

[] | */home/joksim/.gem/ruby/2.5.0/gems/asciidoctor-pdf-*

*1.5.0.alpha.16/data/themes/images/finki\_name.png*

## Структурно програмирање

Аудиториски вежби 11

# 1. Датотеки

## 1.1. Потсетување од предавања

- Обработката на датотеки се состои од запишување, вчитување или менување на содржината на датотека зачувана на некој стандарден медиум како хард диск.
- Обработката на датотеки во програмскиот јазик C се прави со помош на структурата `FILE`, дефинирана во `stdio.h`.
- За да се започне со обработка на датотеката, прво мора да се отвори истата со помош на функцијата `fopen()`, која како резултат враќа покажувач кон структура `FILE*`.

### 1.1.1. Отворање на датотека за читање/запишување

Функција за отворање на датотека:

```
FILE* fopen(const char* ime_datoteka, const char* mod);
```

`ime_datoteka` - целосната патека каде е зачувана датотеката која сакаме да ја отвориме, заедно со името на датотеката

`mod` - начин на отворање на датотеката

- Можните начини на отворање на дадена датотека (можни вредности за вториот аргумент `mod` на функцијата `fopen()`) се дадени во продолжение.

Начин	Значење
r	Отвора постоечка датотека само за читање
w	Отвора (создава) нова датотека за запишување (ако датотеката веќе постои - ќе ја пребрише нејзината содржина)
a	Отвора датотека за додавање содржина на крајот од датотеката (ако датотеката не постои - ќе се креира)
r+	Отвора постоечка датотека за читање и запишување од почетокот на датотеката
w+	Отвора (создава) нова датотека за читање и запишување (ако датотеката веќе постои - ќе ја пребрише нејзината содржина)

Начин	Значење
a+	Отвора датотека за читање и за додавање содржина на крајот од датотеката (ако датотеката не постои - ќе се креира)

*Пример за отворање на датотека*

```
FILE* fp = fopen("test.txt", "r");
```

- Се отвора текстуалната датотека "test.txt" во режим за читање.
- За да се отвори датотеката во бинарен мод, се додава буквата b на крајот на аргументот за начинот на отворање (пр. "rb").

### 1.1.2. Затворање на датотека

*Функција за затворање на датотека:*

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

fp - покажувач асоциран на датотеката што сакаме да ја затвориме

*Пример за затворање на датотека*

```
fclose(fp);
```

- По завршување со работа со датотеката, таа треба да се затвори со помош на функцијата `fclose()`
- Со оваа функција се затвора датотеката на која што во моментот е асоциран покажувачот fp што се предава како аргумент на функцијата

### 1.1.3. Читање и запишување од/во датотека

*Функции за читање од датотека:*

```
int fscanf(FILE* fp, "kontrolna niza", lista_na_argumenti);
```

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

```
char* fgets(char* str, int num, FILE* fp);
```

Функции за запишување во датотека:

```
int fprintf(FILE* fp, "kontrolna niza", lista_na_argumenti);
```

```
int fputc(char c, FILE* fp);
```

```
int fputs(const char* str, FILE* fp); ---
```

### 1.2. Задача 1

Да се напише програма која за дадена текстуална датотека `text.txt` ќе го одреди и отпечати на екран односот на самогласките и согласките.

#### *Пример*

Ако датотеката `text.txt` ја има следнава содржина:

```
Zdravo, kako si?  
Eve, dobro sum. A ti?  
I jas dobro.
```

тогаш програмата треба да отпечати:

```
Odnos samoglasiki/soglasiki: 16/19 = 0.84
```

## Решение p11\_1.c

```

#include <stdio.h>

int e_bukva(char c) {
    return (c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z');
}

int e_samoglaska(char c) {
    c = tolower(c);
    switch (c) {
        case 'a':
        case 'e':
        case 'i':
        case 'o':
        case 'u':
            return 1;
        default:
            return 0;
    }
}

int main() {
    char c;
    int soglaski = 0, samoglaski = 0;
    FILE *dat;
    // Отворање на датотека за читање
    if ((dat = fopen("text.txt", "r")) == NULL) {
        printf("Datotekata text.txt ne mozhe da se otvori.\n");
        return -1;
    }
    // Читање знак по знак се' доека не се прочита EndOfFile (EOF)
    while ((c = fgetc(dat)) != EOF) {
        if (e_bukva(c)) {
            if (e_samoglaska(c))
                samoglaski++;
            else
                soglaski++;
        }
    }
    fclose(dat);
    printf("Odnos samoglaski/soglaski: %d/%d = %5.2f\n", samoglaski, soglaski,
        (float) samoglaski / soglaski);
    return 0;
}

```

## 1.3. Задача 2

Да се напише програма која секој ред од дадена текстуална датотека `vleзна.txt` ќе го копира во друга датотека `izlezna.txt`, така што пред секој прочитан ред од датотеката `vleзна.txt` ќе додаде уште еден ред во кој ќе стои бројот на знаци што ги содржи прочитаниот ред. Во секој ред може да има најмногу 80 знаци.

### Пример

Ако датотеката `vleзна.txt` ја има следнава содржина:

```

Jas uham Strukturno Programiranje.
Koga se polaga vtoriot kolokvium?
Ne znam, seushte ne e objaveno na sajtot.

```

тогаш по извршувањето на програмата содржината на датотеката `izleзна.txt` треба да биде следнава:

```
36
Jas ucham Strukturno Programiranje.
34
Koga se polaga vtoriot kolokvium?
41
Ne znam, seushte ne e objaveno na sajtot.
```

### Решение p11\_2.c

```
#include <stdio.h>
#define MAX 81

int main() {
    char linija[MAX], *c;
    FILE *vlezna, *izleзна;
    if ((vlezna = fopen("vlezna.txt", "r")) == NULL) {
        printf("Datotekata %s ne mozhe da se otvori.\n", "vlezna.txt");
        return -1;
    }
    if ((izleзна = fopen("izleзна.txt", "w")) == NULL) {
        printf("Datotekata %s ne mozhe da se otvori.\n", "izleзна.txt");
        return -1;
    }

    while ((fgets(linija, MAX, vlezna)) != NULL) {
        int br = strlen(linija);
        fprintf(izleзна, "%d\n%s", br, linija);
    }
    fclose(vlezna);
    fclose(izleзна);
    return 0;
}
```

## 1.4. Задача 3

Да се напише програма која ќе ги прочита елементите од една матрица сместена во текстуална датотека `matrica1.txt`. Во првиот ред од датотеката се запишани бројот на редици и бројот на колони на матрицата. Секој елемент од матрицата е реален број запишан во посебен ред од датотеката. Потоа матрицата треба да се транспонира и да се запише во нова датотека `matrica2.txt` на истиот начин.

### Пример

Ако датотеката `matrica1.txt` ја има следнава содржина:

## Структурно програмирање

```
3 4  
2.1  
3.2  
4.3  
5.4  
1.1  
2.2  
3.3  
4.4  
6.0  
5.5  
3.9  
1.8
```

тогаш по извршувањето на програмата содржината на датотеката `matrica2.txt` треба да биде следнава:

```
4 3  
2.1  
1.1  
6.0  
3.2  
2.2  
5.5  
4.3  
3.3  
3.9  
5.4  
4.4  
1.8
```



## Решение p11\_3.c

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 100
int main() {
    int i, j, m, n;
    float a[MAX][MAX], b[MAX][MAX];
    FILE *input, *output;
    if ((input = fopen("matrica1.txt", "r")) == NULL) {
        printf("Datotekata matrica1.txt ne se otvora!\n");
        exit(1);
    }
    if (!feof(input))
        fscanf(input, "%d %d", &m, &n);

    if ((m > MAX) || (n > MAX)) {
        printf("Mnogu golema matrica!");
        return (-1);
    }
    for (i = 0; i < m && !feof(input); i++)
        for (j = 0; j < n && !feof(input); j++)
            fscanf(input, "%f", &a[i][j]);
    fclose(input);
    if (i != m || j != n) {
        printf("Nema dovolno podatoci vo datotekata!");
        return (-1);
    }
    for (i = 0; i < m; i++)
        for (j = 0; j < n; j++)
            b[j][i] = a[i][j];
    if ((output = fopen("matrica2.txt", "w")) == NULL) {
        printf("Datotekata matrica2.txt ne se otvora!\n");
        exit(1);
    }
    fprintf(output, "%d %d\n", n, m); /* obratno */

    for (i = 0; i < n; i++)
        for (j = 0; j < m; j++)
            fprintf(output, "%7.2f\n", b[i][j]);
    fclose(output);
    return (0);
}
```

## 1.5. Задача 4

Дадена е текстуална датотека `SP_primer.txt`. Да се напише програма која ќе ја прочита датотеката и на екран ќе го отпечати бројот на редови во кои има повеќе од 10 самогласки, како и вкупниот број на самогласки во датотеката.

### Пример

Ако датотеката `SP_primer.txt` ја има следнава содржина:

```
Zdravo, kako si?
Eve, dobro sum. A ti?
I jas dobro. Kako se tvoite? Ima li neshto novo?
Dobri se i tie. Si kupiv avtomobil.
```

тогаш програмата треба да отпечати:

Vkupno 2 reda imaat povekje od 10 samoglaski.  
Vo datotekata ima vkupno 42 samoglaski.

### Решение p11\_4.c

```
#include <stdio.h>
#include <stdlib.h>
int e_samoglaska(char c) {
    return c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u';
}
int main() {
    int red = 0, vkupno = 0;
    FILE *dat; char c;
    if ((dat = fopen("SP_primer.txt", "r")) == NULL) {
        printf("Datotekata SP_primer.txt ne se otvora");
        exit(-1);
    }
    int samoglaski = 0;
    while ((c = fgetc(dat)) != EOF) {
        if (e_samoglaska(tolower(c))) {
            ++samoglaski;
            ++vkupno;
        }
        if (c == '\n') {
            if (samoglaski > 10) {
                red++;
            }
            samoglaski = 0;
        }
    }
    if (samoglaski > 10) {
        red++;
    }
    printf("Vkupno %d reda imaat povekje od 10 samoglaski\n", red);
    printf("Vo datotekata ima vkupno %d samoglaski.\n", vkupno);
    return 0;
}
```

## 1.6. Задача 5

Да се напише програма која за дадена текстуална датотека `zborovi.txt` ќе ги отпечати на екран сите зборови во кои се појавуваат повеќе од две исти букви (некоја буква се појавува три или повеќе пати). Да не се прави разлика помеѓу мали и големи букви. На крајот треба да се отпечати и бројот на зборови што го задоволуваат условот.

Секој ред во датотеката содржи по еден збор (зборовите се разделени меѓу себе со знак за нов ред). Секој збор е составен само од букви. Максималната должина на зборовите е 20 знаци.

### Пример

Ако датотеката `zborovi.txt` ја има следнава содржина:

```
banana  
jabolko  
Obratnoto  
binarnata  
dekadniot  
Kopakabana
```

тогаш програмата треба да отпечати:

```
banana  
Obratnoto  
binarnata  
Kopakabana  
Vкупно 4 zborovi.
```

### Решение p11\_5.c

```
#include <stdio.h>  
#include <ctype.h>  
#define DOLZINA 21  
  
int ima_poveke_od2isti(char *w) {  
    char *c;  
    int isti;  
    while (*w) {  
        c = w + 1;  
        isti = 1;  
        while (*c) {  
            if (tolower(*w) == tolower(*c))  
                isti++;  
            c++;  
        }  
        if (isti > 2)  
            return 1;  
        w++;  
    }  
    return 0;  
}  
  
int main() {  
    char zbor[DOLZINA];  
    FILE *f;  
    int brzb = 0;  
    if ((f = fopen("zborovi.txt", "r")) == NULL) {  
        printf("Datotekata %s ne se otvora.\n", "zborovi.txt");  
        return -1;  
    }  
    while (fgets(zbor, DOLZINA, f) != NULL) {  
        if (ima_poveke_od2isti(zbor)) {  
            puts(zbor);  
            brzb++;  
        }  
    }  
    printf("\nVкупно %d zborovi.\n", brzb);  
    fclose(f);  
    return 0;  
}
```

## 1.7. Задача 6

Да се напише програма која на екран ќе го отпечати бројот на појавувања на даден збор составен само од цифри (зборот се чита од тастатура) во текстуална датотека со име `dat.txt`.

## Пример

Ако од тастатура се внесе зборот

123

и ако датотеката `dat.txt` ја има следнава содржина:

```
Zdravo 123, kako si?  
Eve 321, dobro sum. A ti?  
I jas dobro. Kako se tvoite 123? Ima li neshto novo? 123  
Dobri se i tie. Si kupiv avtomobil.
```

тогаш програмата треба да отпечати:

```
Zborot 123 se pojavuva 3 pati vo datotekata.
```

## Решение `p11_6.c`

```
#include <stdio.h>  
#include <stdlib.h>  
#include <ctype.h>  
int main() {  
    char c;  
    int brPojavuvanja = 0;  
    FILE *dat;  
    if ((dat = fopen("dat.txt", "r")) == NULL) {  
        printf("Datotekata %s ne se otvora!\n", "dat.txt");  
        exit(-1);  
    }  
    char zbor[50];  
    printf("Vnesete zbor za koj kje se bara brojot na pojavuvanja:");  
    gets(zbor);  
    int i = 0, br = 0;  
    while ((c = fgetc(dat)) != EOF) {  
        if (isdigit(c)) {  
            if (c != zbor[i++]) {  
                if (br == strlen(zbor)) {  
                    brPojavuvanja++;  
                }  
                br = 0;  
                i = 0;  
            } else {  
                br++;  
            }  
        } else {  
            if (br == strlen(zbor)) {  
                brPojavuvanja++;  
            }  
            br = 0;  
            i = 0;  
        }  
    }  
    printf("Zborot %s se pojavuva %d pati vo datotekata\n", zbor,  
        brPojavuvanja);  
    return 0;  
}
```

## 2. Изворен код од примери и задачи

<https://github.com/finki-mk/SP/> Source code ZIP