2016夏季学期C++课程大作业报告

马玉坤-1150310618

2016年7月30日

目录

1 前言

五门夏季选修课加其他的琐事，现在才写完大作业。非常感谢又萌又强的三位助教，让我在C++课上甚至课下都学到了很多东西。尽管之前用C++写过不少代码，但是就编程知识，特别是面向对象的思想和方法来讲，这三个星期学到的新东西非常非常多。另外，由于代码逻辑很直（一点都不绕），所以加的注释并不多（其实是因为懒），求助教谅解。

2 主要设计

2.1 必须要说的

为了节约空间（删除一大堆大小为1M的文件后，添加一大堆2M的文件时，无法充分利用之前删除的文件所占的数据空间），我仿照文件系统中的“单元空间”，将数据区域分成若干个64KB大小的“格子”。每个“格子”中，有8B的空间存储下一个“格子”的位置，这样就形成了一个链表。从数据区域读某个文件时，实际上就是将一个链表读入内存。写文件类似。同时，setHeader还要记录空白区域的首地址，这样当添加文件时也更为方便。

其余的实现与段艺学长的指导书中的实现大致相同。

2.2 FileSet

2.2.1 属性

```
1     // 归档文件的文件指针
2     FILE* archive;
3
4     // 指向SetHeader的指针
5     SetHeader* header;
6
7     // 存储FileTag数组
8     FileTag** fileTags;
9
10    // 支持的最大文件数量
11    unsigned int maxFileNumber;
```

2.2.2 方法

```
1
2     // 指定大小和位置，将data写入归档文件
3     inline void writeToArchive(byte* data, unsigned int len, long long pos);
4
5     // 将data写入归档文件的文件区中空闲的区域
```

```
    inline long long writeFileToArchive(byte* data, unsigned int len);

    // 从归档文件的文件区中删除起始位置为pos的文件
    inline void deleteFileFromArchive(long long pos);

    // 从归档文件的文件区中读入起始位置为pos，大小为len的文件
    inline byte* readFileFromArchive(long long pos, unsigned int len);

    // 计算整个checkSum的md5值
    inline byte* calcCheckSum();

    // 添加文件
    bool addFile(char* filePath);

    // 删除文件
    bool deleteFile(char* fileName);

    // 获取文件
    bool fetchFile(char* fileName, char* filePath);

    // 打印文件
    void printFileList();
```

2.3 SetHeader

2.3.1 属性

```

    // 文件的md5值
    byte checkSum[16];

    // 最大文件数量
    unsigned int maxFileNumber;

    // 当前文件数量
    unsigned int currentFileNumber;

    // 数据区域中空白区域的首地址
    long long emptyPosition;

    // 文件标记
    char setMark[9];
```

2.3.2 方法

```
1    // 返回一个SetHeader序列化为unsigned char数组后的结果
2    byte* toBytes();
```

2.4 FileBlock

FileBlock的作用是存储数据区域中一个“小格”表示的链表结点。

2.4.1 属性

```
1    // 所在链表中下一个“小格”的位置
2    long long nextPosition;
3
4    // 当前小格中，存储的数据的大小，最大为64 * 1024 - 12
5    unsigned int size;
6
7    // 当前小格存储的数据
8    byte data[64*1024 - 12];
```

2.4.2 方法

```
1    // 返回将一个FileBlock对象序列化为byte数组后的结果
2    byte* toBytes();
```

2.5 FileTag

2.5.1 属性

```
1    // 小文件数据链表中第一项的地址
2    long long filePosition;
3
4    // 小文件的大小
5    unsigned int fileSize;
6
7    // 当前小文件是否被删除（0代表未删除）
8    byte deleted;
9
10   // 文件名
11   char fileName[256];
```

2.5.2 方法

```
1    // 返回将一个FileTag对象序列化后的结果
2    byte* toBytes();
```