Stringhe in C

Luca Abeni

Stringhe in C

- Stringhe: tipo di dato strutturato
  - Sequenze di caratteri
- Linguaggio C: progettato per la semplicità
  - No supporto nativo per le stringhe
  - Stringhe come array di caratteri
  - Operazioni su stringhe come funzioni di libreria
- Lunghezza di una stringa?
  - Convenzione: array terminato con 0 (carattere '\0')

Costanti Stringa

- Una costante stringa in C è rappresentata fra virgolette
  - Esempio: ‘‘`Hi, there!`’’
  - Il compilatore la converte in una sequenza di caratteri
  - Caratteri codificati tramite ASCII
  - Terminatore `0` dopo l'ultimo carattere
- Il compilatore si occupa di inserire automaticamente uno `0` alla fine della stringa
- Possibili utilizzi:
  - `char str[] = "Hi, there!";`
  - `char *str = "Hi, there!";`
  - ...

Stringhe e Puntatori

- `char *str = "Hi there!";` funziona perché...
  - Una stringa costante è un array di caratteri
  - Un array è interpretabile come un puntatore al primo elemento
- Il compilatore trova spazio in memoria per 10 byte (i 9 caratteri e il terminatore 0
- Trova spazio in memoria per un puntatore a char (chiamato `str`)
- Inizializza `str` con l'indirizzo dei 10 caratteri
  - Nota: il valore di `str` può essere modificato dal programma!

Stringhe vs Caratteri

- Differenza fra `char *str = "a";` e `char c = 'a';`
  - `str` è un puntatore inizializzato ad un array di 2 caratteri: `'a'` e `0`
  - `c` è un singolo carattere (`'a'`)
- Ancora: stringa come puntatore a carattere o come array di caratteri
- Nota:
  - `char *str = "Hi, there!";` è ok...
  - `*str = "Hi, there!";` no!

I/O di Stringhe

- Come noto, `scanf()` permette di leggere stringhe
  - Specificatore di formato `%s`
- Ricorda: i parametri di `scanf()` dal secondo in poi devono essere puntatori
  - Operatore `&` prima dei nomi di variabile
  - Le stringhe sembrano essere eccezioni
  - Esempio: `scanf("%d %s", &n, str);`
- Nome di variabile stringa: nome di array
  - Puntatore al primo carattere della stringa
  - Per questo non serve "`&`" !!!

Esempio

```c
#include <stdio.h>

int main()
{
   char str[80];
   unsigned int i = 0;

   scanf("%s", str);
   printf("La stringa immessa e': %s\n", str);

   while (str[i] != 0) {
      printf("%c ", str[i]);
      i = i + 1;
   }
   return 0;
}
```

Funzioni che Operano su Stringhe

- Stringhe come semplici array di caratteri (terminati da `0`)
- Non esistono operazioni che agiscono su stringhe
  - Funzioni definite dalla libreria standard
  - Prototipi in `string.h`: `#include <string.h>`
- Calcolo lunghezza di stringhe, copia di stringhe, etc...
- Consultare documentazione per una lista di tutte le funzioni

Esempio: Lunghezza di una Stringa

- `strlen()` ritorna il numero di caratteri contenuti in una stringa (escluso il terminatore 0)
  - `unsigned int strlen(const char *s);`

```
unsigned int strlen(const char *s)
{
  unsigned int i = 0;

  while(s[i] != 0) {
    i = i + 1;
  }

  return i;
}
```

Esempio: Copia di una Stringa

- `strcpy()` copia una stringa in un array di caratteri
  - `void strcpy(char *dest, const char *src);`

```
void strcpy(char *dst, const char *src)
{
  unsigned int i = 0;

  while(src[i] != 0) {
    dst[i] = src[i];
    i = i + 1;
  }
  dst[i] = 0;

  return i;
}
```