

Argomenti del main

Stefano Ferrari

Università degli Studi di Milano
stefano.ferrari@unimi.it

Programmazione

anno accademico 2016–2017

Il main come funzione

Anche il **main** è una **funzione** chiamata dal sistema operativo

- ▶ ha dei **dati**, descritti dai parametri formali nell'intestazione
- ▶ ha un **risultato**, descritto dal tipo restituito nell'intestazione

Tuttavia

- ▶ **il numero e tipo dei parametri non è noto a priori**
- ▶ **il risultato è sempre intero**

Questo apre una serie di domande

- ▶ come si definisce il numero e tipo dei parametri?
- ▶ come si passano i parametri attuali?
- ▶ a che serve e come si recupera il risultato?

Parametri formali e attuali

```
int main (int argc, char *argv[])
```

I parametri formali non sono definiti direttamente, bensì attraverso

- ▶ `int argc`, che è il **numero dei parametri**
- ▶ `char *argv[]`, che è un **vettore dinamico di lunghezza `argc` composto da stringhe** che descrivono i parametri

I parametri attuali vengono passati attraverso la linea di comando

- ▶ `argv[0]` è il **nome del programma**
- ▶ `argv[1]` è il **primo parametro**
- ▶ `argv[2]` è il **secondo parametro**
- ▶ ...

Esempio

Se da terminale si scrive la linea di comando

```
gcc codice.c -o codice.exe
```

il programma gcc riceve i seguenti parametri

- ▶ `argc` vale 4
- ▶ `argv[0]` vale "gcc"
- ▶ `argv[1]` vale "codice.c"
- ▶ `argv[2]` vale "-o"
- ▶ `argv[3]` vale "codice.exe"

Queste stringhe possono essere usate per guidare il programma

Risultato

Il risultato di un programma C è sempre un intero

La libreria `stdlib.h` definisce le costanti simboliche

- ▶ `EXIT_SUCCESS` (0) da usare se il programma ha avuto successo
- ▶ `EXIT_FAILURE` (-1) da usare in caso di errore

Si usa specificare il tipo di errore definendo altre costanti simboliche

Il valore numerico viene restituito al sistema operativo

- ▶ alcuni ambienti di compilazione lo indicano all'utente
- ▶ i file batch o script lo possono usare