

# C++大作业实验报告

文灏洋 1150310417

计算机科学与技术 1536101(1503104) 班

## 目录

<b>1</b>	<b>实验内容</b>	<b>2</b>
<b>2</b>	<b>实验设计</b>	<b>2</b>
2.1	归档文件结构 . . . . .	2
2.2	类的功能设计 . . . . .	3
2.3	归档功能实现 . . . . .	3
<b>3</b>	<b>其他实现细节若干</b>	<b>4</b>
3.1	关于Convert . . . . .	4
3.2	关于序列化字符串 . . . . .	4
3.3	关于文件状态 . . . . .	4
3.4	关于内存泄露 . . . . .	5
3.5	关于SHA1 . . . . .	5
<b>4</b>	<b>最后的话</b>	<b>5</b>

## 1 实验内容

1. 实现自定的二进制归档文件（小文件集合）存储，其中归档文件包括
  - 文件头（用于记录归档文件的基本属性和内部所归档小文件的索引信息等）；
  - 数据区域（用于存储实际小文件数据，可以自己制定存储方案）。
2. 实现归档文件的初始化、打开、关闭操作。
3. 实现归档功能
  - 添加文件进入归档文件
  - 从归档文件中提取所需文件（根据文件名）
  - 从归档文件中删除指定文件
  - 列出归档文件中现已归档的文件

## 2 实验设计

### 2.1 归档文件结构

归档文件主要由两部分构成，第一部分为文件头，第二部分为实际文件数据存储区域。

#### 文件头

文件头主要包含归档文件的总体信息及归档文件中包含的小文件的具体信息，并且在设计时需要满足可以按照固定模式读取，否则实现打开归档文件时会存在问题。

具体设计时，我主要参照了段艺助教实验指导书内的设计结构，即由一个文件标识（4字节字符数组），归档文件大小（8字节`_int64`），最大文件数容纳量（4字节`int`），SHA1校验码（16字节字符数组），以及最大文件数容纳量个小文件信息标签组成。文件头类的设计也及上述内容。

小文件信息标签包含小文件名（256字节字符数组），文件状态（4字节`int`），文件大小（4字节`int`），小文件头在实际数据储存区距离归档文件头的偏移量（`_int64`），以及SHA1校验码（16位字符数组）。小文件信息标签类的设计也即上述内容。

#### 实际文件数据存储区域

实际文件数据存储区域采用线性表顺序储存，为了防止存储区域重叠，对于已经确定的小文件信息标签里的偏移量，无论这个单元内的文件被添加或删除，这个偏移量始终不变。

## 归档文件类结构

归档文件类包含一个文件指针，和一个文件头类指针。文件指针指向打开的可读写的归档文件，文件头类指针指向创建的该归档文件的文件头类。

## 2.2 类的功能设计

### 小文件信息标签类

首先允许对于所有成员变量的读取和修改。此外还要求对于一个小文件标签有一个序列化函数（即将类中的小文件标签转化为归档文件形式的字符串），同样地，也需要一个根据归档文件形式字符串构造小文件信息标签实例的构造函数。

### 文件头类

同样也是所有成员变量的读取和修改，序列化与反序列化。对于归档文件创建操作，我们需要有一个根据最大归档文件数确立的文件头的构造函数。此外我还设计了一个refresh操作来就当前小文件标签类数组的情况更新文件头类信息的函数，以及得到序列化后字符数组长度的函数。

## 2.3 归档功能实现

### 创建归档文件

这个就是根据最大文件容纳量来创建好文件头类的信息以及开辟好文件信息标签数组。注意初始化的清0。

### 打开归档文件

即利用反序列化构造文件头类即文件信息标签类。在构造之前使用SHA1检验一次校验码。

### 添加文件

主要问题在于找到合适的文件信息标签数组位置。实现时我们认为一个文件信息标签实例合适，其标签状态为0，或者标签状态为2且文件大小小于添加文件大小。

这么搞其实是有点不合适的，因为的偏移量已经固定了，假设对一个实例反复添加和删除，那么其文件大小肯定越来越小，但是我可以利用的空间是不变的。解决方案就是在文件信息标签类里加一个可利用空间长度变量。但是我在意识到这一点时文件信息标签类已经写完了，重新更改代价太高，于是就放弃了。

还有，将一个标签状态为0的实例填信息以后，归档文件的长度会要发生改变，需要同时更新。

对于每一次添加文件操作，都需要在最后refresh文件头信息。

## 删除文件

即利用文件名找到对应的实例，将标签信息更改即可。

## 根据文件名提取文件

也即利用文件名找到对应的实例，在输出到文件前还需检验一遍文件的SHA1值与实例中存的校验码是否匹配。

## 输出列表

直接做就可以了。

## 关闭归档文件

在实际文件关闭之前还是需要先refresh一次，然后对文件头序列化，写入归档文件后才能实际关闭。

# 3 其他实现细节若干

## 3.1 关于Convert

Convert.hpp/Convert.cpp内有用于实现序列化与反序列化的long2bytes, bytes2long, int2bytes, bytes2int函数；用于截取子串的strsub函数，以及和全局常量constMark = " \_HIT"。出现这样的不完全面向对象的设计是因为使用的是字符数组而不是封装好的字符串类，而有些地方有需要频繁的使用具有上述功能的代码段。

## 3.2 关于序列化字符串

这个字符串中由于存在大量0，不能使用strlen函数得到字符串长度，所以需要设计能返回字符串长度的函数。

## 3.3 关于文件状态

文件状态为0是表示该空间未被访问，所以可以由顺序表前一个文件的偏移量与大小计算出当前文件的偏移量，文件状态为1是表示该空间当前正被占用，为2表示该空间已被删除。

设计删除状态主要是为了实现方便以及可以直接标记删除状态而不实际在文件中对一段区域清0，减小了文件操作次数。

### 3.4 关于内存泄露

由于没有涉及到复杂的拷贝过程，所以在拷贝过程中发生的不匹配或者内存泄露可以不用考虑。但是在最开始实现代码的时候没有注意到另一个内存泄露的问题：对于所有返回值为字符数组指针的函数，若返回的是在函数内由new操作创建的字符串，没有在返回并使用后将其空间释放。另外malloc对应free，new对应delete，这两对操作在实现时弄混过。

### 3.5 关于SHA1

SHA1的类并不是自己写的，因为我觉得这大概不是课程要求的内容...

## 4 最后的话

零零总总算上去写了两个整天，代码量不算SHA1也有15 KB左右，这基本上算是自己第一次实现一个偏工程的项目。其中有各种乱七八糟的实现细节又或者是经验不足导致的问题感觉挺糟心的。不过还是体会到了一个工程从设计到实现的一整套过程，还有为了方便日后维护在写代码的时候需要注意的一些问题和技巧等等。感觉收获还是挺大的。

最后谢谢老师和助教们的帮助！