



Introduction to C++ (Season 2)

Unit 7: File Input and Output

第7单元: 出入虽同趣, 所向各有宜— 文件输入输出

Section 08 : Random Access File

第08节: 随机访问文件



Random Access File

❖ *file pointer* (fp):

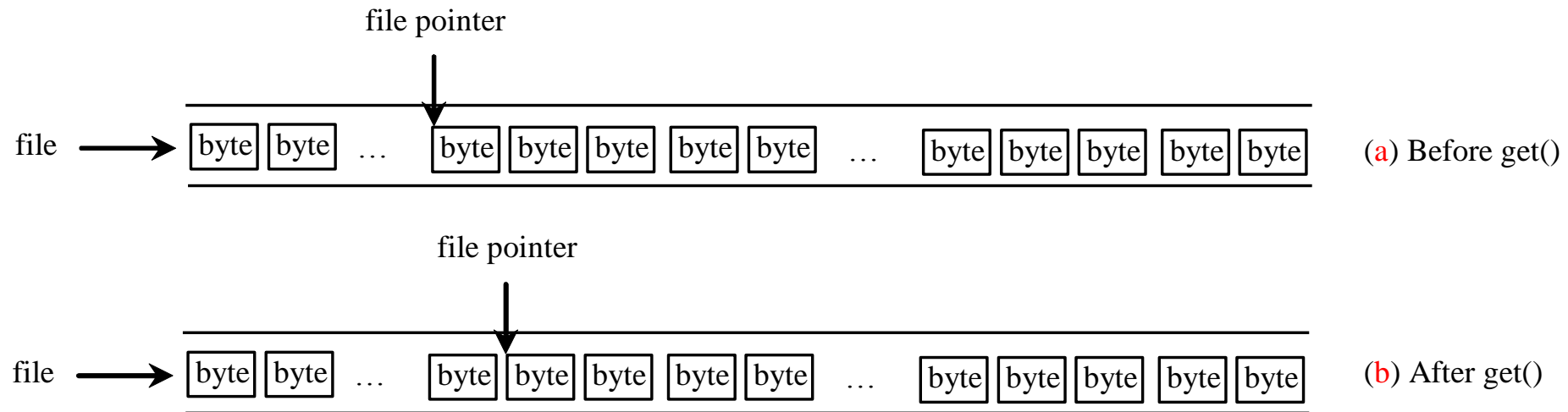
- A file consists of a sequence of bytes.(文件由字节序列构成)
- A special marker that is positioned at one of these bytes. (一个特殊标记指向其中一个字节)

❖ A read or write operation takes place at the location of the file pointer. (读写操作都是从文件指针指向的位置开始)

- When a file is opened, the fp is set at the beginning. (打开文件，fp指向文件头)
- When you read or write data to the file, the file pointer moves forward to the next data item. (读写文件时，文件指针会向前移动)

Random Access File, cont

❖ Example: get() ➔ $fp = fp + 1$



seekp, seekg, tellp, tellg

❖ seek: 移动文件指针

❖ tell: 获取文件指针位置

❖ p: put, 表示操作输出文件中的指针

❖ g: get, 表示操作输入文件中的指针

Seek Base	Description
ios::beg	Calculates the offset from the beginning of the file.
ios::end	Calculates the offset from the end of the file.
ios::cur	Calculates the offset from the current file pointer.

Statement	Description
seekg(100L, ios::beg);	Moves the file pointer to the 100 th byte from the beginning of the file.
seekg(-100L, ios::end);	Moves the file pointer to the 100 th byte backward from the end of the file.
seekp(42L, ios::cur);	Moves the file pointer to the 42 nd byte forward from the current file pointer.
seekp(-42L, ios::cur);	Moves the file pointer to the 42 nd byte backward from the current file pointer.
seekp(100L);	Moves the file pointer to the 100 th byte in the file.

Random Access File Example

❖ Example : demonstrates how to access file randomly.

- first stores 10 student objects into the file (先在文件中存10个学生对象)
- then retrieves the 3rd student from the file.(再从文件中读取第3个学生对象)

Updating Files (更新文件)

```
//.....  
int main() {  
    fstream binaryio; // Create stream object  
  
    // Open file for input and output  
    binaryio.open("object1.dat", ios::in | ios::out  
                  | ios::binary);  
  
    Student student1;  
    binaryio.seekg(2 * sizeof(Student));  
    binaryio.read(reinterpret_cast<char*>  
                  (&student1), sizeof(Student));  
    displayStudent(student1);  
  
    student1.setLastName("Peterson");  
    binaryio.seekp(2 * sizeof(Student));  
    binaryio.write(reinterpret_cast<char*>  
                   (&student1), sizeof(Student));  
}
```

```
Student student2;  
binaryio.seekg(2 * sizeof(Student));  
binaryio.read(reinterpret_cast<char*>  
              (&student2), sizeof(Student));  
displayStudent(student2);  
  
binaryio.close();  
  
return 0;  
}
```