

Datoteke

predavač: Nadežda Jakšić

# Programiranje programski jezik C

# Datoteke

- svi podaci sa kojima smo do sada radili u programima su smeštani u operativnu memoriju računara i trajali su najduže koliko i sam program; kada program završi sa radom, vrednost svih promenljivih se gubi; ukoliko želimo na nam neki podaci ostanu sačuvani i posle izvršavanja programa moramo da ih smestimo na hard disk, a jedan od načina da to uradimo je da ih sačuvamo u fajl (datoteku)
- npr. imenik koji ne bi bio praktičan ako bismo pri svakom pokretanju sve podatke ponovo unosili
- datoteka predstavlja sekvencu bajtova i može biti tekstualna ili binarna
- za rad sa datotekama u **C**-u koriste se pokazivači; datoteka je predstavljena posebnim pokazivačem:

**FILE \* fp; //pokazivač se tumači kao memorijska lokacija fajla na disku**

# Funkcije za rad sa fajlovima

- fajl se otvara funkcijom **fopen** koja vraća pokazivač na strukturu **FILE**
- sve ostale funkcije za rad sa fajlovima koriste taj pokazivač umesto imena fajla
- sve funkcije osim **fopen** vraćaju vrednost **int**
- mnoge funkcije za rad sa fajlovima vraćaju konstantu **EOF** (End Of File) ukoliko nastane greška pri izvršavanju ili ako se dođe do kraja fajla

# Funkcije za rad sa fajlovima

- **fopen (imeFajla, mod)** otvaranje fajla
- **fclose (f)** zatvaranje fajla (rezultat 0=uspešno, EOF neuspešno)
- **feof (f)** ispitivanje da li se došlo do kraja fajla (rezultat 0=nije kraj, >0 kraj)
- **fseek (f, kol, odakle)** pozicioniranje  
SEEK\_SET od početka  
SEEK\_CUR od trenutne pozicije  
SEEK\_END od kraja fajla
- **ftell (f)** trenutna pozicija u fajlu (vraća **int** vrednost ili **-1** za grešku)
- **ferror (f)** spitivanje greške u prethodnoj operaciji (0=nema greške)
- **getc (f)** citanje znaka iz fajla (vraća **int** ili **EOF** u slučaju greške)

# Funkcije za rad sa fajlovima

- **putc (znak, f)** upisivanje znaka u fajl  
(isti karakter ili **EOF** u slučaju greške)
- **fgetc (f)** čitanje znaka iz fajla (isto kao getchar)
- **fputc (znak, f)** upisivanje znaka u fajl
- **fgets (f, kol., string)** čitanje stringa iz fajla  
**NULL** ako nema više podataka
- **fputs (string, f)** upisivanje stringa u fajl  
(vraća >0 ili EOF u slučaju greške)
- **fscanf (f,maska,prom.)** čitanje promenljivih iz fajla - vraća broj pročitanih znakova, negativnu vrednost za grešku ili EOF ako nema podataka
- **fprintf (f,maska,prom.)** upisivanje promenljivih u fajl - vraća broj upisanih znakova ili negativnu vrednost u slučaju greške
- **fflush (f)** pražnjenje bafera  
(0=uspešno, EOF neuspešno)

# Otvaranje fajla - mod

**r** //otvaranje postojećeg fajla za čitanje podataka

**w** // otvaranje fajla za pisanje, ako ne postoji fajl kreira se novi, ako postoji , pisanje počinje od početka (briše se postojeći sadržaj)

**a** //otvaranje postojećeg fajla za dodavanje sadržaja na postojeći sadržaj

**r+** //otvaranje postojećeg fajla za čitanje i pisanje

**w+** //otvaranje fajla za čitanje i pisanje, ukoliko fajl postoji, prvo mu se dužina smanjuje na nulu, ako fajl ne postoji, kreira se novi fajl

**a+** //otvaranje fajla za čitanje i pisanje, čitanje se vrši od početka fajla, a pisanje u fajl se nastavlja na postojeći sadržaj, ukoliko fajl ne postoji kreira se novi

**"b"** //ako se iza slova **w**, **r** ili **a** napiše slovo **b** to označava da datoteku treba otvoriti u binarnom modu, inače se datoteka otvara u tekstualnom modu (rb, wb, ab, rb+,wb+, ab+ ili r+b, w+b...)



# Zatvaranje fajla

po završetku rada sa fajlom potrebno je zatvoriti fajl

```
int fclose (FILE *fp );
```

pozivom ove funkcije se prazni bafer koji je služio za ispis u fajl (sav zaostali sadržaj se upisuje u fajl), oslobađa se operativna memorija koja je korišćena za obradu fajla; ukoliko funkcija uspešno završi snimanje u fajl vraća se **0**, inače se vraća definisana konstanta **EOF**

# Uspis u fajl test.txt

```
#include <stdio.h>
#include <conio.h>
int main()
{
    FILE *fp;

    fp = fopen ("test.txt", "w+");
    fprintf (fp, "Ovo je test za funkciju fprintf...\n"); //upisuje u fajl
    fputs ("Ovo je test za funkciju fputs...\n", fp); //upisuje u fajl
    fclose (fp);
    return 0;
}
```



# Čitanje iz fajla test.txt

```
#include <stdio.h>
int main() {

    FILE *fp;
    char buff[255];
    fp=fopen ("test.txt", "r");
    fgets (buff, 255, (FILE*)fp);
    printf ("1: %s\n", buff ); //ispisuje na ekran
    fgets (buff, 255, (FILE*)fp);
    printf ("2: %s\n", buff ); //ispisuje na ekran
    fclose (fp);
    return 0;
}
```

# Otvaranje i zatvaranje fajla

//proverava da li postoji datoteka **druga** u tekućem folderu, otvori datoteku, ispiše na monitoru i zatvori

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
FILE *f;
char *n="druga.txt";
char c;
f=fopen(n,"r");
if (f== NULL ) //ako pokazivač na datoteku druga.txt ne pokazuje ni
{
    printf ("Datoteka <%s> nije uspesno otvorena.\n", n);
    exit(1);
} printf ("Datoteka <%s> je uspesno otvorena\n", n);
```

```
printf ("Tekstualni sadrzaj datoteke <%s>...\n",
n);
while ((c=fgetc(f))!=EOF)
    putchar(c);
printf ("\nKraj sadrzaja datoteke <%s>...\n", n);
fclose(f);
return 0;}
```

# Otvaranje i zatvaranje fajla

//program otvara fajl **mojFajl.txt** za pisanje, upisuje u njega string i zatvara fajl; kod zatvaranja i otvaranja fajla proverava se uspešnost i ispisuje poruka

```
#include <stdio.h>
#include <stdlib.h>
int main()
{ FILE *fp;
fp = fopen ("mojFajl.txt", "w"); //w ako ne postoji kreira fajl
if (fp!=NULL)
    {printf ("Fajl je uspesno otvoren.\n");
    fputs ("Ispis", fp); } //upisivanje sadržaja u fajl
else {printf ("Greska.\n"); }
    if (fclose(fp)!=EOF)
        {printf ("Fajl je uspesno zatvoren.\n"); exit(0); }
    else exit(1); return 0; }
```

# Upis realnih brojeva

- demonstrira se formatiran upis 5 realnih brojeva u tekstualnu datoteku
- korisnik unosi brojeve
- datoteka ne mora prethodno da se kreira - mod je **w**
- funkcija **fscanf()** je pogodna za formatirani unos brojeva, ali nije pogodna za unos stringova i znakova

# Upis realnih brojeva

```
#include <stdlib.h>
#include <stdio.h>
int main() {
FILE *fp;
float data[5];
int i;
char filename[20];
puts("Unesi 5 real.brojeva");
for (i = 0; i < 5; i++)
scanf ("%f", &data[i]);
//pri unosu imena datoteke napisati npr. realni.txt
//isprazni mogući višak znakova iz bafera ulaza
fflush (stdin);
puts("Unesi ime datoteke:");
gets (filename);
```

```
if ((fp = fopen (filename, "w")) == NULL)
{fprintf (stderr, "Greska pri otvaranju datoteke %s.",
filename);
exit(1);}
//ispiši vrednosti u datoteku i na standardni izlaz
for (i = 0; i < 5; i++)
{fprintf (fp, "\n podatak[%d] = %f", i, data[i]);
fprintf (stdout, "\n podatak[%d] = %f", i, data[i]);}
fclose(fp);
printf ("\n Sada procitaj datoteku: %s, nekim
editorom",filename);
return(0);}
```

# Zbir celih brojeva

//datoteka **ulaz.txt** sadrži cele brojeve - izračunati njihov zbir i zapisati u izlaznu datoteku **izlaz.txt** - kreirati datoteke **ulaz** i **izlaz** u tekućem folderu

```
#include <stdio.h>
#include <stdlib.h>
int main() {
FILE *ul,*iz;
int a,zbir=0;
ul=fopen ("ulaz.txt","r");
iz=fopen ("izlaz.txt","w");
if (ul==NULL)
    {fprintf (stderr,"Neuspesno otvaranje ulaz.txt");
    exit (EXIT_FAILURE); }
if (iz==NULL) {
    fprintf (stderr,"Neuspesno otvaranje izlaz.txt");
    exit(EXIT_FAILURE);}
}
```

```
while (fscanf(ul,"%d",&a)!=EOF)
    zbir+=a;
fprintf (iz,"%d",zbir);
fclose (ul);
fclose (iz);
return 0;
}
```

# Append

//program demonstrira dodavanje sadržaja u datoteku (funkcija append) - svaki put kada se program pokrene na izvršenje dodaje se nov sadržaj u datoteku

```
#include <stdio.h>
int main(){
FILE* datoteka;
int godina = 2018;
if ((datoteka=fopen("druga.txt","a"))==NULL) //mod je a
    {fprintf (stderr,"Greska: nisam uspeo da otvorim dat.txt\n");
    return 1;}
fprintf (datoteka,"Zdravo svima\n");
fprintf (datoteka,"Sve najbolje u %d. godini!!!\n",godina);
fclose(datoteka);
return 0;
}
```

# Broj redova

//odrediti broj redova u tekstualnom fajlu čije ime unosi korisnik  
prethodno kreirati datoteku **redovi.txt** i uneti nekoliko redova teksta

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
FILE *ul;
int BrRed=0;
char ch,ime[100];
printf ("Unesite putanju do fajla:");
scanf ("%s",ime);
ul=fopen (ime,"r");
if (ul==NULL)
    {fprintf (stderr,"Neuspesno otvaranje ulaz.txt");
    exit (EXIT_FAILURE);}
while ((ch=fgetc(ul))!=EOF)
    if (ch=='\n') BrRed++;
printf ("Broj redova je %d.",BrRed);
fclose (ul); return 0;}
```



# Automobili

//u datoteci **automobili.txt** nalazi se spisak sa modelima automobila, godištima i cenom - pročitati podatke iz datoteke i na standardnom izlazu ispisati podatke o automobilima čija je cena veća od 10 000 evra

```
#include<stdio.h>
#define DIM 100
typedef struct {
    char model [20];
    int godiste;
    float cena;}AUTO;
int main(){
FILE *ulaz;
AUTO nizAuto[DIM];
int n, i;
ulaz=fopen ("automobili.txt", "r");
if (ulaz==NULL){
    printf ("Ulazna datoteka nije dostupna");
    return 1; }
i=0;
while (!feof (ulaz))
{fscanf (ulaz, "%s %d %f", &nizAuto[i].model, &nizAuto[i].godiste,
&nizAuto[i].cena); i++; }
```

```
n=i;
for (i=0;i<n;i++)
{
if(nizAuto[i].cena>10000)
    printf ("%s %d %f\n", nizAuto[i].model,
nizAuto[i].godiste, nizAuto[i].cena);
}
fclose (ulaz);
return 0;}
```

# Studenti

//u datoteci **studenti.txt** nalaze se podaci o rezultatima ispita: ime i prezime, broj indeksa i broj bodova - pročitati podatke i u datoteku **rezultati.txt** upisati prvih 30% sortiranih podataka o studentima po broju bodova

```
#include<stdio.h>
#define DIM 100
typedef struct
{
    char imePrezime [60];
    char brIndexa [9];
    int bodovi;
}STUDENT;
void razmeni (STUDENT *a, STUDENT *b)
```

```
{STUDENT tmp;
    tmp=*a; *a=*b; *b=tmp;}
void sort (STUDENT niz [ ], int n)
{int i, j;
    for (i=0;i<n-1;i++)
        {for(j=i+1;j<n;j++)
            {if (niz[i].bodovi<niz[j].bodovi)
                razmeni(&niz[i], &niz[j]); }}}}
```

# Studenti

```
int main()
{FILE *ulaz;
FILE *izlaz;
STUDENT niz[DIM];
int n, i;
float procenat;
ulaz=fopen ("studenti.txt", "r");
if (ulaz==NULL)
{printf ("Ulazna datoteka nije dostupna");
return 1;}
izlaz=fopen ("rezultati.txt", "w");
if (izlaz==NULL){
printf ("Izlazna datoteka nije dostupna");
return 1;}
i=0;
while (!feof (ulaz))
{fscanf (ulaz, "%s%s%d", niz[i].imePrezime, niz[i].brIndexa,
&niz[i].bodovi);
i++;}
```

# Studenti

```
n=i;
procenat=n*30/(float)100;
printf ("\nPr %f\n",procenat);
sort (niz,n);
for (i=0;i<procenat;i++)
{
fprintf (izlaz,"%s %s %d\n", niz[i].imePrezime, niz[i].brIndexa,
    niz[i].bodovi);
printf ("%s %s %d\n", niz[i].imePrezime, niz[i].brIndexa, niz[i].bodovi);}
fclose (ulaz);
fclose (izlaz);
return 0;
}
```