIZZO DEI REGISTRI

L' ABI IA32 divide i registri general purpose in due gruppi:

CALLER SAVED

A, C, D

- La funzione chiamante (caller) deve salvare il valore di questi registri prima di chiamare un'altra funzione
- La funzione chiamata (callee) può modificare liberamente il valore di questi registri

CALLEE SAVED

B, DI, SI, BP

- La funzione chiamata (callee) deve preservare il valore che questi registri avevano nel momento in cui viene eseguita la call, e ripristinarli prima di eseguire ret.

COME SALVARE IL VALORE DI UN REGISTRO?

① Salviamo il valore di %REG nella stack

② Eseguiamo le operazioni che "sporcano" %REG

③ Recuperiamo dalla stack il valore di %REG

▶ METODO ESPLICITO

①
```
subl $4, %esp
movl %REG, (%esp)
```

②
```
# %REG viene sporcato
```

③
```
movl (%esp), %REG
addl $4, %esp
```

▶ PUSH + POP

①
```
pushl %REG
```

②
```
# ....
```

③
```
popl %REG
```

QUALI REGISTRI POSSO USARE?

- Implemento una funzione che NON chiama altre funzioni
  - > Uso liberamente i caller saved A, C, D
  - > Se non sono sufficienti, allora uso i callee saved (B, DI, SI, BP) ma devo preservarne il valore (prologo + epilogo)
- Implemento una funzione che chiama altre funzioni
  - > Uso i callee saved (B, DI, SI, BP) preservandone il valore
  - > Uso liberamente i caller saved (A, B, D) per valori temporanei SOLO in blocchi di istruzioni che NON chiamano altre funzioni
  - > Uso i caller saved (A, C, D) ma ne preservo il valore (push + pop) prima di QUALSIASI chiamata a funzione.