

Programiranje u fizici

13. Rad sa stringovima

Prirodno-matematički fakultet u Nišu
Departman za fiziku

String – niz karaktera u programskom jeziku C u suštini predstavlja jedno-dimenzionalni niz karaktera koji se završava tkz. null karakterom '\0'.

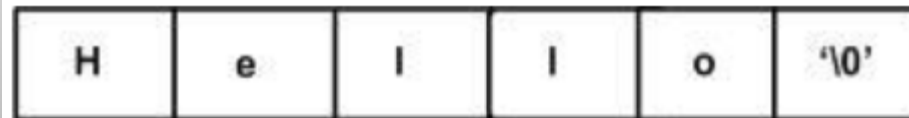
Sledeća deklaracija i inicijalizacija kreira string koji sadrži reč „Hello“.

```
char greeting[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
```

Pošto sadrži null karakter na kraju niza, veličina niza karaktera je za jedan veća od broja slova u samoj reči. Isti string je moguće inicijalizovati na sledeći način:

```
char greeting[] = "Hello";
```

Memorijska reprezentacija stringa Hello je prikazana na slici:



Važno je napomenuti da nije potrebno da programer dodaje null karakter na kraju znakovne konstante.

Sam C prevodilac će automatski dodati null karakter na kraju niza u postupku inicijalizacije istog.

Primer:

```
#include <stdio.h>

int main ()
{
    char pozdrav[6] = {'H', 'e', 'l', 'l', 'o', '\0'};

    printf („Pozdravna poruka: %s\n“, pozdrav);

    return 0;
}
```

```
Pozdravna poruka: Hello
Process returned 0 (0x0)   execution time : 0.022 s
Press any key to continue.
```

Programski jezik C ima niz funkcija za rad sa stringovima.

Funkcija	Objašnjenje
strcpy (s1, s2);	Kopira string s2 u string s1
strcat (s1, s2);	Nadovezuje string s2 na kraj stringa s1
strlen (s1);	Vraća dužinu stringa s1 u karakterima
strcmp (s1, s2);	Vraća 0 ako su s1 i s2 identični stringovi; manje od 0 ako je s1 < s2; veće od 0 ako je s1 > s2
strchr (s1, ch);	Vraća pokazivač na prvo pojavljivanje karaktera ch u stringu s1
strstr (s1, s2);	Vraća pokazivač na prvo pojavljivanje stringa s2 u stringu s1

Kompletna lista funkcija za rad sa stringovima može se naći u biblioteci [string.h](#)

```
#include <stdio.h>
#include <string.h>
int main ()
{
    char str1[12] = "Hello";
    char str2[12] = "World";
    char str3[12];
    int len ;

    /* kopira str1 u str3 */
    strcpy(str3, str1);
    printf("strcpy (str3, str1): %s\n", str3);

    /* spaja str1 i str2 */
    strcat( str1, str2);
    printf("strcat (str1, str2): %s\n", str1);

    /* ukupna duzina str1 posle spajanja */
    len = strlen(str1);
    printf("strlen (str1) : %d\n", len);

    return 0;
}
```

```
strcpy (str3, str1): Hello
strcat (str1, str2): HelloWorld
strlen (str1) : 10

Process returned 0 (0x0)   execution time : 0.024 s
Press any key to continue.
```

Kao što funkciji prosleđujemo parametre, nekada želimo da odmah prosledimo neke parametre programu koji smo pokrenuli.

Primer je program COPY koji možemo pokrenuti iz komandne linije.

Kada ga pokrećemo, pišemo razmak, ime fajla koji treba kopirati, pa razmak i ime novog fajla u koji se sadržaj kopira.

```
e:\>copy br_red.exe novi.exe  
1 file(s) copied.
```

Da bi smo mogli da pišemo slične programe, moramo znati kako da primimo te parametre (argumente) u naš program.

Do sada smo navikli da pišemo funkciju `main` kao:

```
int main () { ... }
```

Međutim, `main` funkcija ima i svoj drugi zapis koji nam omogućava čitanje parametara koji su prosleđeni programu iz komandne linije:

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



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

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

```
{
```

```
    int i;
```

```
    printf("Broj argumenata komandne linije je %d\n", argc);
```

```
    printf("Parametri komandne linije: \n");
```

```
    for (i=0; i<argc; i++) {
```

```
        printf("%s\n", argv[i]);
```

```
    }
```

```
    return 0;
```

```
}
```

```
e:\>prg prvi drugi treci cetvrti  
Broj argumenata komandne linije je 5  
Parametri komandne linije:  
prg  
prvi  
drugi  
treci  
cetvrti
```