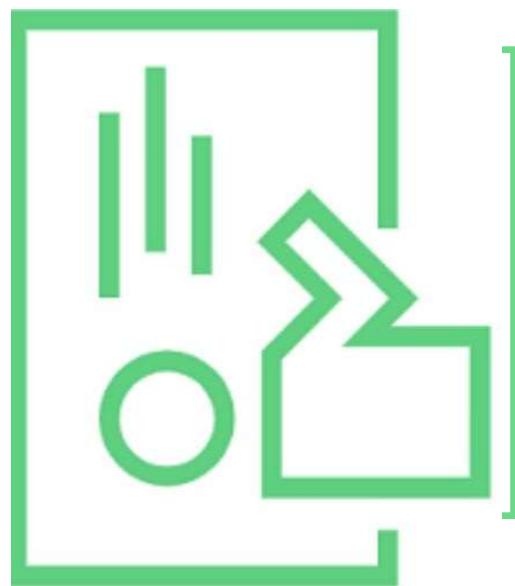


Stefan Holmberg, Systemmentor AB

Strängar i C, del 1



Finns inte ”string”

En sträng i C är en array (SEKvens) av tecken (char[])

- Som har en 0 på slutet
 - Inte tecknen '0' utan värdet 0 eller NULL
 - För att veta hur lång listan är (vi skriver en strlen)
- På engelska har C: "Null-terminated strings"
- printf() tar som parameter en char[] och stegar igenom listan till den när en 0 / NUL

```
| ”alla andra” språk:  
string s1 = “hello”;  
string s2 = “test”;
```

```
if(s1 == s2 ){  
}
```

Deklarera strängar

strings1.c

```
int main() {  
    char str1[] = "Hello!"; ←  
    char str2[] = { 'H', 'e', 'l', 'l', 'o', '\0' }; ←  
    return 0;  
}
```

- Str1 och str2 får likadana värden
- C konverterar allt inom ”..” till en lista över tecken och lägger till en NULL(0)
- Enkel citattecken (‘’) markerar att det är en char och inte en char[]

Mer strings

strings2.c

- Ekvivalenta printf()

```
int main(){
    char str[] = "Hello!";
    printf( str );
    printf( "%s", str );
    return 0;
}
```

Char * vs char[]

strings4.c

```
int main() {  
    char str1[] = "Hello";  
    char *str2 = "Hello";  
  
    printf( "%lu\n", sizeof( str1 ) );  
    printf( "%lu\n", sizeof( str2 ) );
```

- En string med *char[]* är inte samma som en string med *char**
 - *char[]* skapar en ny lista, kopierar texten till den och lägger till en '\0' på slutet
 - Det går att ändra *str1*

strings4.c

- *char** pekar till en konstant som är "Hello"
- *str2* får inte ändras

```
int main() {  
    char str1[] = "Hello";  
    char *str2 = "Hello";  
  
    printf( "%lu\n", sizeof( str1 ) );  
    printf( "%lu\n", sizeof( str2 ) );
```

BUFFER OVERFLOW

strings5.c

```
int main() {
    string str[50];
    scanf( "%s", str );
    printf( "%s", str );
```

- scanf() kan även läsa in en string
- Behöver inte ange &str eftersom str är en lista
- Vad händer om scanf() läser in mer än 50 tecken?

strings6.c

```
#include <string.h>
int main() {
    char str1[] = "Hello";
    char str2[] = "Hello";
    if ( strcmp( str1, str2 ) == 0 ) {
        printf( "%s", str );
    }
}
```

- Inkludera *string.h*
- Använd *strcmp()* om du vill testa if två strängar är likadana
- Returnerar:
 - < 0 om str1 är ”mindre”
 - > 0 om str1 är ”större”
 - == 0 om de är likadana

strcpy

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

int main() {
    char str1[20] = "C programming";
    char str2[20];

    // copying str1 to str2
    strcpy(str2, str1);

    printf(str2); // C programming

    return 0;
}
```

mer

Andra funktioner i string.h:

- strcpy() - kopiera en string upp till en viss längd
- strcat() - sammansatta en string i en annan
- strchr() - hitta en tecken i en string
- strlen() - returnerar längden på en string

- https://en.wikibooks.org/wiki/C_Programming/Strings

string to int

```
int tal = atoi("112345");
```

```
int x = atoi("12monkeys"); // x is 12  
int y = atoi("monkeys12"); // y is 0
```