Chapter 16

# *Advanced I/O Concepts*

©Brooks/Cole, 2004

*O*BJECTIVES

After studying this chapter you will be able to:

❑ Understand and explain the differences between text and binary files.

❑ Understand and explain the three file states (read, write, error).

❑ Read and write binary files.

❑ Use a seek function to set the file position to a specified location.

❑ Describe the basic file update concept.

❑ Write a program to update a sequential file.

©Brooks/Cole, 2004

16.1

Input/Output Classes

©Brooks/Cole, 2004

Figure 16-1   Input/output classes

©Brooks/Cole, 2004

16.2
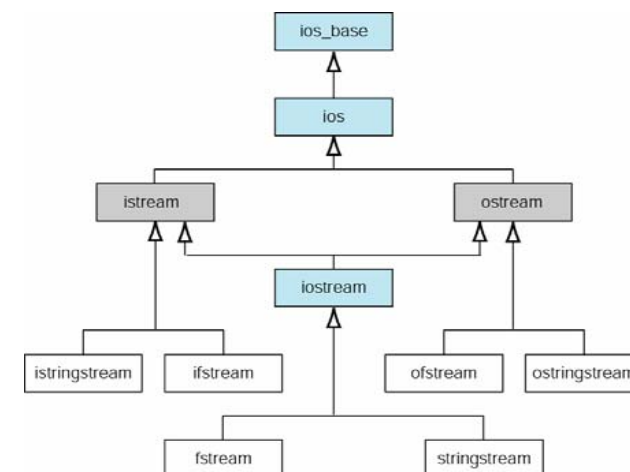
File States

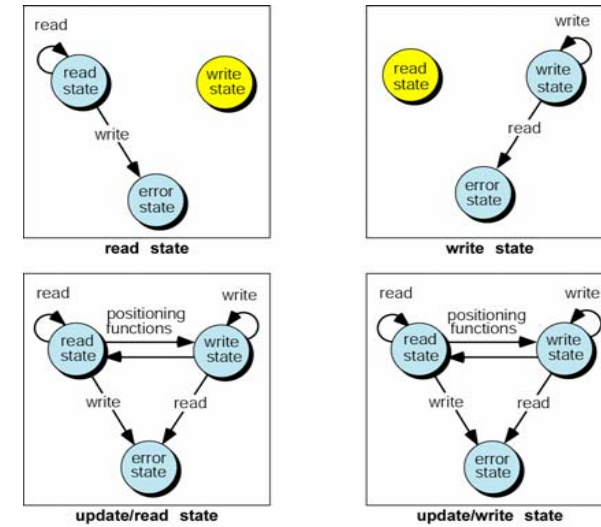---

Figure 16-2   File states

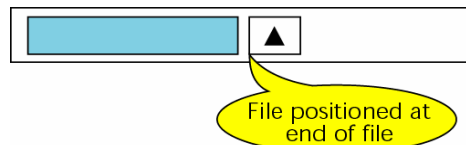**Figure 16-3   Open file in read state**

File positioned at beginning of file

Figure 16-4   File open in write state

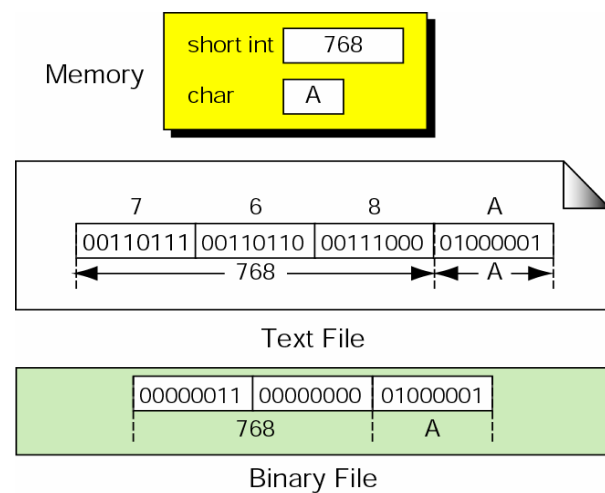
File positioned at beginning of file

Figure 16-5   File open in write/append state

File positioned at end of file

16.3

Text and Binary Files

Figure 16-6   Binary and text files

Memory

short int   768

char   A

7	6	8	A
00110111	00110110	00111000	01000001

768 — A

Text File

00000011	00000000	01000001
768		A

Binary File

16.4

Standard Library Functions for Files

Figure 16-7   Types of standard input/output functions

File
Open/Close

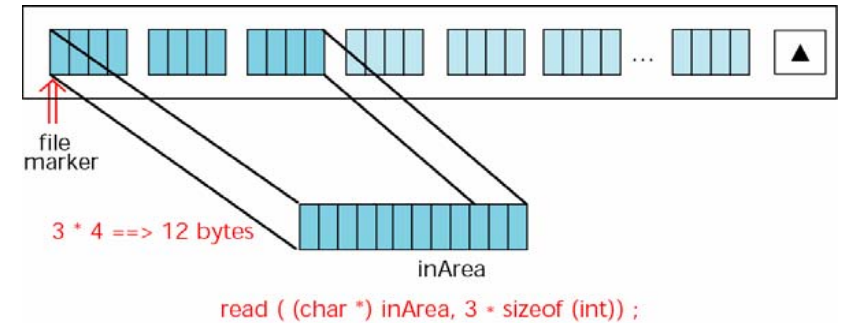Character
Input/Output

Formatted
Input/Output

Categories of
I/O Functions

Line Input

Block
Input/Output

File
Positioning
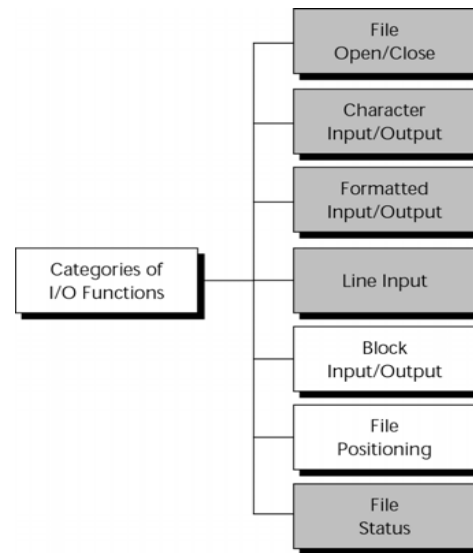
File
Status

Figure 16-8   read operation

file
marker

3 * 4 ==> 12 bytes

inArea

read ((char *) inArea, 3 * sizeof (int)) ;

Figure 16-9   Reading a structure


```
struct STU
{
   char    name[20];
   int     exams[3];
   char    grade;
} // STU
```

Before read

file
marker

oneStudent

?

After read

file
marker

oneStudent
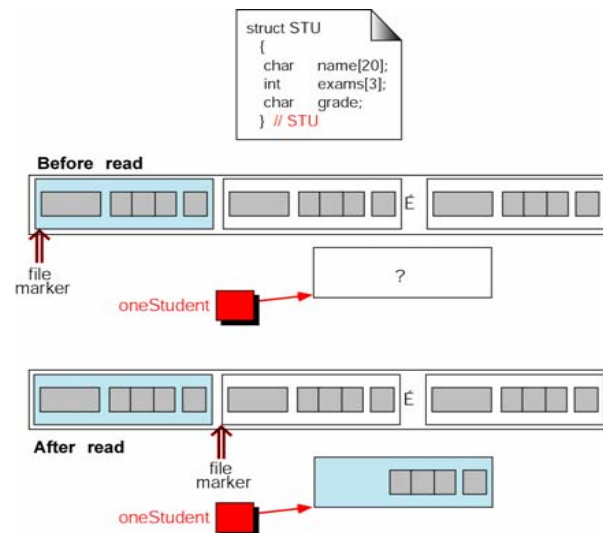
Figure 16-10   write operation

file
marker

4 * 3 ==> 12 bytes

outArea

fsOut.write ((char *) outArea, 3 * sizeof (int)) ;
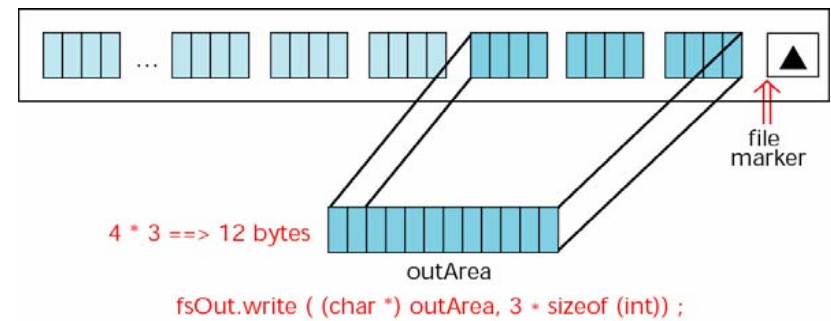
Figure 16-11 Writing a structure

Before write

aStudent

file marker

aStudent

file marker

After write
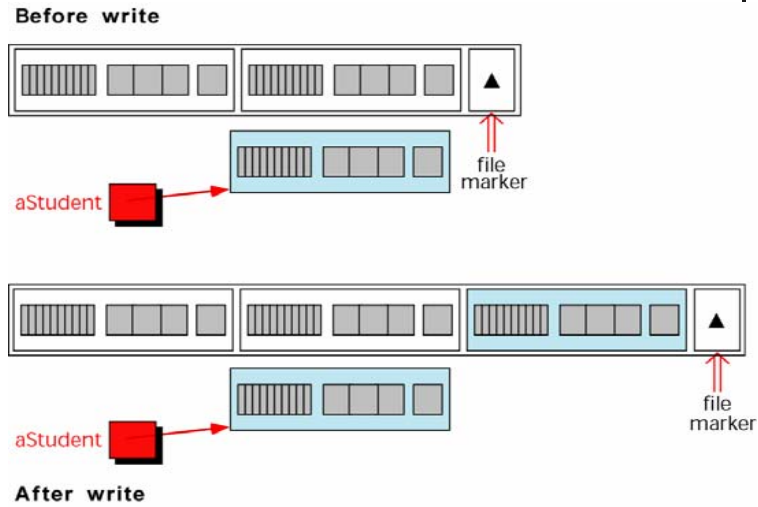
Figure 16-12 tell operation

Beginning of file

Number of bytes

Current file marker location (16)
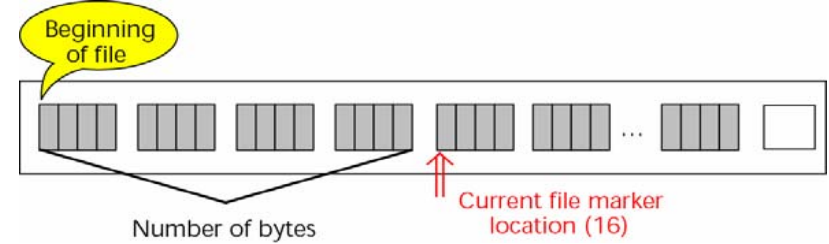
Figure 16-13 seek operation

fsFileName.open (...)

fsFileName.seekg (4 * sizeof (STRUCTURE_TYPE), ios::beg) ;

fsFileName.seekg (-4 * sizeof (STRUCTURE_TYPE), ios::end) ;

fsFileName.seekg (2 * sizeof (STRUCTURE_TYPE), ios::cur) ;
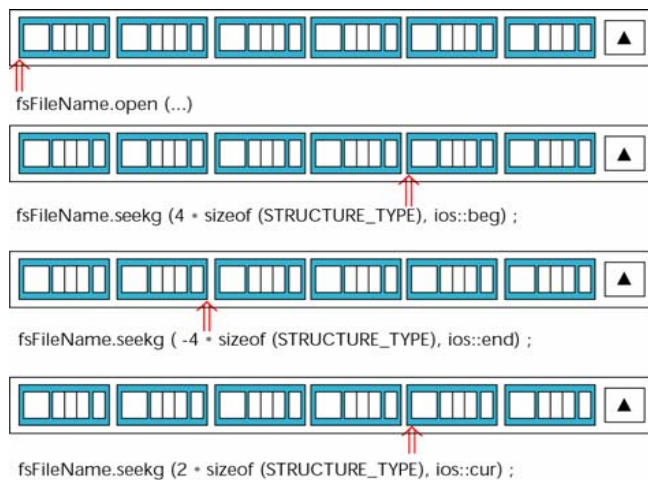
16.5

Converting File Types

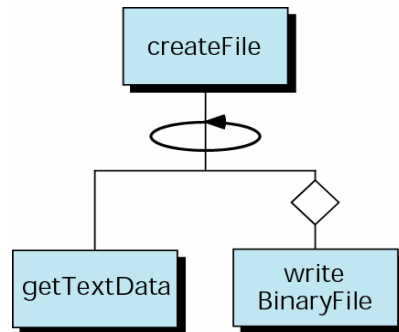**Figure 16-14** Create binary file structure chart
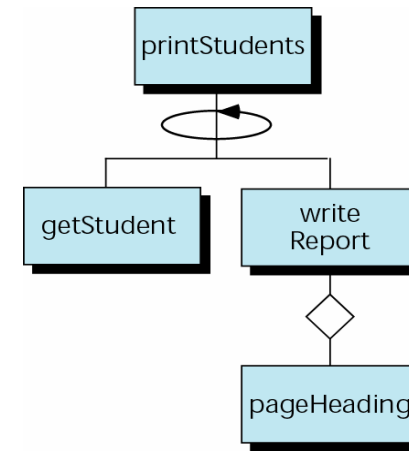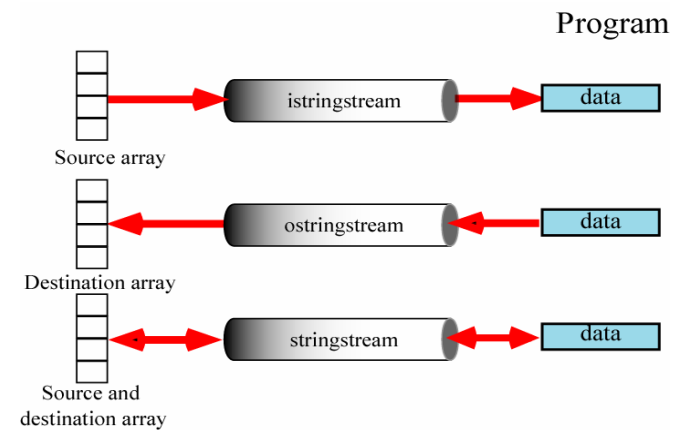
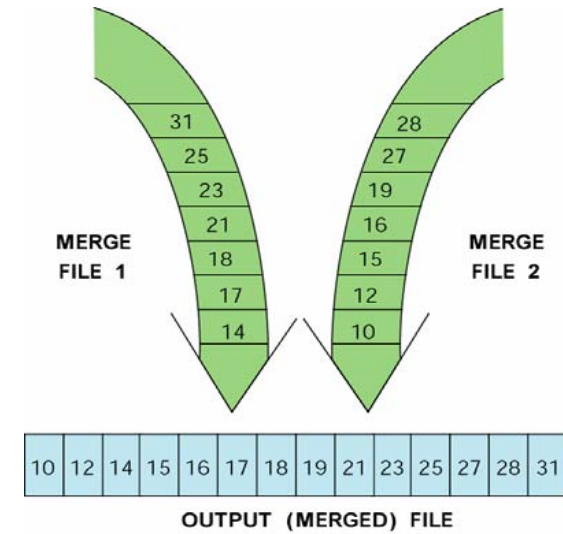**Figure 16-15** Design for print student data

16.6

String Streams

Figure 16-16 Stringstream objects

16.7

File Program Examples

Figure 16-17   File merge concept

16.8

Software Engineering and Programming Style
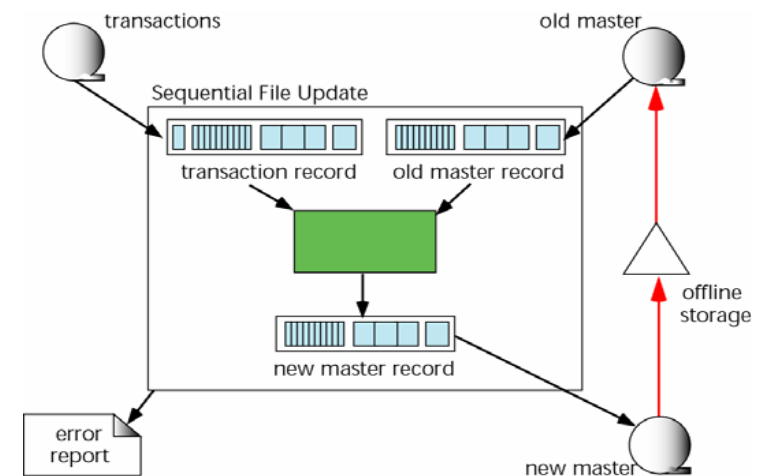

Figure 16-18   Sequential file update environment

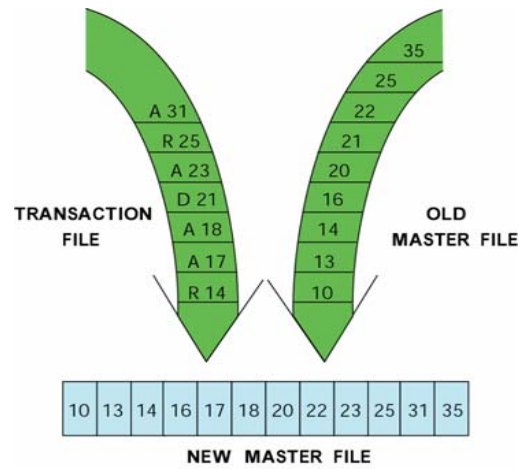Figure 16-19   File updating example

TRANSACTION FILE

OLD MASTER FILE

NEW MASTER FILE

Figure 16-20   Update structure chart