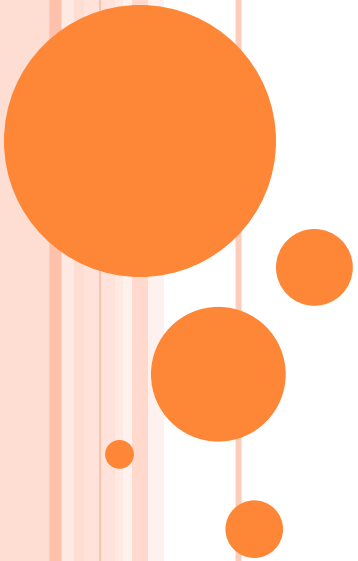
Бинарные файлы



Бинарные файлы

Бинарные (двоичные) файлы представляют собой последовательность данных, структура которых определяется програмно.

Если в файле отсутствует разбиение информации на строки- он бинарный

Режимы доступа

rb - открыть двоичный файл для чтения;
wb - создать двоичный файл для записи;
ab - дополнить двоичный файл;
r+b - открыть двоичный файл для чтения и записи;
w+b - создать двоичный файл для чтения и записи;
a+b - дополнить двоичный файл с предоставлением возможности чтения и записи;

Запись – чтение

- функции посимвольного ввода-вывода
- функции ввода-вывода по блокам

fread()

Функция fread( ) предназначена для чтения блоков данных из потока.
Функция fwrite( ) предназначена для записи блоков данных в потока.

Прототип:
unsigned fread(void *buf, int size, int n, FILE *stream);
unsigned fwrite(void *buf, int size, int n, FILE *stream);

- buf – массив для чтения/записи информации,
- size – размер считываемого блока в байтах,
- n – количество блоков по size байт, считываемых (записываемых) за один раз,
- stream – указатель на файл.

Общее число прочитанных байтов равно произведению n*size.

При успешном завершении функция fread() возвращает число прочитанных элементов данных, при ошибке - 0.

fseek()

Функция fseek( ) позволяет выполнять чтение и запись с произвольным доступом.

Прототип: int fseek(FILE *fp, long count, int access);

fp - указатель на файл,
count - номер байта относительно заданной начальной позиции,
access – начальная позиция.

Переменная access может быть:

0 (SEEK_SET) – отчёт будет производиться от начала файла,
1 (SEEK_CUR) – относительно текущего положения курсора,
2 (SEEK_END) – относительно конца файла.

При успешном завершении возвращается ноль, при ошибке - ненулевое значение

Вызов fseek(fp,0,0) означает, что мы идем в файл, на который ссылается указатель fp, и находим байт, отстоящий на 0 байт от начала, т.е. первый байт.

fseek().Пример

Вывод всего содержимого файла

```
#include <stdio.h>
int main()
{
FILE *fp;
if ((fp = fopen("input.bin”,"rb")) == 0)
   printf("Нельзя открыть%s\n", "input.txt");
else
{
   while (fseek(fp, set++, 0) == 0)
      putchar(getc(fp));
   fclose(fp);
}
return 0;
}
```

fseek().Пример

Считать из первого файла 3 блока по 5 байт и записать эти блоки в другой файл в обратном порядке

```
int main()
{
FILE *in, *out;
char *A = (char *)calloc(5, sizeof (char));
in = fopen("input.bin", "rb");
out = fopen("output.bin", "wb");
fseek(in, 10, SEEK_SET);
fread(A, 5, 1, in); fwrite(A, 5, 1, out);
fseek(in, -10, SEEK_CUR);
fread(A, 5, 1, in); fwrite(A, 5, 1, out);
fseek(in, -10, SEEK_CUR);
fread(A, 5, 1, in); fwrite(A, 5, 1, out);
fclose(in); fclose(out);
}
```

ftell()

Иногда нужно определить текущее положение в файле. Для этого используют функцию ftell()

```
Прототип:
long  ftell(FILE *f);
```

Функция возвращает значение указателя на текущую позицию в файле или –1 в случае ошибки

fgetpos(), fsetpos()

Чтение текущей позиции в файле

```
int fgetpos(FILE *stream, fpos_t *pos);
```

Установка текущей позиции в файле

```
int fsetpos(FILE *stream, const fpos_t *pos);
```

Функции возвращают:
0 – все успешно,
!0 – произошла ошибка.

Пример 1

Дан бинарный файл, содержащий вещественные числа. Найти максимум и минимум, поменять их в файле местами.

Пример 2

Дан бинарный файл, содержащий записи со следующими полями:
ФИО студента (строка 30 символов),
Курс (целое число)
Средний балл (вещественное число).

Переписать файл, упорядочив записи по курсу, а внутри курса – по фамилии.