



План лекции

1. Открытие файла
2. Библиотечные функции
3. Пример создания файла

1. Открытие файла

При последовательном доступе обмен информации производится через специальный буфер, резервируемый системой ввода-вывода. Компиляция языка C++ рассматривает ввод-вывод как поток файлов, которые поступают последовательно, байт за байтом. Связь потока с файлом устанавливается при его открытии. Открытие файла осуществляется функцией `open`. Данная функция возвращает указатель на файл. Указатель на файл необходимо объявлять.

FILE * f;

Здесь FILE-имя типа, описанное в стандартном определении `stdio.h`;

f-указатель на файл (логическое имя).

Обращение к функции `fopen` в программе производится так:

f = fopen (физическое имя файла, вид использования файла);

Вид исполнения.

Если файл открывается для записи или дополнения, но ещё не существует, то он создаётся. Открытие существующего файла для записи приводит к уничтожению его старого содержимого. Попытка прочитать несуществующий файл-это ошибка (`fopen` выдаёт указатель со значением `NULL`).

2. Библиотечные функции

Для работы с файлами используются библиотечные функции `fprintf`, `fscanf`, `fgets`, `fputs`. Их применение рассмотрим ниже на примерах. После окончания работы с файлом он должен быть закрыт. Это делается с помощью библиотечной функции `fclose`, например:

`fclose (lst)`

`lst`-указатель на файл;

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

```
main ( )
```

```
{ int i = 150;
```

```
FILE *lst; /*lst-указатель на файл /
```

```
lst= fopen ("prn","w"); /*lst получает адрес  
открытого файла с именем prn,  
предназначенного для записи в него  
информации (символ w);
```

```
prn-стандартное имя устройства печати,  
информация из файла с адресом lst будет  
выводится на печатающее устройство*/
```

```
fprintf(lst,"\n число i = %d\n",i) ; /*первый параметр  
функции fprintf-это указатель на  
соответствующий файл; на печать будет  
выведена строка: число i = 150*/
```

```
fclose (lst);} /*функция fclose закрывает файл
```

Работа с файлами (запись информации в файл на диск)

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

```
main ( )
```

```
{ int i = 150; FILE *rsd;
```

```
Rsd=fopen("D:ZNI.F","r "); /*второй параметр функции  
fopen-теперь "r ",говорящий о чтении  
информации*/
```

```
fscanf(rsd,"%d",&i); /*из файла ZNI.F(диск D:) будет  
прочитано значение i*/
```

```
printf("число i=%d(начальное значение i)\n",i);
```

```
while (fscanf(rsg,"%d",&i)!=EOF)
```

```
printf("число i = %d\n, i"); /*последовательный  
вывод целых чисел из файла; вывод  
прекращается, когда будет достигнут конец  
файла(EOF)*/
```

```
fclose(rsd);
```

Добавим в файл ZNI.F новые данные:

```
/*Работа с файлами (дополнение файла на диск)*/
```

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

```
main ( )
```

```
{ int c;
```

```
FILE * lds;
```

```
lds = fopen("D:ZNI.F","a"); /*Второй параметр функции  
fopen – теперь "a" , говорящий о возможности  
дополнения файла*/
```

```
puts ("введите целое число");
```

```
scanf ("%d",&c);
```

```
fprintf (lds,"%d\n",c); /*файл ZNI.F будет дополнен  
числом*/
```

```
fclose (lds);
```

```
}
```

С началом работы любой программы автоматически открываются три файла, и для них определяются соответствующие указатели.

Первый из них-это файл для стандартного ввода информации с клавиатуры с указателем **stdin**, второй – для стандартного вывода информации на экран дисплея с указателем **stdout**, третий – для стандартного вывода ошибок на экран дисплея с указателем **stderr**.

Объекты `stdin`, `stdout`, `stderr` – константы, а не переменные, и им нельзя что-либо присваивать. Файл с указателем `stderr` обычно используется для хранения различных диагностических сообщений.

Рассмотрим ввод и запись строк в стандартный файл для ошибок:

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

```
main ( )
```

```
{ char str[50]; FILE *FL; FL=fopen ( "FAL.F", "w" );
```

```
puts ( "Введите строку ( до 49 символов)"); scanf ( "%s", str);
```

```
fputs ( str, FL); /* Функция fputs записывает в файл строку str*/
```

```
fprintf ( stderr, "Демонстрация вывода в стандартный файл  
для ошибок "); /* текст в функции fprintf будет выведен на  
экран*/
```

```
fclose ( FL );
```

```
FL=fopen ( "FAL.F", "r"); /* Файл открывается для чтения */
```

```
fgets ( str, 20, FL); /* функция fgets считывает из файла FAL.F  
строку str, максимальная длина которой, строка  
оканчивается символом '\0'*/
```

```
printf ( "str = %s", str);
```

```
fclose ( FL);
```

```
}
```

После запуска программы на выполнение необходимо ввести строку до 49 символов. На экране появится:

Введите строку (до 49 символов)

Введем строку:

Технический университет

Демонстрация вывода в стандартный файл для ошибок

Подчеркнутая строка будет записана в файл `FA.L.F` с помощью функции `fputs`, первый параметр которой (`str`) – идентификатор записываемой строки, а второй (`FL`)– указатель на соответствующий файл.

Рассмотрим пример создания файла и чтения из него информации по определенному условию.
Информация, входящая в файл:

Ф.И.О. студента, группа, размер стипендии.

Требуется получить информацию о студенте, фамилия которого будет введена.

```
#include <stdio.h>
#include <string.h>
main ( )
{ char fio [15], gr [6], fio1 [15];
float st;
int p=1;
FILE *lf;
clrscr ( );
lf=fopen ( "student.dat", "w");
while (p)
{
printf ("введите фамилию студента\n");
```

```
scanf ("%s",&fio);  
fprintf ( lf, "%s", fio);  
printf ("введите группу\n");  
scanf ("%s", &gr);  
fprintf ( lf, "%s",gr);  
printf ("введите стипендию\n");  
scanf ( "%f",&st);  
fprintf ( lf, "%f\n",st);  
printf ("введите значение для р \n");  
scanf ( "%d",&p);  
}
```

```
fclose (lf);  
printf (“введите фамилию для поиска\n” );  
scanf ( “%s”,&fio1);  
lf=fopen (“student.dat”,“r”);  
P=1;  
while (p)  
{  
fscanf (lf,”%s”,&fio);  
if (feof (lf)) break;
```

```
fscanf (lf,"%s",&gr);  
fscanf (lf,"%f",&st);  
if ( strlen (fio)=strlen (fio1))  
if (strncmp(fio,fio1,strlen(fio))=0)  
printf ("%s %s %f\n", fio, gr, st);  
}  
fclose (lf);  
}
```

В начало программы необходимо поместить процедуру `#include < string.h >`, так как в программе будут применены функции, работающие со строковыми переменными.

Для удобства работы в программе сразу за описанием типа переменных применен оператор очистки экрана – `clrscr ()`. Создание файла происходит в цикле `while`.

При создании файла `student.dat` используется признак окончания ввода информации, обозначенный переменной `P`. До тех пор, пока переменная `P` не получит нулевое значение, будет выполняться цикл `while` и, следовательно, ввод информации.

После того, как файл создан и закрыт, в переменную `fiо1` вводится фамилия студента. Далее идет цикл `while`, в котором читается файл и осуществляется поиск введенной фамилии. Рассмотрим подробно операторы, входящие в этот цикл.

Работа функции `fscanf` описана выше. В следующем операторе `if` применена функция `feof`, аргументом которой является указатель на файл. Эта функция определяет, достигнут ли конец файла.

Если конец файла достигнут, то оператор `break` прервет выполнение цикла. Работа двух последующих операторов `fscanf` не требует объяснений. Далее в программе применен оператор `if`. Длина определяется функцией `strlen`.

Если сравниваемые строки оказались одинаковой длины, то выполняется функция `strncmp`, сравнивающая строки. Первыми двумя аргументами этой функции являются сами строки, а третьим аргументом – длина строки. Если строки совпали, то выводится информация о студенте, фамилия которого была введена.

Контрольные вопросы

1. Что такое файл?
2. Какие существуют функции для работы с файлами последовательного доступа?
3. Какие существуют функции для работы с файлами прямого доступа?
4. Как вычисляется значение переменной `offset`?
5. Какая функция применяется для выполнения произвольного доступа?

Литература

1. Подбельский В.В. Язык Си ++: Учебное пособие. - М.: Финансы и статистика, 1995, - 560 с.
2. Страуструп Б. Язык программирования Сг ++. - М.: Радио и связь, 1991. - 352 стр.
3. Собоцинский В.В. Практический курс Turbo Си ++. Основы объектно- ориентированного программирования. - М.: Свет, 1993. - 236 с.
4. Романов В.Ю. Программирование на языке Си ++. Практический подход. - М.: Компьютер, 1993. - 160 с.
5. Уинер Р. Язык турбо Си . - М.: Мир, 1991. - 384 с.
6. Юлин В.А., Булатова И.Р. Приглашение к Си. - Мн.: Высш. Шк., 1990,- 224 с.