

## Упражнение №10

### ВХОДНО-ИЗХОДНИ ОПЕРАЦИИ. РАБОТА С ТЕКСТОВИ ФАЙЛОВЕ

*Текстов файл* – представлява последователност от символи, като всеки символ се съхранява в отделен байт. Съдържанието формално може да се раздели на две групи – символи за данни (ASCII, Unicode) и управляващи символи за „нов ред“. Символите за „нов ред“ не се отпечатват на екран и често се представят като последователност от два символа (за DOS и Windows): „връщане в началото на текущия ред“ (CR – Carriage Return) и „преминаване на нов ред“ (LF – Line Feed). Всеки от тях се кодира различно.

*Двоичен файл* – съдържа двоичното представяне на данните. Представлява последователност от байтове, които точно съответстват на байтовете във външното устройство.

Всеки поток, свързан с файл, има структура за управление на файла **FILE**, която е дефинирана в **stdio.h**. *Указателят към файл* е свързан с информация, която описва името, състоянието и текущата позиция във файла.

```
FILE *fp;
```

Основни етапи при работа с файлове:

а) *Отваряне на файл*

```
fp = fopen(char *filename, char *mode);
```

Връща указател **fp** към файл с име **filename** или **NULL** при грешка, където **mode** определя режима на отваряне на файла:

<b>"r"</b>	четене	<b>"+"</b>	актуализация
<b>"w"</b>	запис	<b>"t"</b>	текстов файл
<b>"a"</b>	добавяне	<b>"b"</b>	двоичен файл

б) *Обработка* – обмен на данни (запис, четене, актуализация).

в) *Затваряне на файл*

```
fclose(fp); Връща 0 или EOF при грешка.
```

#### I. Запис и четене на числа от текстов файл.

Пример 1: Да се състави програма, организирана като меню, която създава в текущата директория на проекта текстов файл с име **"Test.txt"** за запис на цели числа, въведени последователно от клавиатурата (за край на въвеждането се използва **Ctrl+Z**), отваря файла за четене, отпечатва на екрана съдържанието му и броя на прочетените цели числа. Четенето и записването на данни във файла да се реализират с отделни функции.

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

FILE *writeFile (char *fname);           // прототип на функция за запис
int readFile(FILE *fp, char *fname);     // прототип на функция за четене

int main()
{
    FILE *text;
    int i, mode, count;
    char *menu[] = {"MENU",
                    "1. Zapis v tekstov fajl",
                    "2. Chetene i pechat na tekstov fajl",
                    "3. Izhod"};

    do
    {
        system("cls");                    // изпълнява командата на ОС „изчистване на екрана“
        for(i=0; i < 4; i++)              // отпечатва менюто
            printf ("\n%s\n", menu[i]);
    }
```

```

printf ("\n\nIzberete rejim [1-3]: ");
scanf("%d", &mode);

switch(mode)
{
case 1:
    text = writeFile("Test.txt");           // обръщение към функцията за запис
    if (text == NULL)
        printf("Ne moje da otvori fajla za zapis.");
    system("pause");                         // изпълнява командата на ОС „пауза“
    break;

case 2:
    count=0;                                // брой на числата
    printf("Cetene....\n");
    count = readFile(text, "Test.txt");     // обръщение към функцията за четене
    if (count >= 0)
        printf("%d celi ciska \n", count);
    else
        printf("Ne moje da otvori fajla za cetene.\n");
    system("pause");                         // изпълнява командата на ОС „пауза“
    break;

case 3:
    break;

default:
    printf("\nGreshen izbor!\n");
    system("pause");                         // изпълнява командата на ОС „пауза“
}
}
while(mode != 3);
return 0;
}

/* Дефиниция на функция за запис на цели числа във файл с име fname. Връща
указател към файл или NULL. */
FILE *writeFile(char *fname)
{
    FILE *fp;
    int number;
    fp = fopen (fname, "w");
    if (!fp) return NULL;
    printf ("Vyvedete celi ciska. Za kraj: Ctrl+Z.\n");
    fflush (stdin);
    while ( scanf("%d", &number) != EOF )
        fprintf(fp, "%d\n", number);
    fclose (fp);
    return (fp);
}

// Дефиниция на функция за четене на цели числа от файл с име fname и указател
към файл fp. Връща броя на прочетените цели числа или -1. */
int readFile(FILE *fp, char *fname)
{
    int number, n=0;
    fp = fopen (fname, "r");
    if (!fp) return -1;
    while (fscanf(fp, "%d", &number) != EOF)
    {
        printf("%d\n", number);
        n++;
    }
    fclose (fp);
    return n;
}

```

## II. Четене и запис в текстов файл символ по символ.

Пример 2: Да се реализират двете функции, като записът и четенето на информацията се осъществява символ по символ.

```
/* Дефиниция на функция за запис на символ по символ във файл с име fname. Връща
указател към файл или NULL. */
FILE *writeFile(char *fname)
{
    FILE *fp;
    int ch;
    fp = fopen (fname, "w");
    if (!fp) return NULL;
    printf ("Vyvedete simvoli. Za kraj: Ctrl+Z.\n");
    fflush (stdin);
    while ( (ch=getchar()) != EOF )
        fputc(ch, fp);
    fclose (fp);
    return (fp);
}

// Дефиниция на функция за четене на символ по символ от файл с име fname и
указател към файл fp. Връща броя на прочетените символи или -1. */
int readFile(FILE *fp, char *fname)
{
    int ch, n=0;
    fp = fopen (fname, "r");
    if (!fp) return -1;
    while ((ch=fgetc(fp)) != EOF)
    {
        putchar(ch);
        n++;
    }
    fclose (fp);
    return n;
}
```

### Задачи:

1. Да се добави код, с който да се определя общия брой на въведените гласни : 'a', 'o', 'u', 'e', 'i'.
2. Да се добави код, с който всички препинателни знаци се заменят с 'x'.

## III. Четене и запис на символни низове в текстов файл.

Пример 3: Да се състави програма, която записва в текстов файл символен низ, въведен от клавиатурата (за край на въвеждането се използва **Ctrl+Z**). Отваря файла за четене и определя броя на въведените думи. Записва във временен текстов файл "tmp.txt" всяка дума на отделен ред, след което съдържанието му се отпечатва на екрана. Четенето, записът и обработката на данните да се реализират с отделни функции.

**Алгоритъм** за определяне на брой думи в един низ и добавяне на откритите думи във временния файл: (Дума – един или повече последователни символи, оградени от разделители или пунктуационни знаци.)

- декларира се временен низ **word**, в които ще се записва текущата дума;
- нулира се флаг за попадение в дума; нулира се броячът на думите;
- отваря се временният текстов файл за добавяне на запис на отделните думи;
- обхожда се низът символ по символ до достигане на '\0' при следните проверки:
- проверява се всеки символ от низа дали не е разделител и пунктуационен знак при **истина** – символът е част от дума и се записва в **word**; флагът става 1;
- при **лъжа**

ако флаг  $\neq 0$  – имаме дума, добавя се в края '\0' и се записва във временния файл;  
увеличава се броячът на думите; нулира се флагът;

Алгоритъмът е реализиран чрез функцията: `int words(char *str, char *outname);`

```

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

FILE *writeFile(char *fname); // прототип на функция за запис
int readFile(FILE *fp, char *fname); // прототип на функция за четене
FILE *words_file(FILE *fp, char *fname, char *outname); // функция за обработка
int words(char *str, char *outname); // функция за определяне на думи в 1 низ

int main()
{
    FILE *text, *output;
    char fname[20];
    char outname[]="tmp.txt"; // име на временен текстов файл
    int i, mode, words;

    printf("Vyvedete ime na fajl:\n");
    scanf("%s", &fname);
    char *menu[] ={"MENU",
                  "1. Zapis v tekstov fajl",
                  "2. Chetene i pechat na tekstow fajl",
                  "3. Pechat na dumi",
                  "4. Izhod"};

do
{
    system("cls");
    for(i=0; i < 5; i++)
        printf ("\n%s\n",menu[i]);
    printf ("\n\nIzberete rejim [1-4]: ");
    scanf("%d", &mode);

    switch(mode)
    {
    case 1:
        text = writeFile(fname); // обръщение към функция за запис
        if (text == NULL)
            printf("Ne moze da otvori fajla za zapis.");
        system("pause");
        break;

    case 2:
        words=0; // брой на думите
        readFile(text, fname); // обръщение към функция за четене
        output = words_file(text,fname, outname); // към функция за обработка
        if (output == NULL)
            printf("Ne moze da otvori fajla.");
        system("pause");
        break;

        case 3:
            readFile(output, outname); // четене и печат на временния файл
            system("pause");
            break;

        case 4:
            break;

    default:
        printf("\nGreshen izbor!\n");
        system("pause");
    }
}
while(mode != 4);
return 0;
}

```

```

FILE *writeFile(char *fname)
{
    FILE *fp;
    char s[81];
    fp = fopen(fname, "w");
    if (!fp) return NULL;
    printf ("Vyvedete nizove, za kraj: Ctrl/Z.\n");
    fflush (stdin);
    while ( fgets(s, 80, stdin ) != NULL)
        fputs (s, fp);
    fclose (fp);
    return fp;
}

int readFile(FILE *fp, char *fname)
{
    char s[81];
    int num_wrd=0;
    fp = fopen (fname, "r");
    if (!fp) return 0;
    while (fgets(s, 80, fp) != NULL)
        fputs(s, stdout);
    fclose (fp);
    return 0;
}

/* функция за обработка - определяне на броя думи и запис на всяка дума на нов
ред във временен текстов файл; връща указател към временния файл или NULL.
Параметри:
    FILE *fp - указател към входния файл за четене,
    char *fname - име на входен файл,
    char *outname - име на временен файл за запис. */
FILE *words_file(FILE *fp, char *fname, char *outname)
{
    FILE *tmp;
    int num_words=0;
    char str[81];

    fp = fopen (fname, "r");
    if (!fp) return NULL;

    tmp = fopen(outname, "w");
    if (!tmp) return NULL;

    while ( fgets(str, 80, fp) != NULL )// прочита се по 1 низ за всяка итерация
        num_words += words(str, outname); // обръщение към функция за брой думи
    printf("\n Broi dumi =%d\n", num_words);
    fclose(tmp);
    fclose(fp);
    return tmp;
}

/* функция за определяне на брой думи в 1 низ, думите се добавят към временния
файл при всяко обръщение; връща брой думи или 0.
Параметри:
    char *str - указател към текущ низ,
    char *outname - име на временен файл за запис. */
int words(char *str, char *outname)
{
    FILE *out;
    int j=0, i=0, num_wrd=0, flag=0;
    char word[80];

```

```

out = fopen(outname, "aw");
if (out == NULL) return 0;

while( *str != '\0')
{
    if( (!ispunct(*str)) && (!isspace(*str)) )
    {
        word[j++]=*str;
        flag=1;
    }
    else
        if(flag)
        {
            word[j]='\0';
            fputs(word,out);
            fprintf(out, "\n");
            num_wrd++;
            flag=0;
            j=0;
        }
    str++;
}
fclose(out);
return num_wrd;
}

```

**Задача:**

1. Да се отпечата на екрана дължините на отделните думи.
2. Да се състави отделна функция, която проверява дали е коректно въведено името на файла – започва с малка латинска буква, не съдържа специални символи, има разширение “.txt”.