# Файлы, обработка текста

Шокуров Антон В.
shokurov.anton.v@yandex.ru
http://машинноезрение.рф

12 февраля 2017 г.

Версия: 0.12

**Аннотация**

В данной заметке будет показано как взаимодействовать с файлами, сущности относящейся к операционным системам. Последние означает, что речь пойдет не о конструкции языка Си, а об очередных вспомогательных функциях стандартной библиотеки.

Обработка текстовых файлов: открытие, закрытие, ввод и вывод.

Это предварительная версия!

## 1 Файлы

Как осуществить ввод/вывод из/в файл. Самое простое через терминал/консоль используя встроенные возможности консоли (перенаправленный потоков ввода/-вывода), но можно и вручную используя соответствующий вспомогательный код (функции). Необходимо уметь делать ввод/вывод из файла обеими способами. Последний естественно предпочтительный, хоты бы потому, что позволяет обрабатывать нескольких фалов.

### 1.1 Консольный подход

Фактически за его реализацию отвечает сама консоль. Она позволяет перенаправить потоки ввода и вывода в файл. Это означает, что текст программы не изменяется, а вот при вызове функций ввода (scanf) и вывода (printf) данные считываются или записываются в файл автоматически.

**Ввод** Для ввода из файла input.txt в консоли пишется

```
./run << input.txt
```

Подчеркну, что при этом программа не претерпевает изменения, т.е. вывозы scanf автоматически будут считывать данные из отмеченного файла.

**Вывод** Для вывода в файла output.txt в консоли пишется

```
./run >> output.txt
```

Опять же все вызовы printf автоматически будут выводить текст не в консоль, а в соответствующий файл.

## 1.2 Инициирование взаимодействия

Ввод и вывод можно осуществить Сишными вспомогательными функциями. Нам потребуются функции для открытия/закрытия файлов, считывания и записи данных и ещё возможно некоторые вспомогательные.

Для взаимодействия с файлом, его сначала необходимо "открыть". Само по себе открытие также позволяет проверить существование файла.

**Открытие** Для считывания или записи данных файл сначала необходимо его открыть.

```
FILE *in = fopen("input", "rt");//Открыли текстовый (t)
//файл для чтения (r). В переменную in сохранили хендлер
//файла, т.е. ссылку на него.
if( in == NULL )//По хорошему нужно проверить факт
{//успешного открытия файла. Если хендлер равен NULL, то
  //это значит, что файл не открылся и нам нужно что-то
  //с этим сделать, например, завершить программу.
  printf("Error openning file!\n");
  return -1;
}
```

Файл открывается функцией fopen (см. стр. 1), которой в качестве первого аргумента передается имя файла в кавычках, а второго режим открытия.

Имя файла может быть как просто имя ("input.txt"), так и включающий путь к нему. Путь же может быть как относительный("./data/input.txt"), так и абсолютный ("/home/vasya/project/data/input.txt"). Если файл не существует, то файл не откроется. Последние также влечет возврат NULL из функции fopen.

Режим может быть текстовый (необходима буква t, text), так и бинарный (буква t отсутствует, часто желательно указать b, binary). Данные особенности в крайне важны для виндоуз систем. В Ликусе обычно не играют роли. Далее речь пойдет о текстовом режиме. Бинарный будет описан в отдельной заметке.

Помимо этого, файл может быть открыт для чтения (указывается буква r, read) или для записи (указывается буква w, write). Если прав у пользователя нет (например, на запись), то файл не откроется. Последние также влечет возврат NULL из функции.

В данном случае текстовый файл открыт для чтения. Хэндлер нужен для связывания переменой с вснутренними устройствами стандартной библиотеки и операционной системой. Иначе бы код был бы неэффективным.

**Закрытие** Данная действие необходимо для завершения взаимодействия с данным файлом. Файл закрывается вызовом функции fclose с указанием хендлера файла:

```
1 FILE *in = fopen("input", "rt");//Открыли файл.
2 fclose( in );//Закрыли файл.
```

После завершения работы с файлом (считывание и запись) его необходимо обязательно закрыть (дабы операционная система могла освободить часть выделенных под него ресурсов).

## 1.3 Обработка данных

Файл открывается с целью обработки данных находящихся в файле. В файле данные находятся последовательно, т.е. один за другим. Поэтому с ним ассоциируются две операции: последовательное считывание и запись. При выполнении каждой из этих операций происходит автоматический переход к следующему элементу в файле.

**Считывание** Для считывания используется функция fscanf полностью аналогичная scanf. Она отличается только тем, что указывается дополнительный параметр указывающий на ранее открытый файл (хендлер).

```
1 FILE *in = fopen("input", "rt");//открыли файл
2 int d;
3 //Считываем значение d из файла. Первый аргумент (in)
4 fscanf( in, "%d", &d);//указывает на открытый ранее файл.
5 double f;
6 int rr = fscanf( in, "%lf", &f);//как и ранее,
```

```
7 //возвращаемое значение (rr) указывает на количество
8 //успешно считанных чисел.
```

**Запись** Для записи используется функция fprintf полностью аналогичная printf. Она отличается только тем, что указывается дополнительный параметр указывающий на ранее открытый файл (хендлер). Файл должен быть открыт с режимом для записи (w), иначе будет ошибка.

```
1 FILE *out = fopen("output", "wt");//Открыли файл
2 //на запись (w). Файл будет текстовый (t).
3 int d=5;
4 //Записываем значение d в файла. Первый аргумент (out)
5 fprintf( out, "%d", d);//указывает на открытый файл.
6 double f=6.5;
7 fprintf( out, "%f", f);
```

## 1.4 Другие

Еще есть rewind, feof, понятие о pipe, ...