



# 实验报告

院 系 数据科学与计算机学院      班 级 18 计科 8 班      学 号 18340215      姓 名 张天祎

## 【实验题目】文件打包实验

**【实验目的】**掌握文件打包的方法。

### 【实验说明】

- 把源程序和可执行文件放在相应的**上交源码**目录中。
- **截屏**用按键(Ctrl+Alt+PrintScreen)截取当前窗口
- **把每段具有独立功能的代码单独写入一个函数**有助于编码和调试。

### 【参考资料】

- C 语言字符串函数: [http://msdn.microsoft.com/zh-cn/library/f0151s4x\(v=vs.110\).aspx](http://msdn.microsoft.com/zh-cn/library/f0151s4x(v=vs.110).aspx)
- C 语言程序设计: <http://www.runoob.com/cprogramming/>
- C 语言函数分类: [http://msdn.microsoft.com/zh-cn/library/2aza74he\(v=vs.110\).aspx](http://msdn.microsoft.com/zh-cn/library/2aza74he(v=vs.110).aspx)

### 【实验环境】

- ♦ Windows + VS 2017
- ♦ Linux + gcc

### 【实验内容】

1、(StructSave.cpp)把输入的结构数据保存到文件中。

■ 实验要求:

循环输入员工(Person)的信息, 每输入一个员工的信息, 立即写入文件(Persons.stru), 直到输入的姓名为 **exit** 时跳出循环。

Person 的信息表示:

```
struct Person {
    char username[USER_NAME_LEN];    // 员工名
    int level;                        // 工资级别
    char email[EMAIL_LEN];           // email 地址
    DWORD sendtime;                   // 发送时间
    time_t regtime;                   // 注册时间
};
```

- 字符串函数和自定义函数(仅作参考):

```
printf(), scanf_s(), strcpy_s()
int inputOnePerson(Person *personSent) {...}
```

■ 参考运行结果:

```
C:\C-92-001\StructSave\Debug>StructSave
name: zhang
level: 8
email: zhang@sina.com

name: li
level: 10
email: li@163.com

name: wang
level: 9
email: wang@mail.sysu.edu.cn

name: exit
press any key to continue..
```

- 完成后截屏运行结果:



Microsoft Visual Studio 调试控制台

```
name:zty
level:9
email:1065426133@qq.com
name:wz11
level:1
email:zhangty56@mail2.sysu.edu.cn
name:zyc
level:9
email:1111111111@qq.com
name:exit
```

▪ 源代码:

```
#pragma warning( disable : 4996)
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<time.h>
#define USER_NAME_LEN 10
#define EMAIL_LEN 50
#define BUFF_LEN 10000
#define END "exit"
#define FILE_NAME "C:\\\\Users\\E480\\Desktop\\Persons.stru"

typedef struct {
    char username[USER_NAME_LEN];    // 员工名
    int level;                       // 工资级别
    char email[EMAIL_LEN];           // email 地址
    time_t sendtime; //DWORD         // 发送时间
    time_t regtime;                  // 注册时间
}Person;

Person input_one_person()
{
    char buff[BUFF_LEN] = {};
    time_t now;
    Person curr;
    printf("name:");
    scanf_s("%s", buff, BUFF_LEN);
    strcpy(curr.username, buff);
    if (!strcmp(curr.username, END)) return curr;
    printf("level:");
    scanf_s("%s", buff, BUFF_LEN);
    curr.level = atoi(buff);
    printf("email:");
    scanf_s("%s", buff, BUFF_LEN);
    strcpy(curr.email, buff);
    (void)time(&now);
    curr.regtime = now;
```



```
curr.sendtime = now;
return curr;
}

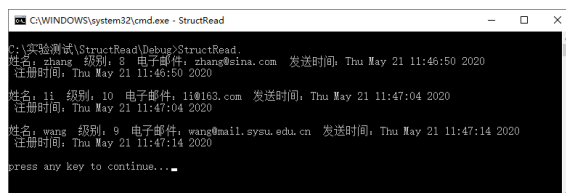
int main()
{
    //printf("%d", sizeof(Person));
    FILE* outFile = fopen(FILE_NAME, "wb");
    if (!outFile)
    {
        printf("ERROR!\n");
        return 0;
    }
    char buff[BUFF_LEN] = {};
    Person curr;
    while (1)
    {
        Person curr = input_one_person();
        if (!strcmp(curr.username, END)) break;
        memcpy(buff, (void*)&curr, sizeof(Person));
        if (fwrite(&curr, sizeof(Person), 1, outFile) != 1) {
            printf("file write error!\n");
        }
    }
    fclose(outFile);
    return 0;
}
```

## 2、(StructRead.cpp)从文件读出结构数据并显示出来。

### ▪ 实验要求:

读出上个步骤在文件(Persons.stru)中保存的结构数据并显示出来。

### ▪ 参考运行结果:



```
C:\WINDOWS\system32\cmd.exe - StructRead
2:\实验测试\StructRead\Debug>StructRead
姓名: zhang 级别: 8 电子邮件: zhang@sina.com 发送时间: Thu May 21 11:46:50 2020
-----
姓名: li 级别: 10 电子邮件: li@163.com 发送时间: Thu May 21 11:47:04 2020
-----
姓名: wang 级别: 9 电子邮件: wang@mail.sysu.edu.cn 发送时间: Thu May 21 11:47:14 2020
-----
press any key to continue...
```

### ▪ 完成后截屏运行结果:



```
Microsoft Visual Studio 调试控制台
姓名: zty 级别: 9 电子邮件: 106542613@qq.com 发送时间: Tue May 26 14:55:07 2020
-----
姓名: wz11 级别: 1 电子邮件: zhangty56@mail2.sysu.edu.cn 发送时间: Tue May 26 14:55:35 2020
-----
姓名: zyc 级别: 9 电子邮件: 1111111111@qq.com 发送时间: Tue May 26 14:56:05 2020
-----
2:\Users\E480\Desktop\计网\实验\实验四\W\2\Debug>2.exe (进程 19708) 已退出, 代码为 0。
请在调试停止时自动关闭控制台, 请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”,
按任意键关闭此窗口。...
```



▪ 源代码:

```
#pragma warning( disable : 4996)
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<time.h>
#define USER_NAME_LEN 10
#define EMAIL_LEN 50
#define BUFF_LEN 10000
#define TIME_BUF_LEN 30
#define END "exit"
#define FILE_NAME "C:\\\\Users\\E480\\Desktop\\Persons.stru"

typedef struct {
    char username[USER_NAME_LEN];    // 员工名
    int level;                        // 工资级别
    char email[EMAIL_LEN];           // email 地址
    time_t sendtime; //DWORD          // 发送时间
    time_t regtime;                  // 注册时间
}Person;

int main()
{
    //printf("%d", sizeof(Person));
    FILE* inFile = fopen(FILE_NAME, "rb");
    if (!inFile)
    {
        printf("ERROR!\n");
        return 0;
    }
    char buff[BUFF_LEN] = {};
    char sendtime[TIME_BUF_LEN] = {};
    char regtime[TIME_BUF_LEN] = {};

    Person curr;
    while (fread(buff, sizeof(Person), 1, inFile))
    {
        memcpy((void*)&curr, buff, sizeof(Person));
        ctime_s(sendtime, TIME_BUF_LEN, &curr.sendtime);
        ctime_s(regtime, TIME_BUF_LEN, &curr.regtime);
        printf("姓名:%s 级别:%d 电子邮件:%s 发送时间:%s\n 注册时间:%s\n",
            curr.username, curr.level, curr.email, sendtime, regtime)
    }
    fclose(inFile);
    return 0;
}
```



}

### 3、打包文件 (FilePack.cpp)

#### ■ 实验要求:

- (1) 先输入打包文件名 (含路径);
- (2) 循环输入要打包的文件名 (含路径), 每输入一个, 就把该文件的文件名 (最多 300 字节)、文件大小(long) 和文件内容写入文件 FileSet.pak 中, 当输入文件名为 **exit** 时跳出循环。

#### ■ 老师用到的字符串函数和自定义函数(仅作参考):

strcpy\_s(), scanf\_s(), printf(), strcmp(), sprintf\_s()—用于多个字符串和整数合并成一个字符串  
strrchr()—反向查找字符

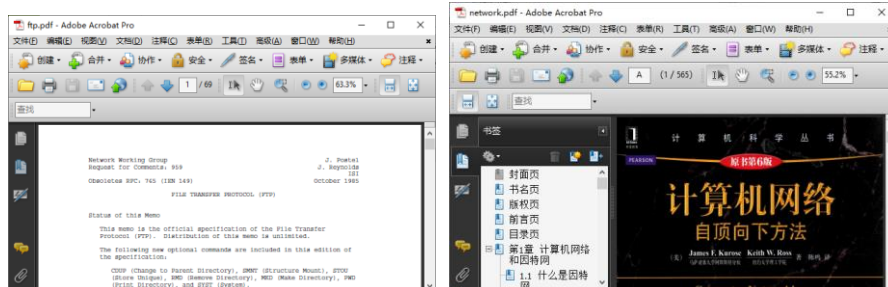
```
struct FileStruct {  
    char fileName[300];  
    __int64 fileSize;  
};  
  
__int64 getFileSize(char * fileName) {...}  
  
char * getFileName(char *pathName) {...}  
  
int packFile(char *srcFileName, FILE * destFile) {...}
```

#### ■ 参考运行结果:

c:\Test 下有两个文件 ftp.pdf 和 network.pdf, c:\Test\s 为空文件夹。

此电脑 > OS (C:) > Test				
名称	修改日期	类型	大小	
s	2020/5/21 14:26	文件夹		
ftp.pdf	2002/3/28 4:53	PDF 文件	84 KB	
network.pdf	2017/10/30 16:47	PDF 文件	121,686 KB	

打开 ftp.pdf 和 network.pdf



运行程序:

```
C:\WINDOWS\system32\cmd.exe - FilePack  
C:\实验测试>cmd  
Microsoft Windows [版本 10.0.18362.836]  
(c) 2019 Microsoft Corporation. 保留所有权利。  
  
C:\实验测试>FilePack  
输入目标文件名(含路径): c:\Test\FilePack.pak  
输入要打包的#1文件(含路径): c:\Test\ftp.pdf  
输入要打包的#2文件(含路径): c:\Test\network.pdf  
输入要打包的#3文件(含路径): c:\Test\ftp.pdf  
输入要打包的#4文件(含路径): c:\Test\network.pdf  
输入要打包的#5文件(含路径): c:\Test\ftp.pdf  
输入要打包的#6文件(含路径): exit  
打包结束!  
按任意键继续...
```

c:\Test 的内容变为:



名称	修改日期	类型	大小
s	2020/5/21 14:26	文件夹	
FilePack.pak	2020/5/21 14:45	PAK 文件	243,622 KB
ftp.pdf	2002/3/28 4:53	PDF 文件	84 KB
network.pdf	2017/10/30 16:47	PDF 文件	121,686 KB

## ▪ 完成后测试截图：



计网 > 实验 > 实验四 > MY > 3

名称	修改日期	类型	大小
kon.pak	2020/5/26 20:32	PAK 文件	84 KB
test.txt	2020/5/26 20:30	文本文件	1 KB
源.cpp	2020/5/26 20:16	C++ 文件	3 KB
3.vcxproj	2020/5/26 16:01	VC++ Project	7 KB
3.vcxproj.filters	2020/5/26 16:01	VC++ Project Fil...	1 KB
3.sln	2020/5/26 16:00	Visual Studio Sol...	2 KB
3.vcxproj.user	2020/5/26 16:00	USER 文件	1 KB
ftp.pdf	2002/3/28 4:53	Foxit PhantomP...	84 KB
Debug	2020/5/26 20:16	文件夹	
.vs	2020/5/26 16:00	文件夹	

## ▪ 源代码：

```
#pragma warning( disable : 4996)
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<Windows.h>
#define MAX_FILE_NAME 300
#define BUFF_LEN 100
#define MAX_FILE_NUM 10
#define END "exit"
typedef struct{
    char fileName[MAX_FILE_NAME];
    __int64 fileSize;
}FileStruct;

long getFileSize(char* fileName)
{
    long rst = 0;
    char buff[BUFF_LEN] = {};
    FILE* in_File = fopen(fileName, "rb");
    if (!in_File)
    {
        printf("IN FILE OPEN ERROR!");
        return 0;
    }
    while (fread(buff, 1, 1, in_File)) {
        rst ++;
    }
    fclose(in_File);
}
```



```
        return rst;
    }

void packFile(char* out_File_name, FileStruct* file, int num)
{
    FILE* out_File = fopen(out_File_name, "wb");
    if (!out_File)
    {
        printf("OUT FILE OPEN ERROR!");
        return;
    }
    for (int i = 0; i < num; i++)
    {
        FILE* in_File = fopen(file[i].fileName, "rb");
        if (!in_File)
        {
            printf("IN FILE OPEN ERROR!");
            return;
        }
        char buff[BUFF_LEN] = {};
        int len = 0;
        strcpy(buff, strrchr(file[i].fileName, '\\') + 1);
        strcpy(file[i].fileName, buff);
        fwrite((void*)&file[i], sizeof(FileStruct), 1, out_File);
        while ((len = fread(buff, 1, BUFF_LEN, in_File)) >= BUFF_LEN) {
            fwrite(buff, 1, BUFF_LEN, out_File);
        }
        fwrite(buff, 1, len, out_File);    // 写入剩余部分
        fclose(in_File);
    }
    fclose(out_File);
}

int main()
{
    char buff[BUFF_LEN] = {};
    char tem[BUFF_LEN] = {};
    int num = 0;
    FileStruct file[MAX_FILE_NUM] = {};
    printf("输入目标文件名（含路径）：");
    scanf_s("%s", buff, BUFF_LEN);
    strcpy(tem, buff);
    while (1)
    {
```



```
printf("输入要打包的%d 文件（含路径）：", num+1);  
scanf_s("%s", buff, BUFF_LEN);  
if (!strcmp(buff, END)) break;  
strcpy(file[num].fileName, buff);  
file[num].fileSize = getFileSize(buff);  
num++;  
}  
packFile(tem, file, num);  
return 0;  
}
```

#### 4、打包文件 (FileUnpack.cpp)

##### 实验要求：

- (1) 输入解包路径
- (2) 输入打包文件名，然后解包其中每一个文件, 有重名文件时文件名加上序号（从 2 开始）。

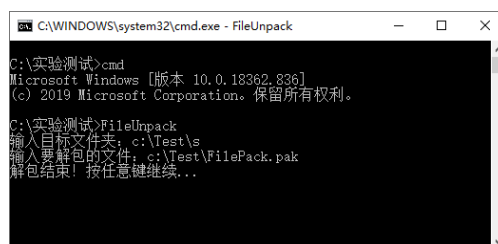
##### 老师用到的字符串函数和自定义函数(仅作参考)：

strcpy\_s(), scanf\_s(), printf(), sprintf\_s()

```
struct FileStruct {  
    char fileName[300];  
    __int64 fileSize;  
};  
// 拷贝filePathName中前面长度为len的字符串到fileFullName  
int mystrcpy(char * fileFullName, int len, char * filePathName) {...}  
void getUniqueName(char *newFileName, char *filePathName) {...}  
int unpackFile(FILE *srcFile, char *Path) {...}
```

##### 参考运行结果：

运行程序：



```
C:\WINDOWS\system32\cmd.exe - FileUnpack  
C:\实验测试>cmd  
Microsoft Windows [版本 10.0.18362.836]  
(c) 2019 Microsoft Corporation. 保留所有权利。  
C:\实验测试>FileUnpack  
输入目标文件夹: c:\Test\s  
输入要解包的文件: c:\Test\FilePack.pak  
解包结束! 按任意键继续...
```

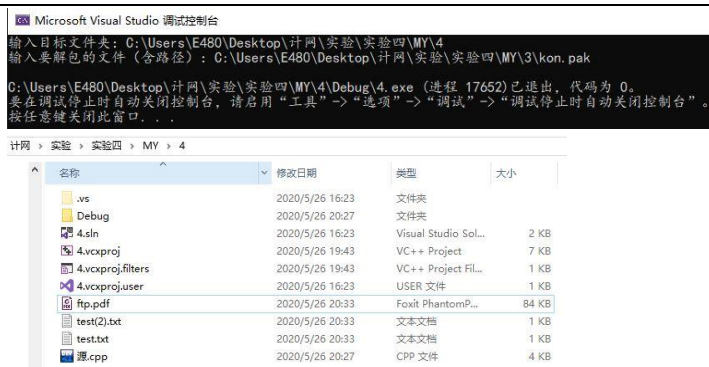
c:\Test\s 的内容变为：

名称	修改日期	类型	大小
ftp(2).pdf	2020/5/21 14:58	PDF 文件	84 KB
ftp(3).pdf	2020/5/21 14:58	PDF 文件	84 KB
ftp.pdf	2020/5/21 14:58	PDF 文件	84 KB
network(2).pdf	2020/5/21 14:58	PDF 文件	121,686 KB
network.pdf	2020/5/21 14:58	PDF 文件	121,686 KB

c:\Test\s 下的五个文件均可打开，显示内容正常，字节数与源文件相同。

##### 完成后截屏运行结果：





## 源代码:

```
#pragma warning( disable : 4996)
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<Windows.h>
#define MAX_FILE_NAME 300
#define BUFF_LEN 100
#define MAX_FILE_NUM 10
#define END "exit"
typedef struct{
    char fileName[MAX_FILE_NAME];
    __int64 fileSize;
}FileStruct;

void add_str(char* str, char* front_app)
{
    int len = strlen(front_app);
    char tem[BUFF_LEN] = {};
    strcpy(tem, front_app);
    tem[len] = '\\';
    tem[len + 1] = '\\0';
    strcpy(tem + len + 1, str);
    strcpy(str, tem);
}

int stoi(char* str)
{
    int pos = 0;
    int rst = 0;
    while (str[pos])
    {
        if (str[pos] < '0' || str[pos] > '9') return 0;
        rst = rst * 10 + str[pos] - '0';
        pos++;
    }
}
```



```
    return rst;
}

void UnpackFile(char* folder, char* file)
{
    FILE* in_File = fopen(file, "rb");
    char buff[BUFF_LEN] = {};
    char name[MAX_FILE_NUM][BUFF_LEN] = {};
    int num = 0;
    FileStruct curr;
    while(fread(&curr, sizeof(FileStruct), 1, in_File))
    {
        int i = 0;
        for (; i < num; i++)
        {
            if (!strcmp(curr.fileName, name[i]))
            {
                char* app = NULL;
                char* tem = NULL;
                char* ttem = NULL;
                char dot[BUFF_LEN] = {'\0'};
                app = strrchr(curr.fileName, '.');
                if (app)
                {
                    strcpy(dot, app);
                }
                if ((tem = strrchr(curr.fileName, '(')) && (ttem = strrchr(curr.fileName, ')')))
                {
                    char times[BUFF_LEN] = {};
                    strncpy(times, tem+1, ttem - tem-1);
                    int n = stoi(times);
                    if (!n)
                    {
                        strcpy(app, "(2)");
                        strcpy(app + 3, dot);
                        break;
                    }
                    n++;
                    strcpy(tem + 1, itoa(n, times, 10));
                    int len = strlen(times);
                    tem[len + 1] = ')';
                    if (app)
                    {
                        strcpy(tem + len + 2, dot);
                    }
                }
            }
        }
    }
}
```



```
//tem[len + 2 +strlen(dot)] = '\0';
break;
}
strcpy(app, "(2)");
strcpy(app + 3, dot);
break;
}
}
if (i == num) strcpy(name[num++], curr.fileName);
add_str(curr.fileName, folder);
FILE* out_File = fopen(curr.fileName, "wb");
int len = curr.fileSize;
while (len)
{
    if (len > BUFF_LEN)
    {
        fread(buff, BUFF_LEN, 1, in_File);
        fwrite(buff, BUFF_LEN, 1, out_File);
        len -= BUFF_LEN;
    }
    else
    {
        fread(buff, len, 1, in_File);
        fwrite(buff, len, 1, out_File);
        len = 0;
    }
}
fclose(out_File);
}
fclose(in_File);
}

int main()
{
    char buff[BUFF_LEN] = {};
    char tem[BUFF_LEN] = {};
    printf("输入目标文件夹: ");
    scanf_s("%s", buff, BUFF_LEN);
    strcpy(tem, buff);
    printf("输入要解包的文件（含路径）: ");
    scanf_s("%s", buff, BUFF_LEN);
    UnpackFile(tem, buff);
    return 0;
}
```



## 5、与同学互测并截屏运行结果：

把打包的文件发给同学，看他是否可以取出其中文件，同样测试是否可以读出并取出同学发来的打包文件。

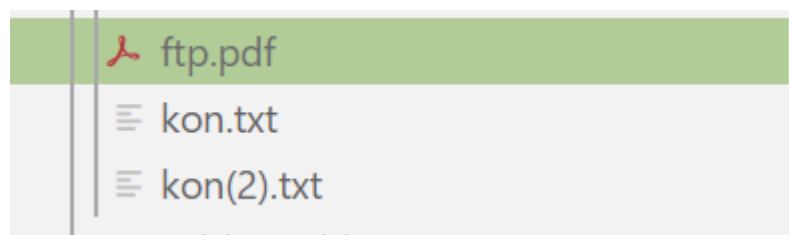
截屏同学发来的文件和解包结果：

解压同学 18340208 的文件：

```
C:\Users\E480\Desktop\计网\实验\实验四\MY\4\Debug\4.exe
输入目标文件夹：C:\Users\E480\Desktop\计网\实验\实验四\MY\4
输入要解包的文件（含路径）：C:\Users\E480\Desktop\计网\实验\实验四\MY\4\FileSet.pak
```

名称	修改日期	类型	大小
.vs	2020/5/26 16:23	文件夹	
Debug	2020/5/26 21:54	文件夹	
4.sln	2020/5/26 16:23	Visual Studio Sol...	2 KB
4.vcxproj	2020/5/26 19:43	VC++ Project	7 KB
4.vcxproj.filters	2020/5/26 19:43	VC++ Project Fil...	1 KB
4.vcxproj.user	2020/5/26 16:23	USER 文件	1 KB
chat.zip	2020/5/26 21:59	WinRAR ZIP 压缩...	45 KB
FileSet.pak	2020/5/26 21:16	PAK 文件	129 KB
ftp.pdf	2020/5/26 21:59	Foxit PhantomP...	84 KB
源.cpp	2020/5/26 21:53	CPP 文件	4 KB

同学 18340210 解压我的文件：



### 【完成情况】

是否完成以下步骤？（√完成 ×未做）

(1) [√] (2) [√] (3) [√] (4) [√] (5) [√]

互测同学的学号姓名：18340208 张洪宾 18340210 张景润

### 【实验体会】

上学期有幸上了乔海燕老师的算法与数据结构课程，其中一个大作业是完成霍夫曼编码压缩。这次作业不过是其中一部分工作而已。所以完成的很顺利。

### 【交实验报告】

每位同学单独完成本实验内容并填写实验报告。

交作业地点：<http://103.26.79.35/netdisk/default.aspx?vm=18net>  
文件上交\编程实验\文件打包

截止日期：2020 年 5 月 27 日晚上 9:00（周三）。

上传文件：学号\_姓名\_文件打包.doc

学号\_姓名\_文件打包.rar（源程序和可执行程序）