

Argomenti della funzione main

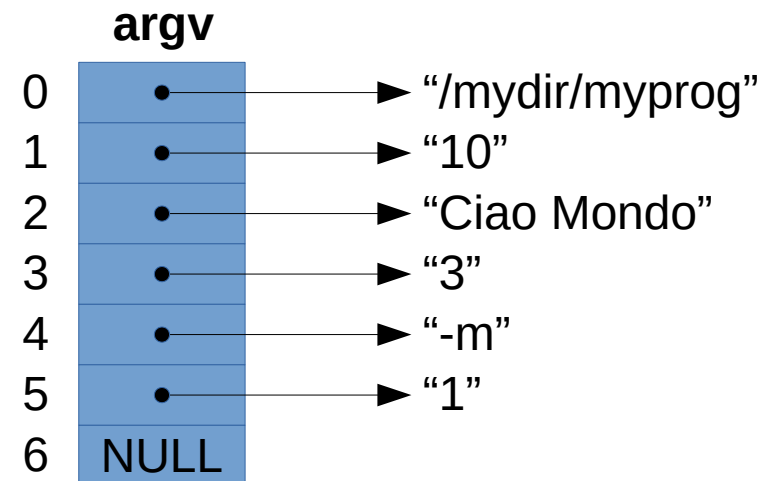
Funzione main

- La funzione `main()` è una funzione C come tutte le altre.
- Ci sono alcune peculiarità:
 - Una funzione con il nome 'main' deve essere sempre presente in un programma eseguibile C
 - E' la prima funzione che viene invocata (dal Sistema Operativo) per far partire il programma
- Le sue signature tipiche sono:
 - **`int main(int argc, char *argv[])`**
 - **`int main(void)` // usata quando non ho argomenti**
 - **argc** è il numero di argomenti della funzione main (sempre ≥ 1)
 - **argv** è un array di stringhe che ha come ultima entry NULL (cioè `argv[argc] == NULL`) e come prima entry il nome del programma (cioè `argv[0] == "myprog"`)
 - Ci sono altre signature accettate dai compilatori C, alcune non standard!
- Il valore di ritorno del main è un intero. Convenzionalmente:
 - 0 indica “programma terminato correttamente”
 - $\neq 0$ indica “programma terminato con errore”, tipicamente l'intero indica l'errore (vedere ad esempio i codici di errore in `"/usr/include/asm-generic/errno-base.h"`)
 - Per vedere il codice di ritorno di un programma si testa la variabile di shell '\$?'

Parametri del main

`/mydir/myprog 10 "Ciao Mondo" 3 -m 1`

- `argc = 6`
- `argv` è un vettore contenente un puntatore a char
- es. `argv[2][1] == 'i'` `argv[2][10] == '\0'`
- Per trasformare la stringa "10" in numero intero si può usare:
 - `atoi()` `atol()`
 - `strtol()` **// raccomandata !**
 - `sscanf()`



```
long x = atol(argv[1]);  
long x = strtol(argv[1],NULL,10);  
long x; sscanf(argv[1], "%ld", &x);
```

Per convertire da stringa ad intero usare **sprintf (snprintf)**

Esempio

- Scrivere un programma che somma tutti gli argomenti passati a linea di comando.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    long sum = 0;

    while( --argc > 0 )
        sum += strtol(argv[argc], NULL, 10);
    printf("sum=%ld\n", sum);

    return 0;
}
```

```
$> gcc -std=c99 -Wall -Werror somma.c -o somma
$> seq 2 2 1024 | xargs somma
sum=262656
```

Esempio (cnt)

- Se l'argomento non è un numero?
 - ./somma a b c 1 2 cosa stampa?
 - Vogliamo stampare un messaggio di errore per gli argomenti 'a' 'b' 'c'
- ./somma 9999999999999999999999999999 1 2 cosa stampa?
 - Vogliamo gestire overflow/underflow...
- Dobbiamo leggere **ATTENTAMENTE** il **man** di *strtol*

```
char isNumber(const char* s, long* n) {
    char* e = NULL;
    errno=0;
    long val = strtol(s, &e, 10);
    if (errno == ERANGE) return 2; // overflow/underflow
    if (e != NULL && *e == (char)0) {
        *n = val;
        return 0; // è un numero valido
    }
    return 1; // non è un numero
}
```