# 这一节我们来学习文件复制的各种方法

# 学习大纲

|  |
| --- |
|  |

# 演练：这一节做3个练习项目

# 第一个项目是c语言的

## 老师范例1.

|  |
| --- |
|  |

## 1.新建一个Lesson17-3-way-copy文件夹，在里面新建一个叫做copy\_c的c++常规空项目，然后添加一个叫做copy\_c.cpp的源文件，如图

|  |
| --- |
|  |

## 2.我们编写下面的代码

### copy\_c.cpp

|  |
| --- |
| #include<stdio.h>  #include<errno.h>  #include<stdlib.h>  #define BUF\_SIZE 256  int main(int argc,char\* argv[])  {  FILE \*fIn,\*fOut;  char szRead[BUF\_SIZE];  size\_t nIn,nOut;    if(argc < 3)  {  printf("Usage: copy\_c file1 file2\n");  return -1;  }  fIn = fopen(argv[1],"rb");  if(fIn ==NULL)  {  printf("open source file failed\n");  return 1;  }  fOut = fopen(argv[2],"wb");  if(fOut ==NULL)  {  printf("create file failed\n");  return 2;  }  while((nIn = fread(szRead,1,256,fIn)) && nIn > 0)  {  nOut = fwrite(szRead,1,nIn,fOut);  if(nOut != nIn)  {  printf("Fatal Error!!\n");  return 3;  }  }  fclose(fIn);  fclose(fOut);  printf("Copy success....\n");  system("pause");  return 0;  } |

### 效果

|  |
| --- |
|  |

# 第二个是Win32的ReadFile和WriteFile两个API的使用实例

## 老师范例2.

|  |
| --- |
|  |

## 1，新建一个c++常规空项目，取名：copy\_w1,新建一个cpp文件叫做copy\_w1.cpp

|  |
| --- |
|  |

## 2.然后添加下面的代码

### copy\_w1.cpp

|  |
| --- |
| #include<Windows.h>  #include<stdio.h>  #include<stdlib.h>  #define BUF\_SIZE 256  int main(int argc,char\* argv[])  {  HANDLE hIn,hOut;  char szRead[BUF\_SIZE];  DWORD dIn,dOut;  if(argc < 3)  {  printf("Usage: copy\_w1 file1 file2\n");  return -1;  }  hIn = CreateFile(argv[1],GENERIC\_READ,FILE\_SHARE\_READ,NULL,OPEN\_EXISTING,FILE\_ATTRIBUTE\_NORMAL,NULL);  if(INVALID\_HANDLE\_VALUE == hIn)  {  printf("Open file error:%x\n",GetLastError());  return 1;  }  hOut = CreateFile(argv[2],GENERIC\_WRITE,0,NULL,CREATE\_ALWAYS,FILE\_ATTRIBUTE\_NORMAL,NULL);  if(INVALID\_HANDLE\_VALUE == hOut)  {  printf("Create file error:%x\n",GetLastError());  return 2;  }  while(ReadFile(hIn,szRead,BUF\_SIZE,&dIn,NULL) && dIn >0)  {  WriteFile(hOut,szRead,dIn,&dOut,NULL);  if(dOut != dIn)  {  printf("Fatal Error:%x\n",GetLastError());  return 3;  }  }  printf("Copy File Successfully...\n");  CloseHandle(hIn);  CloseHandle(hOut);  system("pause");  return 0;  } |

### 效果：

|  |
| --- |
|  |

### 注意：写入的字节数一个和读取到的字节数一样，否则拷贝是不完整的，不正确的。使用我们需要添加对这两个值的判断。

# 第三个项目直接使用CopyFile这个API来实现

### copy\_w2.cpp

|  |
| --- |
| #include<Windows.h>  #include<stdio.h>  #include<stdlib.h>  int main(int argc,char\* argv[])  {  if(argc < 3)  {  printf("Usage: copy\_w2 file1 file2\n");  return -1;  }  if(!CopyFile(argv[1],argv[2],TRUE))  {  printf("File %s already exists,over write(y/n)?: ",argv[1]);  if('y' == getchar())  {  if(!CopyFile(argv[1],argv[2],FALSE))  {  printf("Copy Error: %x\n",GetLastError());  return 1;  }  else  {  printf("Copy Finished Successfully...");  }  }  }  else  {  printf("Copy Finished Successfully...");  }  return 0;  } |

### 效果

|  |
| --- |
|  |

# 小结，

## 这里学习了3种文件复制方法，其中CopyFile函数是最简便的，它可以自动打开文件和创建文件，可以一次完成复制，还会自动关闭句柄。我们以后就可以使用它。当然我们还是需要多写一些前两个例子的代码，这对于理解文件复制原理很有好处

# 扩展,这里有一个宽字符版本的程序

|  |
| --- |
|  |

## 3wcopy.cpp的源代码如下

|  |
| --- |
| #include<cstdio>  #include<Windows.h>  #include<fstream>  using namespace std;  #define bufsiz 256  int ccopy(PWCHAR src, PWCHAR dst)  {  FILE\* fin, \* fout;  size\_t sRead, sWrite;  WCHAR buf[bufsiz];  \_wfopen\_s(&fin, src, L"rb,ccs=utf-8");  \_wfopen\_s(&fout, dst, L"wb,ccs=utf-8");//这里使用字节拷贝,跟编码没有关系  while ((sRead=fread(buf,1, bufsiz, fin))>0)  {  sWrite = fwrite(buf,1, sRead, fout);  if (sWrite != sRead)  {  return -1;  }  }  fclose(fin);  fclose(fout);  return 0;  }  int cppcopy(PWCHAR src, PWCHAR dst)  {  WCHAR buf[bufsiz];  memset(buf, 0, bufsiz);  wifstream wifs;  wifs.open(src, ios::in|ios::binary);//使用二进制拷贝比较好,什么文件都能够拷贝  wofstream wofs;  wofs.open(dst, ios::out|ios::binary);  while (!wifs.eof())  {  wifs.read(buf, bufsiz);  wofs.write(buf, bufsiz);  }  wofs.flush();  wifs.close();  wofs.close();  return 0;  }  BOOL win32copy(PWCHAR src, PWCHAR dst)  {  HANDLE hIn, hOut;  DWORD dwRead, dwWrite;  WCHAR buf[bufsiz];  memset(buf, 0, bufsiz);  hIn = CreateFile(src, GENERIC\_READ, 0, NULL, OPEN\_EXISTING, FILE\_ATTRIBUTE\_NORMAL, NULL);  if (hIn == INVALID\_HANDLE\_VALUE)  {  wprintf\_s(L"Open Source File Failed,code:%d\n", GetLastError());  return FALSE;  }  hOut = CreateFile(dst, GENERIC\_WRITE, 0, NULL, CREATE\_ALWAYS, FILE\_ATTRIBUTE\_NORMAL, NULL);  if (hOut == INVALID\_HANDLE\_VALUE)  {  wprintf\_s(L"Create Target File Failed,code:%d\n", GetLastError());  CloseHandle(hIn);  CloseHandle(hOut);  return FALSE;  }  while ((ReadFile(hIn, buf, bufsiz, &dwRead, NULL)) && dwRead > 0)  {  WriteFile(hOut, buf, dwRead, &dwWrite, NULL);  if (dwWrite != dwRead)  {  wprintf\_s(L"Copy File Failed,code:%d\n", GetLastError());  CloseHandle(hIn);  CloseHandle(hOut);  return FALSE;  }  }  wprintf\_s(L"Copy File SuccessFully...\n");  CloseHandle(hIn);  CloseHandle(hOut);  return TRUE;//拷贝成功返回TRUE  }  int wmain(int argc, PWCHAR argv[])  {  if (argc < 3)  {  wprintf\_s(L"Usage: %s file1 file2\n", argv[0]);  return -1;  }  //方式1,c语言OK  /\*if (ccopy(argv[1], argv[2])!=0)  {  wprintf\_s(L"File copy Failed \n");  return -1;  }\*/  //方式2,c++  //cppcopy(argv[1], argv[2]);//ok  //方式3 win32手动拷贝  /\*if (!win32copy(argv[1], argv[2]))  {  wprintf\_s(L"File copy Failed \n");  return -1;  }\*/  //方式4 直接调用CopyFile函数  if (!CopyFile(argv[1], argv[2],FALSE))  {  wprintf\_s(L"File copy Failed \n");  return -1;  }  else  {  wprintf\_s(L"Copy File SuccessFully...\n");  }  return 0;  } |

## 注意:复制文件的时候,最好使用字节复制的方法,因为它既可以复制文本文件也可以复制二进制文件如图片视频等等

# 这一节的学习到此为止