

# **Rad sa datotekama Komunikacija sa OS**

# Rad sa datotekama



- Vrste datoteka
- Otvaranje / zatvaranje datoteka
- Pristup datotekama za upis / čitanje
- Funkcije za rad sa datotekama

# Rad sa datotekama...

---

- **Ulazno / izlazni tokovi za rad sa datotekama**
  - **Osnovni koraci u radu sa datotekama:**
    - **uvođenje pokazivača na strukturu tipa FILE**
    - **otvaranje toka za razmjenu podataka sa datotekom i definisanje načina pristupa datoteci (pomoću uvedenog pokazivača)**
    - **upis – čitanje (pomoću uvedenog pokazivača)**
    - **zatvaranje toka za razmjenu podataka sa datotekom**

# Rad sa datotekama...

- **Vrste datoteka:**
  - Postoje dvije osnovne vrste datoteka
    - **Tekstualne – koriste tekstualne “streamove” za razmjenu podataka – podrazumijevana vrsta datoteka**
    - **Binarne - koriste binarne streamove za razmjenu podataka**

## Tekstualne datoteke

Čuvaju nizove znakova (šampajućih i upravljčkih), na kraju svakog reda znak ‘\n’, a na kraju kompletnog sadržaja znak EOF. Preglediv sadržaj

## Binarne datoteke

Čuvaju nizove binarnih cifara (0 i 1) bez ikakvih oznaka. Nepreglediv sadržaj, ali je kraći zapis za istu količinu podataka u odnosu na tekstualne datoteke

# Rad sa datotekama...



- **Razmjena podataka sa datotekama**
- Za tekstualne datoteke
  - U toku razmjene podataka neophodna je konverzija: iz binarnog u neki drugi format u datoteci i obrnuto, iz nekog drugog formata u binarni.
- **Binarne datoteke**
  - U toku razmjene podataka nije neophodna konverzija jer je i u memoriji i u binarnoj datoteci zastupljen binarni format.

# Rad sa datotekama...

---

- **Podrazumijeva sledeće korake:**
  - Uvodi se u program pokazivač na jednu strukturu tipa FILE
  - Poziva se funkcija za otvaranje datoteke i njen rezultat čuva uvedeni pokazivač, na taj način pokazivač se pridružuje datoteci
  - Za pristup datoteci pozivaju se odgovarajuće funkcije za upis / čitanje
  - Kad više nije potreban pristup, poziva se funkcija za zatvaranje datoteke

# Rad sa datotekama...

- Otvaranje datoteke, **FILE \*fptr;**
- Za početak, potrebna je deklaracija pokazivača
- Funkcija za otvaranje datoteke iz <stdio.h>
- **FILE \*fopen( const char \*ime\_dat, const char \*mod );**



Pokazivač vezan za otvorenu datoteku ili NULL pokazivač, ako datoteka nije otvorena



Pokazivač na string koji čuva ime datoteke



Pokazivač na string koji čuva mod pristupa

# Rad sa datotekama...

---

- Funkcija za zatvaranje datoteke iz <stdio.h>
- **int fclose( FILE \*fptr );**



0 ako je uspješno  
zatvorena datoteka  
EOF (-1) u slučaju  
greške



Pokazivač povezan sa  
datotekom koja se zatvara



# Rad sa datotekama...

---

- Ime datoteke može biti:
  - zadato u programu između znakova navoda i sačuvano u stringu

```
char ime_dat1[ ] = "primer.txt";
```

```
char ime_dat2[ ] = "C:\\biblioteka\\primer.txt";
```



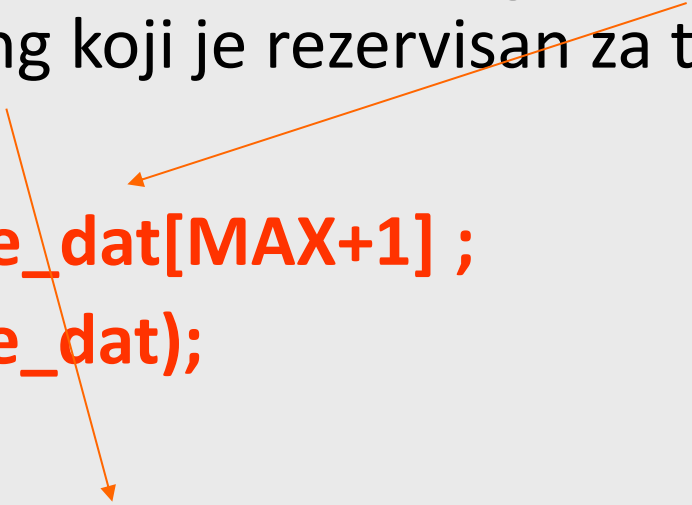
Umesto jednog znaka obrnuta kosa crta, neophodna su dva ova znaka, jer samo jedan znak označava upravljačku sekvencu!!!

# Rad sa datotekama...

---

- **Ime datoteke može biti:**
- Inicijalizovano sa nekog ulaznog uređaja ili iskopirano u string koji je rezervisan za to ime.

```
char ime_dat[MAX+1];  
gets(ime_dat);
```



```
char ime_dat[MAX+1];  
copy(ime_dat, "primer.txt");
```

# Rad sa datotekama...

- Mod otvaranja, ".. **“tekstualna, “.b”-binarna**

| Oznaka      | Značenje   |
|-------------|--|
| <b>“w”</b>  | Otvoranje za upis od početka   |
| <b>“r”</b>  | Otvoranje za čitanje   |
| <b>“a”</b>  | Otvoranje za upis u produžetku postojećeg sadržaja                         |
| <b>“w+”</b> | Otvoranje za upis od početka i za čitanje posle toga                       |
| <b>“r+”</b> | Otvoranje za čitanje i za upis posle toga                                  |
| <b>“a+”</b> | Otvoranje za upis u produžetku postojećeg sadržaja i za čitanje posle toga |
| <b>“wb”</b> | Otvoranje za upis od pocetka   |
| <b>“rb”</b> | Otvoranje za citanje   |
| <b>“ab”</b> | Otvoranje za upis u produžetku postojećeg sadržaja                         |

# Rad sa datotekama...

- Mod otvaranja...

| Oznaka       | Znacenje   |
|--------------|--|
| <b>“wb+”</b> | Otvaranje za upis od početka i za čitanje posle toga                       |
| <b>“rb+”</b> | Otvaranje za čitanje i za upis posle toga                                  |
| <b>“ab+”</b> | Otvaranje za upis u produžetku postojećeg sadržaja i za čitanje posle toga |

# Rad sa datotekama...

---

- **Provjera statusa datoteke nakon otvaranja.**
- Otvaranje toka za razmjenu podataka sa datotekom

```
fptr = fopen("primer.txt", "w") ;  
  
if(fptr == NULL)  
{  
    fprintf(stderr, "Greska otvaranja datoteke");  
    exit(1);  
}  
/*pristup datoteci na odgovarajuci nacin*/
```

# Rad sa datotekama...

---

- Rad sa tekstualnim datotekama
  - **fprintf(), fscanf()**
    - formatiran upis i čitanje
  - **fputc(), fgetc()**
    - upis i čitanje karakter po karakter
  - **fputs(), fgets()**
    - upis i čitanje red po red

# Rad sa datotekama...

- Formatiran upis u datoteku

```
int i;  
FILE *fptr;  
fptr = fopen("dat.txt", "w");  
/*provera*/  
fprintf(fptr, "\nNiz:\n");  
for(i=0; i<10; i++)  
    fprintf(fptr, "%d ", i+1);  
fclose(fptr);
```

uveden  
pokazivač

Otvorena datoteka i  
pokazivač vezan  
za datoteku

Upisan sadržaj u datoteku

Zatvorena datoteka

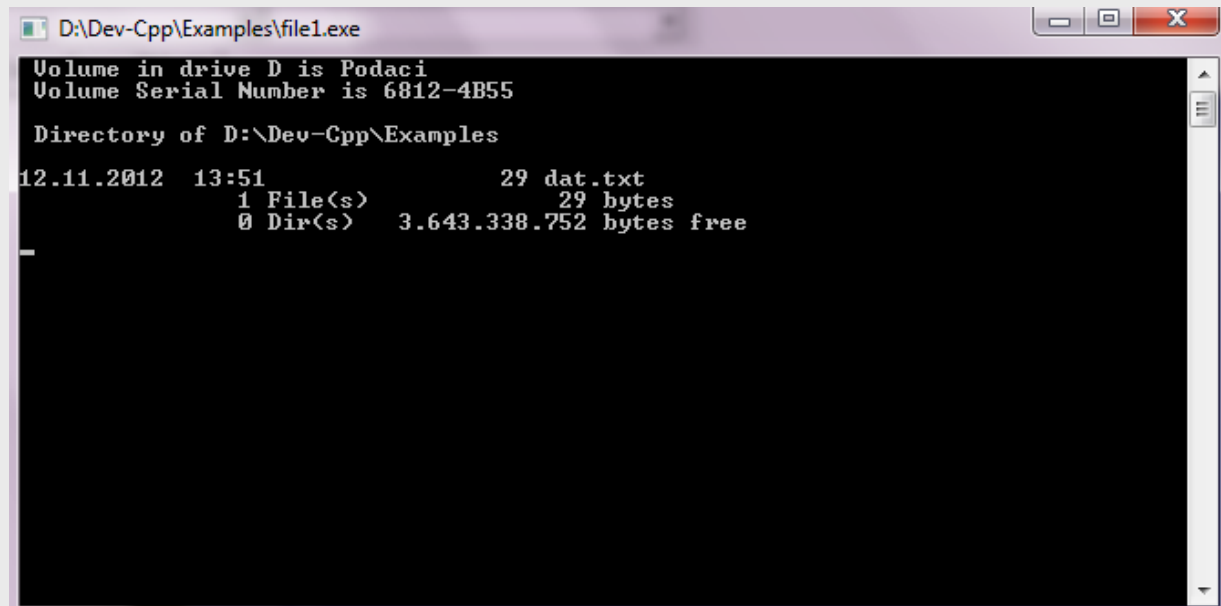
# Rad sa datotekama...

## Primjer:

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

int main()
{
int i;
FILE *fptr;
fptr = fopen("dat.txt", "w") ;
/*provera*/
fprintf(fptr, "\nNiz:\n");
for(i=0; i<10; i++)
fprintf(fptr, "%d ", i+1);
fclose(fptr);
system("dir *.txt");
getchar();

return 0;
}
```



```
D:\Dev-Cpp\Examples\file1.exe
Volume in drive D is Podaci
Volume Serial Number is 6812-4B55

Directory of D:\Dev-Cpp\Examples

12.11.2012  13:51                29 dat.txt
              1 File(s)                29 bytes
              0 Dir(s)  3.643.338.752 bytes free
```



# Rad sa datotekama...

- Formatirano čitanje iz datoteke

```
int i, broj; char bafer[81];  
FILE *fptr;  
fptr = fopen("dat.txt", "r");  
/*provera*/  
fscanf(fptr, "%s", bafer);  
puts(bafer);  
for(i=0; i<10; i++)  
{ fscanf(fptr, "%d ", &broj);  
  printf("%d ", broj); }  
fclose(fptr);
```

uveden  
pokazivač

Otvorena datoteka i  
pokazivač vezan  
za datoteku

Pročitani sadržaj iz datoteku

Zatvorena datoteka

# Rad sa datotekama...

- Čitanje kompletnog sadržaja jedne datoteke i upis u drugu datoteku: karakter po karakter.

```
int c;  
FILE *fptr1, *fptr2;
```

uvedeni  
pokazivači

```
fptr1 = fopen("dat1.txt", "r") ;  
/*provera*/  
fptr2 = fopen("dat2.txt", "w") ;  
/*provera*/
```

Otvorene datoteka i  
Povezivanje pokazivača  
sa datotekama

```
while( (c=fgetc(fptr1)) != EOF)  
    fputc(c, fptr2);  
fclose(fptr1);  
fclose(fptr2);
```

Kopiranje  
kompletnog sadržaja iz  
datoteke u datoteku

Zatvaranje datoteka

# Rad sa datotekama...

- Čitanje kompletnog sadržaja jedne datoteke i upis u drugu datoteku: karakter po karakter.

```
int max=80; char bafer[81];  
FILE *fptr1, *fptr2;
```

→ uvedeni  
pokazivači

```
fptr1 = fopen("dat1.txt", "r") ;  
/*provera*/  
fptr2 = fopen("dat2.txt", "w") ;  
/*provera*/
```

→ Otvorene datoteka i  
povezivanje pokazivača  
sa datotekama

```
while(1)  
{ fgets(bafer, max+1, fptr1);  
  if( strlen(bafer) == 0)  
    break;  
  fputs(bafer, fptr2); }
```

→ Kopiranje redova do  
1. praznog reda iz  
datoteke u datoteku

```
fclose(fptr1); fclose(fptr2);
```

→ Zatvaranje datoteka

# Rad sa datotekama...



- Rad sa binarnim datotekama
- **fwrite()**
  - upis blokova podataka iz memorije u datoteku
- **fread()**
  - čitanje blokova podataka iz datoteke u memoriju

# Rad sa datotekama...

- **Upis u binarnu datoteku, f-ja iz <stdio.h>**
- **int fwrite( void \*bafer, int vel, int broj, FILE \*fptr) ;**

Adresa u memoriji sa koje se upisuju podaci

Veličina jednog bloka u bajtovima i broj blokova

Pokazivač vezan za datoteku u koju se upisuju podaci

Broj uspješno upisanih podataka ili EOF (-1) u slučaju greške

# Rad sa datotekama...

- Upis u binarnu datoteku, f-ja iz <stdio.h>

- **int fread( void \*bafer, int vel, int broj, FILE \*fptr) ;**

Adresa u memoriji sa koje se upisuju podaci

Veličina jednog bloka u bajtovima i broj blokova

Pokazivač vezan za datoteku iz koje se čitaju podaci

Broj uspješno pročitanih podataka ili EOF (-1) u slučaju greške

# Rad sa datotekama...

- **Upis u binarnu datoteku**

```
int i, niz[10];  
FILE *fptr;
```

uveden  
pokazivač

```
fptr = fopen( "dat.bin", "wb" );  
/*provera*/
```

Otvorena datoteka i  
pokazivač vezan  
za datoteku

```
for(i=0; i<10; i++)  
niz[i] = i+1;  
fwrite( niz, sizeof(int), 10, fptr );
```

Upisan sadržaj u  
datoteku

```
fclose(fptr);
```

Zatvorena  
datoteka

# Rad sa datotekama...

- Čitanje iz binarne datoteke

```
int i, niz[10];  
FILE *fptr;
```

uveden  
pokazivač

```
fptr = fopen( "dat.bin", "rb" );  
/*provera*/
```

Otvorena datoteka i  
pokazivač vezan  
za datoteku

```
fread( niz, sizeof(int), 10, fptr );
```

Pročitani sadržaj iz  
datoteke

```
for(i=0; i<10; i++)  
printf( "%d", niz[i] )
```

Odštampani sadržaj na  
ekranu

```
fclose(fptr);
```

Zatvorena  
datoteka



# Rad sa datotekama

---

- **Pozicioniranje unutar sadržaja datoteke**
  - Kada se datoteka otvori za upis, ip=0
  - Kada se upiše 10 bajtova, ip=10
  - Kada se datoteka zatvori i ponovo otvori za čitanje ip=0
  - Kada se pročita 5 bajtova, ip=5
  - Kada se datoteka zatvori i ponovo otvori za upis od kraja
  - postojećeg zapisa, ip=10

# Rad sa datotekama...

---

- Čitanje trenutne pozicije u datoteci, f-ja iz `<stdio.h>`

- **long ftell( FILE \*fptr );**



Trenutna pozicija  
ili (-1) u slučaju greške



Pokazivač vezan za datoteku u  
kojoj se traži pozicija

# Rad sa datotekama...


- Čitanje trenutne pozicije u datoteci

```
int i;  
FILE *fptr;  
  
fptr = fopen("dat.txt", "w") ;  
printf("\n ip=%ld", ftell(fptr)); → ip=0  
  
for(i=0; i<10; i++)  
fprintf(fptr, "%d ", i+1);  
printf("\n ip=%ld", ftell(fptr)); → ip=40  
  
fclose(fptr);
```

# Rad sa datotekama...

---

- Povratak na početak datoteke, f-ja iz `<stdio.h>`.
- **`void rewind( FILE *fptr );`**

 Pokazivač vezan za datoteku u kojoj se realizuje pozicioniranje

# Rad sa datotekama...

- Povratak na početak datoteke, f-ja iz `<stdio.h>`.

```
int i, niz[10]; FILE *fptr;
```

```
fptr = fopen( "dat.bin", "wb" );  
printf( "\n ip=%ld", ftell(fptr));
```

ip=0

```
for(i=0; i<10; i++) niz[i] = i+1;  
fwrite( niz, sizeof(int), 10, fptr );  
printf( "\n ip=%ld", ftell(fptr));
```

ip=40

```
rewind(fptr);
```

```
printf( "\n ip=%ld", ftell(fptr));  
fread( niz, sizeof(int), 10, fptr );  
fclose(fptr);
```

ip=0

Čitanje niza od početka

# Rad sa datotekama...

- Pozicioniranje unutar datoteke
- **void fseek( FILE \*fptr, long pomeraj, int pocetak) ;**

Pokazivač vezan  
za datoteku u kojoj  
se pomjera pozicija

za vrijednost  
pomeraj

od pozicije  
pocetak

- Konstante koje se koriste na mestu 3. argumenta u funkciji fseek, kao oznaka za početnu poziciju
- **SEEK\_SET**, vrijednost 0, pomjeraj od početka datoteke
- **SEEK\_CUR**, vrijednost 1, pomjeraj od trenutne pozicije u datoteci.
- **SEEK\_END**, vrijednost 2, pomjeraj od kraja datoteke

# Rad sa datotekama...

- Pozicioniranje unutar datoteke

```
int i, niz[10]; FILE *fptr;
```

```
fptr = fopen( "dat.bin", "wb" );  
printf( "\n ip=%ld", ftell(fptr));
```



**ip=0**

```
for(i=0; i<10; i++) niz[i] = i+1;  
fwrite( niz, sizeof(int), 10, fptr );  
printf( "\n ip=%ld", ftell(fptr));
```



**ip=40**

```
fseek(fptr, -5*sizeof(int), SEEK_CUR);  
printf( "\n ip=%ld", ftell(fptr));
```



**ip=20**

```
fread( niz, sizeof(int), 5, fptr );  
fclose(fptr);
```



**Čitanje druge polovine  
niza**

# Rad sa datotekama...

---

- Detekcija kraja tekstualne i binarne datoteke, f-ja iz `<stdio.h>`.
- **int feof( FILE \*fptr );**

Pokazivač vezan za datoteku  
čiji se kraj traži

Vrijednost različita od 0 – ako je nađen kraj datoteke. Vrijednost 0 – ako nije nađen kraj datoteke



# Rad sa datotekama...

---

- Detekcija greške u datoteci

- **int ferror( FILE \*fptr ) ;**



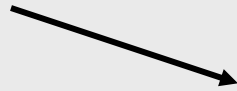
Pokazivač vezan za datoteku  
čiji se kraj traži

Vrijednost različita od 0 – ako je nađena greška. Vrijednost 0 – ako nije nađena greška

# Rad sa datotekama...

---

- Zatvaranje svih otvorenih datoteka, f-ja iz `<stdlib.h>`
- **exit( status ) ;**



Prekid izvršavanja programa i zatvaranje svih otvorenih ulazno / izlaznih tokova

# Rad sa datotekama...



- **Izmena imena datoteka, iz <stdio.h>**
- **int rename(const char \*staro\_ime, const char \*novo\_ime) ;**




Vrijednost 0 – ako je ime datoteke promijenjeno  
Vrijednost (-1) – u slučaju greške

# Rad sa datotekama...

---

- Brisanje datoteke, iz <stdio.h>
- **int remove( const char \*ime\_dat );**



Vrijednost 0 – ako je datoteka izbrisana  
Vrijednost (-1) – u slučaju greške

# Komunikacija sa OS

---

- Prethodno smo se upoznali sa nekim od funkcijama za komunikaciju sa OS.

```
main()
{
.....
exit(1);
.....
return(0);
.....
}
```

# Komunikacija sa OS...

- Funkcija operativnog sistema **system()**

- Podaci programu  
main(.....)

```
{
```

```
.....
```

```
system("dir");
```

```
.....
```

```
exit(1)
```

```
.....
```

```
return(0);
```

```
.....
```

```
}
```

Izlistava stanje direktorijuma iz DOSa

```
C:\ D:\Dev-Cpp\dos_ilu.exe
12.09.2010 23:02 <DIR> mingw32
11.10.2010 15:07          498 Neimenovano1.cpp
11.10.2010 15:07        16.129 Neimenovano1.exe
03.12.2004 18:15        10.807 NEWS.txt
11.10.2010 15:08          609 nzd.c
03.10.2010 23:01          1.138 okvir.c
12.09.2010 23:02 <DIR> Packages
19.02.2005 23:04       380.928 Packman.exe
19.02.2005 23:04       697.941 Packman.map
12.09.2010 23:08          91 proba.c
13.09.2010 10:43        16.458 proba.exe
13.09.2010 10:45          91 proba2.c
13.09.2010 10:45        16.458 proba2.exe
04.12.2010 20:41          936 Projekt1.dev
04.12.2010 21:07       1.452.505 Projekt1.exe
13.09.2010 14:42          85 prvi.c
13.09.2010 14:42        16.458 prvi.exe
20.09.2010 11:30          110 shift_1.c
20.09.2010 11:30       15.840 shift_1.exe
12.09.2010 23:02 <DIR> Templates
12.09.2010 23:02       72.997 uninstall.exe
          34 File(s)        6.374.562 bytes
          13 Dir(s)   13.101.584.384 bytes free
Returned value is: 0.
```

Za LINUX je potrebno npr. **system ("ls |more -al")**

# Komunikacija sa OS (nastavak)

## Poziv \*.exe programa

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

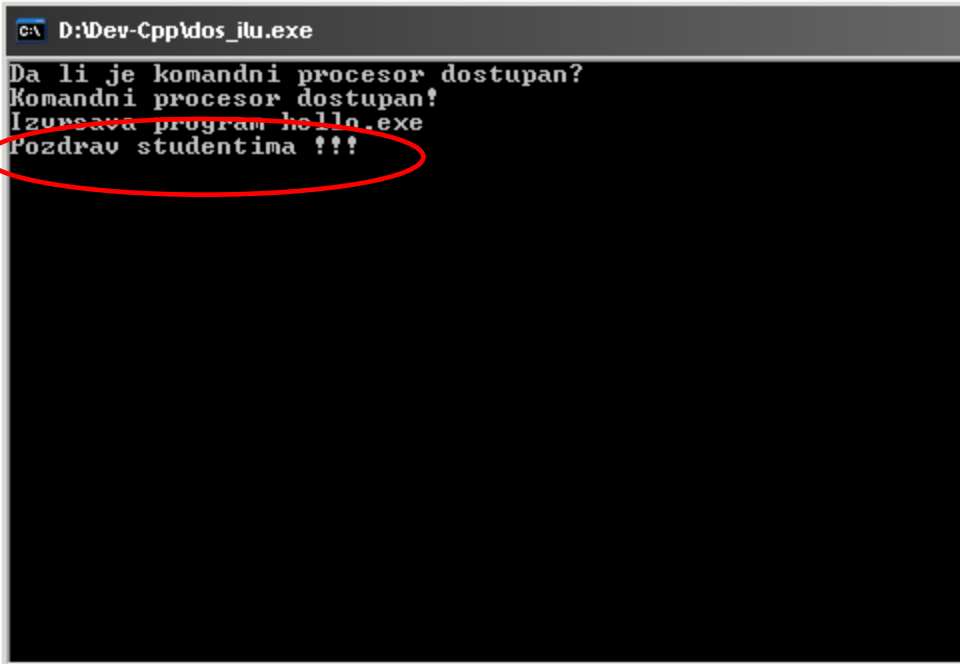
```
int main ()
{
    int i;

    printf ("Da li je komandni procesor dostupan?\n");
    if (system(NULL))
    {
        printf ("Komandni procesor dostupan!\n");
    }
    else
    {
        printf ("Komandni procesor nije dostupan!!\n");
        exit (1);
    }

    /*POZIVA PROGRAM IZ DOSA*/
    printf ("Izvršava program hello.exe \n");
    i=system ("hello.exe");

    printf ("Vracena vrijednost je: %d.\n",i);

    getchar();
    return 0;
}
```



```
C:\ D:\Dev-Cpp\dos_ilu.exe
Da li je komandni procesor dostupan?
Komandni procesor dostupan!
Izvršava program hello.exe
Pozdrav studentima !!!
```

```
#include <stdio.h> //hello.c
```

```
int main ()
{
    printf("Pozdrav studentima !!!");
    getchar();
}
```

Prethodno hello.c treba izkompajlirati i napraviti izvršnu \*.exe verziju

# Podaci sa komandne linije

---

- Oblik glavne funkcije koja prihvata podatke iz komandne linije

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

```
{  
}
```

- Broj podataka iz komandne linije (računa se i ime programa).
- Pokazivač na niz pokazivača na stringove u kojima su sami podaci iz komandne linije.



# Podaci sa komandne linije...

---

- `C:\>ime_programa`
- `argv[0]` - 'i' 'm' 'e' '\_' 'p' 'r' 'o' 'g' 'r' 'a' 'm' 'a' '\0'
- Broj podataka iz komandne linije je `argc = 1`.
- Pokazivač `argv[0]` čuva adresu stringa "ime\_programa".

# Podaci sa komandne linije...

---

- `C:\>ime_programa -opcija`
- `argv[0]` 'i' 'm' 'e' '\_' 'p' 'r' 'o' 'g' 'r' 'a' 'm' 'a' '\0'
- `argv[1]` '-' 'o' 'p' 'c' 'i' 'j' 'a' '\0'
- Broj podataka iz komandne linije: `argc = 2`
- Pokazivač `argv[0]` čuva adresu stringa: "ime\_programa".
- Pokazivač `argv[1]` čuva adresu stringa: "-opcija".

# Podaci sa komandne linije...

- `C:\>echo pozdrav svima`
- `argv[0] 'e' 'c' 'h' 'o' '\0'`
- `argv[1] 'p' 'o' 'z' 'd' 'r' 'a' 'v' '\0'`
- `argv[2] 's' 'v' 'i' 'm' 'a' '\0'`
- Broj podataka iz komandne linije: **`argc = 3`**.
- Pokazivač **`argv[0]`** čuva adresu stringa: “echo”.
- Pokazivač **`argv[1]`** čuva adresu stringa: “pozdrav”.
- Pokazivač **`argv[2]`** čuva adresu stringa: “svima”.

# Podaci sa komandne linije...

- C:\>primjer rijec1 rijec2 rijec3

```
main(int argc, char *argv[ ])      →  argc: 4
{ int i;
if(argc<4)
{ fprintf( stderr, "Greska" ); exit(1); }
printf( "Ime programa: %s", argv[0] ); →  primjer
for( i=1; i<argc; i++ )
printf( "\n%s", argv[i] );          →  rijec1
return 0;                             rijec2
}                                       rijec3
```

# Podaci sa komandne linije...

Komanda je ime .c programa

- **C:\>komanda 10 20 30 40 50**

```
main(int argc, char *argv[ ]) → argc: 6
{ int i, niz[5];
  if(argc<2) /*nije unaprijed poznat broj podataka*/
  { fprintf( stderr, "Greska" ); exit(1); }
  for( i=1; i<argc; i++ )
  niz[i-1] = atoi( argv[i] ); 10
  for( i=0; i<(argc-1); i++ ) 20
  printf( "\n%d", niz[i] ); 30
40
50
  return 0;
}
```

**Izlaz sa komandne linije**

```
D:\TC\BIN>komanda 20 20 30 40 50
20
20
30
40
50
```