

## Текстовые файлы. Стил С

В программе на С++ для доступа к файлам можно воспользоваться стандартными библиотечными функциями С-подмножества, предназначенными для работы с файловой системой С. Напомним, что система ввода-вывода языка С обеспечивает единообразный интерфейс, который независим от физических устройств. В основание этого интерфейса кладётся концепция потока – абстрактного средства связи между пользователем и конкретным физическим устройством. С точки зрения файловой системы поток – это логическое устройство, а физическое устройство – это файл. Существуют всего два вида потоков: текстовый и бинарный.

Текстовый поток представляет собой последовательность символов, которую можно трактовать и как совокупность строк. Строка, как правило, заканчивается символом новой строки (ASCII-код 12).

Бинарный поток представляет собой последовательность байтов, которая однозначно соответствует последовательности байтов, записанной на внешнем запоминающем устройстве. Отметим, что бинарный поток может содержать и дополнительные нулевые байты, например, в случае выравнивания блоков данных при записи их на внешнее запоминающее устройство.

Для того чтобы можно было записать в файл или прочесть из файла, он должен быть предварительно открыт с помощью библиотечной функции *fopen()*. Операция открытия позволяет открыть поток и связать его с указанным файлом. Возвращаемым значением функции *fopen()* является так называемый указатель файла – указатель на структуру *FILE*, которая содержит информацию о файле (например, имя файла, адрес буфера, положение текущего символа в буфере, режим открытия файла, индикатор ошибки при работе с файлом).

Представим объявления типа *FILE* и неполного набора стандартных библиотечных функций С-подмножества для работы с файлами из *<stdio>*:

```
struct _iobuf
```

```
{
```

```
    char* _ptr;
```

```
    int _cnt;
```

```
    char* _base;
```

```
    int _flag;
```

```
    int _file;
```

```
    int _charbuf;
```

```
    int _bufsiz;
```

```
    char* _tmpfname;
```

```
};
```

```
typedef struct _iobuf FILE;
```

```
FILE* fopen(const char* file_name, const char* mode);
```

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

```
int fprintf(FILE* stream, const char* format, ...);
```

```
int feof(FILE* stream);
```

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

Первый аргумент функции *fopen()* – это строка, содержащая имя файла, второй аргумент – это строка для указания режима работы с файлом.

Возможны, например, следующие режимы: *чтение* (*read* – “*r*”), *запись* (*write* – “*w*”) и *добавление* (*append* – “*a*”). Наряду с текстовыми файлами могут быть и бинарные, в этом случае используется режим *бинарный* (*binary* – “*b*”).

При закрытии файла с помощью функции *fclose()* его связь с потоком разрывается. Если файл был открыт для записи, его содержимое записывается на внешнее устройство. Это так называемый процесс *очистки потока* (*flushing*), он гарантирует, что после закрытия файла в потоке не останется никаких “забытых” данных.

Функция форматного вывода в файл *fprintf()* идентична функции форматного вывода *printf()*.

Функция *feof()* распознаёт *конец файла*.

Функция *fgets()* считывает строку из потока, пока не обнаружит символ новой строки или не прочитает *num – 1* символов. Символ новой строки считывается и записывается в результирующую строку. Результирующая строка – это C-строка.

## *Создание пустого и непустого текстовых файлов*

```
/* Programming in C Style */

#include <stdio>

int main()
{
    FILE* pFile;
    pFile = fopen("empty.txt", "w");
    if (!pFile)
    {
        printf("Invalid opening file!\n");
        return -1;
    }
    else
        fclose(pFile);
    pFile = fopen("text.txt", "w");
    if (!pFile)
    {
        printf("Invalid opening file!\n");
        return -1;
    }
    else
    {
        fprintf(pFile, "Hi!\n");
        fprintf(pFile, "So long!\n");
        fclose(pFile);
    }
    return 0;
}
```

*empty.txt 0 KB*

*text.txt 1 KB*

Hi!  
So long!

*Пустой текстовый файл*  
*Исследование механизма доступа*

```
/* Programming in C Style */

#include <stdio>

int main()
{
    const int n = 80;
    char line[n];
    FILE* pFile;
    pFile = fopen("empty.txt", "r");
    if (!pFile)
    {
        printf("Invalid opening file!\n");
        return -1;
    }
    else
        while(!feof(pFile))
            if (!fgets(line, n, pFile))
            {
                printf("Invalid reading file!\n");
                if (feof(pFile)) printf("The End-Of-File is set!\n");
            }
    fclose(pFile);
    return 0;
}
```

*Результат работы программы:*

Invalid opening file!

*Результат работы программы:*

Invalid reading file!  
The End-Of-File is set!

## *Текстовый файл*

### *Исследование механизма доступа*

```
/* Programming in C Style */

#include <stdio>

int main()
{
    const int n = 80;
    char line[n];
    FILE* pFile;
    pFile = fopen("text.txt", "r");
    if (!pFile)
    {
        printf("Invalid opening file!\n");
        return -1;
    }
    else
        while(!feof(pFile))
            if (!fgets(line, n, pFile))
            {
                printf("Invalid reading file!\n");
                break;
            }
            else
                printf(line);
    fclose(pFile);
    return 0;
}
```

#### *Результат работы программы:*

Invalid opening file!

#### *Результат работы программы:*

Hi!  
So long!  
Invalid reading file!

## *Текстовый файл*

### *Исследование механизма доступа*

```
/* Programming in C Style */

#include <stdio>

int main()
{
    const int n = 80;
    char line[n];
    FILE* pFile;
    pFile = fopen("text.txt", "r");
    if (!pFile)
    {
        printf("Invalid opening file!\n");
        return -1;
    }
    else
        while(!feof(pFile))
            if (fgets(line, n, pFile)) printf(line);
    fclose(pFile);
    return 0;
}
```

*Результат работы программы:*

Invalid opening file!

*Результат работы программы:*

Hi!  
So long!

## *Текстовый файл*

### *Исследование механизма доступа*

```
/* Programming in C Style */

#include <stdio>

int main()
{
    const int n = 80;
    char line[n];
    FILE* pFile;
    pFile = fopen("text.txt", "r");
    if (!pFile)
    {
        printf("Invalid opening file!\n");
        return -1;
    }
    else
        while(fgets(line, n, pFile)) printf(line);
    fclose(pFile);
    return 0;
}
```

*Результат работы программы:*

Invalid opening file!

*Результат работы программы:*

Hi!  
So long!