Білет №21

**Файлы и потоки. Работа с файлами.**

**1.1. Файлы и потоки.**

                Файлом называют последовательность байтов, хранящихся на внешнем носителе информации. Под доступом к файлу понимают запись и чтение данных из файла. Потоком называется логический интерфейс (программа), который обеспечивает доступ к файлу. Прежде чем использовать поток для доступа к файлу, его необходимо соединить с этим файлом, то есть обеспечить поток информацией о файле. Эта информация хранится в структуре типа FILE. Потому считается, что поток имеет тип FILE\*, то есть является указателем на файл. Когда поток соединяют с файлом, то говорят, что файл открывают. Когда поток отсоединяют от файла, то говорят, что файл закрывают.

                Каждый поток может работать в двух режимах: текстовом и бинарном. Режим работы потока задается при его соединении с файлом.

                В текстовом режиме поток записывает и читает из файла текстовые строки, которые заканчиваются символом ‘\n’ и могут содержать символ ‘\t’. По стандарту поток должен обеспечивать обработку строк длиной не менее 254 символа, включая символ ‘\n’. Стандартом допускается, что при чтении и записи данных текстовым потоком может происходить их преобразование.

                В бинарном режим поток записывает и читает данные из файла в том виде, в котором они хранятся в оперативной памяти.

                В следующих параграфах будут описаны стандартные функции для работы с файлами, используемые в языке С. Прототипы этих функций находятся в заголовочном файле stdio.h.

**1.2. Соединение и отсоединение потока от файла. Перенаправление потока.**

                Для соединения потока с файлом используется функция:

                FILE\*  fopen(const  char\*  filename, const  char\*  mode);

которая открывает файл, имя которого задано параметром filename, в режиме, заданнном параметром mode. В случае успешного завершения функция fopen возвращает указатель на поток, а в случае неудачи – NULL.

                Параметр mode может принимать следующие значения:

                “r” – чтение в текстовом режиме,

                “w” – запись в текстовом режиме,

                “a” – присоединение в текстовом режиме,

                “rb” – чтение в бинарном режиме,

                “wb” – запись в бинарном режиме,

                “ab” – присоединение в бинарном режиме,

                “r+” или “w+” или “a+” – чтение и запись в текстовом режиме,

                “r+b” или “w+b” или “a+b” – чтение и запись в бинарном режиме,

                “rb+” или “wb+” или “ab+” – чтение и запись в бинарном режиме.

                При открытии файла в режимах “r”, “rb”, “r+”, “r+b” его индикатор позиции устанавливается на начало файла. В случае, если открывается несуществующий файл, то функция fopen заканчивается неудачей.

                При открытии файла в режимах “w”, “wb”, “w+”, “w+b” создается новый файл. Если файл с заданным именем существует, то его содержимое стирается, а индикатор позиции устанавливается на начало файла.

                При открытии файла в режимах “a”, “ab”, “a+”, “a+b” создается новый файл. Если файл с заданным именем существует, то он открывается, и индикатор позиции устанавливается на конец файла.

                Следует учитывать, что если текстовый файл открывается в режиме чтения и записи, то базовая операционная система может открыть его в бинарном режиме. Максимальное количество файлов, которые можно открыть одновременно задается переменной FOPEN\_MAX (равно 20 в Visual C++). Максимальная длина имени файла задается переменной FILENAME\_MAX (равно 260 в ОС Windows).

                Для отсоединения потока от файла используется функция:

                int  fclose(FILE\*  stream);

которая закрывает файл, при этом освобождая все буферы потока. При успешном завершении функция возвращает 0, а в случае неудачи – EOF.

                Для перенаправления потока используется функция:

                FILE\*  freopen(const  char\*  filename, const  char\*  mode, FILE\*  stream);

которая закрывает файл, соединенный с потоком stream, и соединяет с этим потоком файл filename в режиме mode. В случае успеха функция возвращает указатель на поток, а в случае неудачи – NULL. Параметр mode принимает те же значения, что и в функции fopen.

                Поясним подробнее разницу между текстовым и бинарным режимами работы потока. Как в текстовом так и в бинарном режиме можно использовать все функции для доступа к файлу. При работе в бинарном режиме поток записывает на диск и считывает с диска точные копии данных, переданные функциями записи данных на диск и требуемые функциями чтения данных с диска соответственно. Работа потока в текстовом режиме отличается тремя моментами от работы потока в бинарном режиме. Во-первых, в этом случае символ CTRL+Z интерпретируется как конец файла. Во-вторых, при записи в текстовый поток из комбинации символов “\r” (carriage return, возврат каретки) и “\n” (new line, новая строка) в файл записывается только символ “\n”, а при чтении из текстового потока символ “\n” преобразуется в комбинацию символов “\r”+”\n”. В-третьих, так как при записи в текстовый поток может происходить преобразование количества и представления символов, то для получения требуемой позиции в файле нужно использовать только функции fgetpos и ftell.

  FILE\* out;                           // выходной поток

                Char[100] w;

cin.getline(w, 100); // введите путь к файлу

                if(!(out = fopen( w, "w")) == NULL)

                {

                          printf("Open file failed.\n");

return 0;

                }

if(!feof(out)) //проверка на пустой файл

{

printf(“file empty.\n”);

}

2. Работа с индикатором конца файла. Структура FILE содержит индикатор конца файла, который устанавливается в ненулевое значение функцией чтения из файла при достижении этой функцией конца файла. Состояние конца файла читается функцией

                int  feof(FILE\*  stream);

которая возвращает ненулевое значение, если индикатор конца файла установлен, в противном случае функция возвращает 0.