**山东师范大学**

**实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 院系部所 | 信息科学与工程学院 | | 课程名称 | | 计算机操作系统 |
| 所在班级 | 计工本1702 | 学号 | 201711010202 | 姓名 | 王汝芸 |
| 实验编号 | 08 | 题目 | 采用高速缓存实现文件读写 | | |

一、实验目的和要求

了解Windows系统文件高速缓存的概念

熟悉Windows系统文件读/写相关API

掌握采用缓冲方式实现文件读/写相关参数的设置

二、实验环境

Windows10

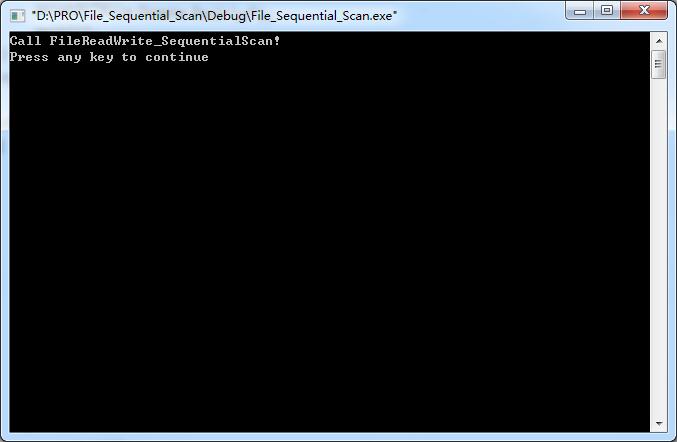
Visual C++6.0

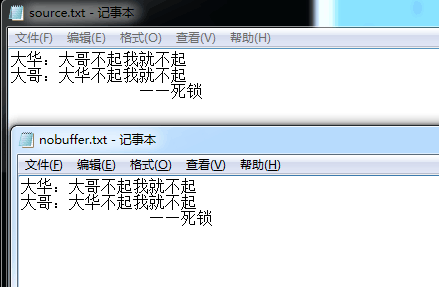
三、实验内容及实施

文件创建->读文件->写文件->关闭文件句柄

建立一个函数，使用该函数将源文件source.txt中的内容读出，再写到目标文件sequential.txt中去。

四、实验结果





五、实验分析

该实验完成缓冲方式的文件读/写操作。先创建两个文件：source.txt和sequential.txt，然后反复从文件source.txt中读出数据块，并写到文件sequential.txt中去，直到文件尾为止。

六、实验讨论（可选）

无。

七、附录 （可选）

1 *// File\_Sequential\_Scan.cpp : Defines the entry point for the console application.*

2 *//*

3

4 #include "stdafx.h"

5 #include "File\_Sequential\_Scan.h"

6

7 #ifdef \_DEBUG

8 #define new DEBUG\_NEW

9 #undef THIS\_FILE

10 **static char** THIS\_FILE[] = \_\_FILE\_\_;

11 #endif

12

13 DWORD BufferSize=1024;

14 **char** buf[1024];

15 **void** **FileReadWrite\_Sequential\_Scan**(**char**\*source,**char**\*destination);

16 */////////////////////////////////////////////////////////////////////////////*

17 *// The one and only application object*

18

19 CWinApp theApp;

20

21 **using namespace** std;

22

23 **int** **\_tmain**(**int** argc, TCHAR\* argv[], TCHAR\* envp[])

24 {

25 **int** nRetCode = 0;

26

27 **printf**("Call FileReadWrite\_SequentialScan!\n");

28 **FileReadWrite\_Sequential\_Scan**("source.txt","nobuffer.txt");

29

30 **return** nRetCode;

31 }

32

33 **void** **FileReadWrite\_Sequential\_Scan**(**char**\*source,**char**\*destination)

34 {

35 HANDLE handle\_src,handle\_dst;

36 DWORD NumberOfByteRead,NumberOfByteWrite;

37 BOOL cycle;

38 **char**\*buffer;

39 buffer=buf;

40

41 handle\_src=**CreateFile**(source,

42 GENERIC\_READ,

43 0,

44 NULL,

45 OPEN\_EXISTING,

46 FILE\_FLAG\_NO\_BUFFERING,

47 NULL);

48 handle\_dst=**CreateFile**(destination,

49 GENERIC\_WRITE,

50 NULL,

51 NULL,

52 CREATE\_ALWAYS,

53 NULL,

54 NULL);

55 **if**( handle\_src==INVALID\_HANDLE\_VALUE||handle\_dst==INVALID\_HANDLE\_VALUE)

56 {

57 **printf**("File Create Fail!\n");

58 **exit**(1);

59 }

60 cycle=TRUE;

61

62

63 **while**(cycle)

64 {

65 NumberOfByteRead=BufferSize;

66 *//è???ä??*

67 **if**(!**ReadFile**(handle\_src,buffer,NumberOfByteRead,&NumberOfByteRead,NULL))

68 {

69 **printf**("Read File Error!%d\n",**GetLastError**());

70 **exit**(1);

71 }

72

73 **if**(NumberOfByteRead<BufferSize)cycle=FALSE;

74 *//??ä??*

75 **if**(!**WriteFile**(handle\_dst,buffer,NumberOfByteRead,&NumberOfByteRead,NULL))

76 {

77 **printf**("Write File Error!%d\n",**GetLastError**());

78 **exit**(1);

79 }

80 }

81 *//?é???ä????*

82 **CloseHandle**(handle\_src);

83 **CloseHandle**(handle\_dst);

84 }