

Tema 7

Datoteke

Datoteke

- Sekvenca bajtova sačuvana na disku
 - Životni vek datoteke nije direktno uslovljen trajanjem programa
- Binarne i tekstualne
- Rad isključivo preko pokazivača na tip `FILE`
- Funkcije za rad sa datotekama nalaze se u `stdio.h` biblioteci
- Osnovne funkcije za rad sa datotekama
 - `fopen` - otvaranje datoteke
 - Režimi pristupa
 - `fclose` - zatvaranje datoteke
 - Pokazivač tipa `FILE` se ne postavlja na `NULL` vrednost!

Tekstualne datoteke

- Ektenzija `.txt`
- Sadržaj datoteke je isključivo tekst
- Funkcije za rad sa tekstualnim datotekama
 - Univerzalne funkcije za rad sa svim tipovima podataka
 - `fscanf` - formatirano čitanje sadržaja datoteke
 - `fprintf` - formatirani ispis sadržaja u datoteku
 - Specijalizovane funkcije za rad sa nizovima znakova
 - `fgets` - učitavanje određenog broja znakova iz datoteke u niz karaktera
 - `fputs` - ispis određenog broja znakova iz niza znakova u datoteku

Primer 1

Učitavanje iz i zapis u tekstualnu datoteku.

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

#define MAX_SIZE 30

typedef struct tacka_st
{
    double x;
    double y;
} TACKA;

FILE *safe_fopen(char *, char *, int);
void ucitaj_tacke(FILE *, TACKA *, int *);
void ispisi_koordinate(FILE *, TACKA *, int);
```

Programski jezici i strukture podataka - Tema 7

```
int main(int argc, char **argv)
{
    TACKA tacke[MAX_SIZE];
    int n;
    FILE *ulazna, *izlazna;

    if(argc != 3)
    {
        puts("Primer poziva programa: ./a.out in.txt out.txt");
        exit(EXIT_FAILURE);
    }

    ulazna = safe_fopen(argv[1], "r", 2);
    ucitaj_tacke(ulazna, tacke, &n);
    fclose(ulazna);

    izlazna = safe_fopen(argv[2], "w", 3);
    ispisi_koordinate(izlazna, tacke, n);
    fclose(izlazna);

    return 0;
}
```

Programski jezici i strukture podataka - Tema 7

```
FILE *safe_fopen(char *ime, char *rezim, int kod_greske)
{
    FILE *pf = fopen(ime, rezim);

    if(pf == NULL)
    {
        printf("Datoteku %s nije moguće otvoriti.\n", ime);
        exit(kod_greske);
    }

    return pf;
}

void ucitaj_tacke(FILE *ulazna, TACKA *t, int *pn)
{
    int i = 0;

    while(fscanf(ulazna, "%lf %lf",
                &t[i].x, &t[i].y) != EOF)
    {
        i++;
    }

    *pn = i;
}
```

Programski jezici i strukture podataka - Tema 7

```
}  
  
void ispisi_koordinate(FILE *izlazna, TACKA *t, int n)  
{  
    int i;  
  
    for(i = 0; i < n; i++)  
    {  
        fprintf(izlazna, "(%.2lf, %.2lf)\n", t[i].x, t[i].y);  
    }  
}
```

Primer 1 - dodatna pojašnjenja

- `typedef` za preimenovanje tipa (zarad kraćeg zapisa)
- Funkcija `exit` deo je biblioteke `stdlib.h`
 - Za razliku od `return` okončava program iz bilo koje funkcije
 - Konstante definisane u `stdlib.h`
 - `EXIT_SUCCESS` - uspešno izvršen program
 - `EXIT_FAILURE` - neuspešno izvršen program
- Pomoćna funkcija `safe_fopen`
 - Provera da li je uspešno otvorena datoteka je neophodna
 - U slučaju neuspešno otvorene datoteke, `fopen` vraća `NULL`
 - Izlazi iz programa pomoću funkcije `exit` sa prosleđenim kodom greške (celobrojna vrednost, često navedena u tekstu zadatka)
 - Primer: ne postoji fajl koji se otvara u režimu za čitanje `"r"`

Primer 1 - poziv programa i rezultat

Za poziv programa:

```
./a.out tacke.txt koordinate.txt
```

Gde je sadržaj ulazne datoteke `tacke.txt`:

```
1 2  
2 3  
3 4  
4 5
```

Na osnovu kog je dobijen sadržaj izlazne datoteke `koordinate.txt`:

```
(1.00, 2.00)  
(2.00, 3.00)  
(3.00, 4.00)  
(4.00, 5.00)
```

Zadatak 1

Napisati program koji iz tekstualne datoteke učitava n elemenata strukture tipa `auto_st` koja se sastoji iz sledećih polja:

- Marka automobila (jedna reč, do 20 karaktera)
- Kubikaža (prirodan broj)
- Godište (prirodan broj)

Ime ulazne tekstualne datoteke i kubikaža zadaju se kao argumenti komandne linije. Na osnovu zadatih informacija pronaći najnoviji auto sa kubikažom ne većom od zadate i rezultat upisati u tekstualnu datoteku, koja ima naziv oblika `<marka_automobila>.txt`.

Zadatak 2

Napisati program koji kao argument komandne linije prima jednu reč i celobrojnu vrednost. Šifrirati reč pomoću zadatog broja koristeći [Cezarovo šifriranje](#) i upisati celobrojnu vrednost *i*, bez razmaka, dobijeni rezultat u tekstualni fajl `rec.txt`. U slučaju da program ne može da otvori zadatau izlaznu datoteku, izaći iz programa sa kodom greške 1.

Zadatak 3

Proširiti zadatak 6 ("Igra vešala") iz teme "Funkcije" da čita i dekodira reč iz ulazne datoteke, čije ime je zadato argumentom komandne linije. Doraditi da program koristi dekodiranu reč kao zadatak, umesto string konstante, što je bio slučaj u prethodnoj verziji zadatka.

Uraditi sledeće obrade kategorija grešaka u programu:

- Ukoliko ulazna datoteka ne postoji, izaći iz programa sa kodom greške 1
- U slučaju da pročitani string ne počinje brojem, ispisati poruku o grešci prilikom dekodiranja i izaći iz programa sa kodom greške 2