

Programiranje 1

Beleške sa vežbi

Smer *Informatika*
Matematički fakultet, Beograd

Jelena Tomašević

Sadržaj

1		5
1.1	Rad sa datotekama	5

1

1

1.1 Rad sa datotekama

1. /* Program demonstrira otvaranje datoteka ("r" - read i "w" - write mod) i osnovne tehnike rada sa datotekama */

```
/* U datoteku se upisuje prvih 10 prirodnih brojeva, a zatim se iz iste datoteke citaju brojevi dok se ne stigne do kraja i ispisuju se na standardni izlaz */
```

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

```
/* Zbog funkcije exit */
```

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

```
main()
```

```
{
```

```
    int i;  
    int br;
```

```
    /* Otvaramo datoteku sa imenom podaci.txt za pisanje */  
    FILE* f = fopen("podaci.txt", "w");
```

```
    /* Ukoliko otvaranje nije uspeo, fopen vraca NULL. U tom slucaju, prijavljujemo gresku i završavamo program */
```

```
    if (f == NULL)
```

```
    {
```

```
        printf("Greska prilikom otvaranja datoteke podaci.txt za pisanje\n");  
        exit(1);
```

```
    }
```

```
    /* Upisujemo u datoteku prvih 10 prirodnih brojeva (svaki u posebnom redu) */
```

```
    for (i = 0; i<10; i++)  
        fprintf(f, "%d\n", i);
```

```
    /* Zatvaramo datoteku */
```

```
    fclose(f);
```

¹Zasnovano na primerima sa sajta <http://www.matf.bg.ac.yu/~filip>

```

/* Otvaramo datoteku sa imenom podaci.txt za citanje */
f = fopen("podaci.txt", "r");

/* Ukoliko otvaranje nije uspjelo, fopen vraca NULL. U tom slucaju,
   prijavljujemo gresku i završavamo program */
if (f == NULL)
{
    printf("Greska prilikom otvaranja datoteke podaci.txt za citanje\n");
    exit(1);
}

/* Citamo brojeve iz datoteke dok ne stignemo do kraja i ispisujemo ih
   na standardni izlaz */

    /* Pokušavamo da procitamo broj */
    while(fscanf(f, "%d", &br) == 1)
        /* Ispisujemo procitani broj */
        printf("Procitano : %d\n", br);

/* Zatvaramo datoteku */
fclose(f);
}

2. /* Program demonstrira "a" - append mod datoteka - nadovezivanje */
#include <stdio.h>

main()
{
    FILE* datoteka;

    /* Otvaramo datoteku za nadovezivanje i proveravamo da li je doslo do greske */
    if ( (datoteka=fopen("dat.txt","a"))==NULL)
    {
        fprintf(stderr,"Greska : nisam uspeo da otvorim dat.txt\n");
        return 1;
    }

    /* Upisujemo sadrzaj u datoteku */
    fprintf(datoteka,"Zdravo svima\n");

    /* Zatvaramo datoteku */
    fclose(datoteka);
}

3. /* Program ilustruje rad sa datotekama. Program kopira
   datoteku ulaz.txt u datoteku izlaz.txt. */
/* Uz svaku liniju se zapisuje i njen broj */
#include <stdio.h>

#define MAX_LINE 256

/* Funkcija getline iz K&R jednostavno realizovana preko funkcije fgets */

```

```
int getline(char s[], int lim)
{
    char* c = fgets(s, lim, stdin);
    return c==NULL ? 0 : strlen(s);
}

main()
{
    char line[MAX_LINE];
    FILE *in, *out;
    int line_num;

    if ((in = fopen("ulaz.txt","r")) == NULL)
    {
        fprintf(stderr, "Neuspesno otvaranje datoteke %s\n", "ulaz.txt");
        return 1;
    }

    if ((out = fopen("izlaz.txt","w")) == NULL)
    {
        fprintf(stderr, "Neuspesno otvaranje datoteke %s\n", "izlaz.txt");
        return 1;
    }

    /* Prepisivanje karakter po karakter je moguće ostvariti preko:
       int c;
       while ((c=fgetc(in)) != EOF)
           putc(c,out);
    */

    line_num = 1;
    /* Citamo liniju po liniju sa ulaza*/
    while (fgets(line, MAX_LINE, in) != NULL)
    {
        /* Ispisujemo broj linije i sadržaj linije na izlaz */
        fprintf(out, "%-3d :\t", line_num++);
        fputs(line, out);
    }

    /* Zatvaramo datoteke */
    fclose(in);
    fclose(out);
}
```

4. /* Citanje niza struktura iz tekstualne datoteke - artikli prodavnice */

```
/* Datoteka cije se ime unosi sa standardnog ulaza sadrzi podatke o
   proizvodima koji se prodaju u okviru odredjene prodavnice.
   Svaki proizvod se odlikuje sledecim podacima :
       bar-kod    - petocifreni pozitivan broj
       ime       - niska karaktera
```

cena - realan broj zaokružen na dve decimale
 pdv - stopa poreza - realan broj zaokružen na dve decimale
 Pretpostavljamo da su podaci u datoteci korektno zadati.

Pretpostavljamo da se u prodavnici ne prodaje više od 1000 različitih artikala.
 Na standardni izlaz ispisati podatke o svim proizvodima koji se prodaju.

```

*/
#include <stdio.h>

/* Maksimalna dužina imena proizvoda */
#define MAX_IME 30

/* Struktura za čuvanje podataka o jednom artiklu */
typedef struct _artikal
{
    int bar_kod;
    char ime[MAX_IME];
    float cena;
    float pdv;
} artikal;

/* Maksimalni broj artikala */
#define MAX_ARTIKALA 1000

/* Niz struktura u kome se čuvaju podaci o artiklima */
artikal artikli[MAX_ARTIKALA];

/* Broj trenutno učitanih artikala */
int br_artikala = 0;

/* Učitava podatke o jednom artiklu iz date datoteke.
   Vraca da li su podaci uspesno procitani */
int učitaj_artikal(FILE* f, artikal* a)
{
    /* Citamo podatke */
    if((fscanf(f, "%d", &(a->bar_kod))==1)
    && (fscanf(f, "%s", a->ime)==1)
    && (fscanf(f, "%f", &(a->cena))==1)
    && (fscanf(f, "%f", &(a->pdv))==1))
        /* Prijavljujemo uspeh */
        return 1;
    else
        /* Prijavljujemo neuspeh. */
        return 0;
}

/* Izračunava ukupnu cenu datog artikla */
float cena(artikal a)
{
    return a.cena*(1+a.pdv);
}

```

```
/* Ispisuje podatke o svim artiklima */
void ispisi_artikle()
{
    int i;
    for (i = 0; i < br_artikala; i++)
        printf("%-5d %-10s %.2f %.2f = %.2f\n",
               artikli[i].bar_kod, artikli[i].ime,
               artikli[i].cena, artikli[i].pdv, cena(artikli[i]));
}

main()
{
    FILE* f;

    /* Ucitavamo ime datoteke */
    char ime_datoteke[256];
    printf("U kojoj datoteci se nalaze podaci o proizvodima: ");
    scanf("%s", ime_datoteke);

    /* Otvaramo datoteku i proveravamo da li smo uspeli */
    if ( (f = fopen(ime_datoteke, "r")) == NULL)
    {
        printf("Greska : datoteka %s ne moze biti otvorena\n",
               ime_datoteke);
    }

    /* Ucitavamo artikle */
    while (ucitaj_artikal(f, &artikli[br_artikala]))
        br_artikala++;

    /* Ispisujemo podatke o svim artiklima */
    ispisi_artikle();

    /* Zatvaramo datoteku */
    fclose(f);
}
}
```

Primer 1 Program ilustruje čitanje etiketa iz neke HTML datoteke.

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

/* Maksimalna duzina etikete */
#define MAX_TAG 100

#define OTVORENA 1
#define ZATVORENA 2
#define GRESKA 0

/* Funkcija ucitava sledecu etiketu
```

```

i smesta njen naziv u niz s duzine max.
Vraca OTVORENA za otvorenu etiketu,
ZATVORENA za zatvorenu etiketu,
odnosno GRESKA inace */
int gettag(FILE *f, char s[], int max)
{
    int c, i;
    int zatvorenost=OTVORENA;

    /* Preskacemo sve do znaka '<' */
    while ((c=fgetc(f))!=EOF && c!='<')
        ;
    /* Nismo naisli na etiketu */
    if (c==EOF)
        return GRESKA;

    /* Proveravamo da li je etiketa zatvorena */
    if ((c=fgetc(f))=='/')
        zatvorenost=ZATVORENA;
    else
        ungetc(c,f);

    /* Citamo etiketu dok nailaze slova
    i smestamo ih u nisku */
    for (i=0; isalpha(c=fgetc(f))
        && i<max-1; s[i++] = c)
        ;
    /* Vracamo poslednji karakter na ulaz
    jer je to bio neki karakter koji nije
    slovo*/
    ungetc(c,f);

    s[i]='\0';

    /* Preskacemo atribut do znaka > */
    while ((c=fgetc(f))!=EOF && c!='>')
        ;

    /* Greska ukoliko nismo naisli na '>' */
    return c=='>' ? zatvorenost : GRESKA;
}

main()
{
    char tag[MAX_TAG];
    int zatvorenost;

    FILE* f;

    /* Ucitavamo ime datoteke */
    char ime_datoteke[256];
    printf("Unesite naziv html dokumenta iz kog se vrsi citanje etiketa: ");

```

```
scanf("%s", ime_datoteke);

/* Otvaramo datoteku i proveravamo da li smo uspeli */
if ( (f = fopen(ime_datoteke, "r")) == NULL)
{
printf("Greska : datoteka %s ne moze biti otvorena\n",
      ime_datoteke);
}

while ((zatvorenost = gettag(f,tag,MAX_TAG))>0)
{
if (zatvorenost==OTVORENA)
printf("Otvoreno : %s\n",tag);
else
printf("Zatvoreno : %s\n",tag);
}

fclose(f);
}
```