|  |  |
| --- | --- |
| МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  Федеральное государственное автономное образовательное учреждение  высшего профессионального образования  **«Дальневосточный федеральный университет»** | |
| **ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ** | |
| **Департамент программной инженерии и искусственного интеллекта** | |
| Вычислительные системы сети и низкоуровневое программирование | |
| Отчет по лабораторной работе №5 «Работа с файлами на ассемблере» | |
|  | |
|  | Выполнил студент гр. Б9121-09.03.04 Шевелёв Р.В.  Проверил преподаватель  Терентьева А.М.  \_\_\_\_\_\_\_\_\_ |
| г. Владивосток  2022 г. | |

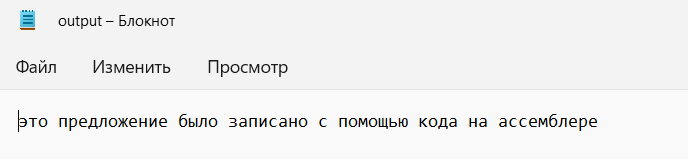
Цель лабораторной работы:

1 часть: В Visual Studio создать два проекта. Один проект пишет в файл, другой читает файл и добавляет в него данные.

2 часть: Чтение диска. Вывести полученные данные на экран в шестнадцатеричном виде.

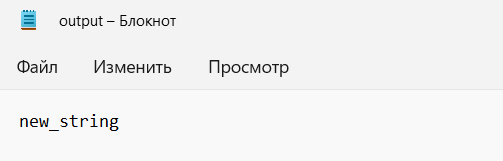
Скриншот экрана или содержимое файла вывода:

1 часть, запись в файл



1 часть, дозапись и чтение файла.

Начальный файл:



Чтение:

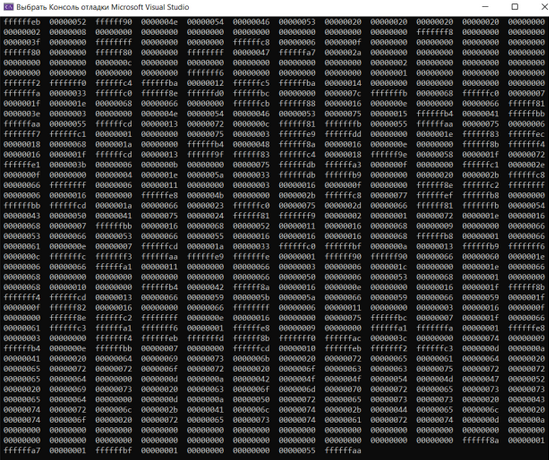


Дозапись:

Изображение выглядит как текст

Автоматически созданное описание

2 часть, чтение жёсткого диска.



Текст программы:

1 часть, запись в файл.

.386

.model flat,stdcall

option casemap: none

includelib kernel32.lib

DO\_NOT\_SHARE EQU 0

NULL EQU 0

FILE\_ATTRIBUTE\_NORMAL EQU 80h

GENERIC\_WRITE EQU 40000000h

CreateFile EQU <CreateFileA>

WriteConsole EQU <WriteConsoleA>

HANDLE TEXTEQU <DWORD>

CREATE\_ALWAYS EQU 2

CreateFile PROTO,

lpFilename: PTR BYTE,

dwDesiredAccess: DWORD,

dwShareMode: DWORD,

lpSecurityAttributes: DWORD,

dwCreationDisposition: DWORD,

dwFlagsAndAttributes: DWORD,

hTemplateFile: DWORD

WriteFile PROTO,

hFile: HANDLE,

lpBuffer: PTR BYTE,

nNumberOfBytesToWrite: DWORD,

lpNumberOfBytesWritten: PTR DWORD,

lpOverlapped: PTR DWORD

CloseHandle PROTO,

hObject:DWORD

.data

buffer BYTE "это предложение было записано с помощью кода на ассемблере", 0dh, 0ah

bufSize = 60

filename BYTE "output.txt",0

fileHandle DWORD ?

bytesWritten DWORD ?

.code

WriteProc PROC

INVOKE CreateFile, ADDR filename, GENERIC\_WRITE, DO\_NOT\_SHARE, NULL, CREATE\_ALWAYS, FILE\_ATTRIBUTE\_NORMAL, 0

mov fileHandle,eax

INVOKE WriteFile, fileHandle, ADDR buffer, bufSize, ADDR bytesWritten, 0

INVOKE CloseHandle, fileHandle

ret

WriteProc ENDP

END WriteProc

1 часть, чтение и дозапись в файл

.386

.model flat,stdcall

includelib kernel32.lib

GENERIC\_READ EQU 80000000h

DO\_NOT\_SHARE EQU 0

NULL EQU 0

OPEN\_EXISTING EQU 3

FILE\_ATTRIBUTE\_NORMAL EQU 80h

STD\_OUTPUT\_HANDLE EQU -11

GENERIC\_WRITE EQU 40000000h

FILE\_END EQU 2

CreateFile EQU <CreateFileA>

WriteConsole EQU <WriteConsoleA>

HANDLE TEXTEQU <DWORD>

CreateFile PROTO,

lpFilename: PTR BYTE,

dwDesiredAccess: DWORD,

dwShareMode: DWORD,

lpSecurityAttributes: DWORD,

dwCreationDisposition: DWORD,

dwFlagsAndAttributes: DWORD,

hTemplateFile: DWORD

ReadFile PROTO,

hFile: HANDLE,

lpBuffer: PTR BYTE,

nNumberOfBytesToRead: DWORD,

lpNumberOfBytesRead: PTR DWORD,

lpOverlapped: PTR DWORD

WriteConsole PROTO,

hConsoleOutput: HANDLE,

lpBuffer: PTR BYTE,

nNumberOfCharsToWrite: DWORD,

lpNumberOfCharsWritten: PTR DWORD,

lpReserved: DWORD

GetStdHandle PROTO,

nStdHandle:HANDLE

WriteFile PROTO,

hFile: HANDLE,

lpBuffer: PTR BYTE,

nNumberOfBytesToWrite: DWORD,

lpNumberOfBytesWritten: PTR DWORD,

lpOverlapped: PTR DWORD

SetFilePointer PROTO,

hFile: HANDLE,

lDistanceToMove: SDWORD,

lpDistanceToMoveHigh: PTR SDWORD,

dwMoveMethod: DWORD

CloseHandle PROTO,

hObject:DWORD

.data

buffer BYTE 100 DUP(?)

bAffer BYTE 100 DUP("new\_string ")

bufSize = 11

filename BYTE "output.txt",0

fileHandle DWORD ?

byteCount DWORD ?

.code

ReadProc PROC

INVOKE CreateFile, ADDR filename, GENERIC\_READ, DO\_NOT\_SHARE, NULL, OPEN\_EXISTING, FILE\_ATTRIBUTE\_NORMAL, 0

mov fileHandle,eax

INVOKE ReadFile, fileHandle, ADDR buffer, 100, ADDR byteCount, 0

INVOKE CloseHandle, fileHandle

INVOKE GetStdHandle, STD\_OUTPUT\_HANDLE

mov fileHandle, eax

INVOKE WriteConsole, fileHandle, ADDR buffer, 100, ADDR byteCount, 0

INVOKE CreateFile, ADDR filename, GENERIC\_WRITE, DO\_NOT\_SHARE, NULL, OPEN\_EXISTING, FILE\_ATTRIBUTE\_NORMAL, 0

mov fileHandle,eax

INVOKE SetFilePointer, fileHandle, 0, 0, FILE\_END

INVOKE WriteFile, fileHandle, ADDR bAffer, bufSize, ADDR byteCount, 0

INVOKE CloseHandle, fileHandle

ret

ReadProc ENDP

END ReadProc

2 часть, чтение диска.

Main.cpp

#include <iostream>

extern "C" char\* ReadDrive(char\* driveName);

int main()

{

char driveName[] = "\\\\.\\C:";

char\* sector = ReadDrive(driveName);

for (int i = 0; i < 512; i++)

{

printf(" %08x ", \*sector);

sector++;

}

return 0;

}

Read.asm

.686

.model flat, stdcall

option casemap : none

includelib kernel32.lib

GENERIC\_READ EQU 80000000h

DO\_NOT\_SHARE EQU 0

NULL EQU 0

OPEN\_EXISTING EQU 3

FILE\_ATTRIBUTE\_NORMAL EQU 80h

STD\_OUTPUT\_HANDLE EQU -11

GENERIC\_WRITE EQU 40000000h

FILE\_END EQU 2

CreateFile EQU <CreateFileA>

WriteConsole EQU <WriteConsoleA>

HANDLE TEXTEQU <DWORD>

FILE\_SHARE\_READ = 00000001h

FILE\_SHARE\_WRITE = 00000002h

CreateFile PROTO,

lpFilename: PTR BYTE,

dwDesiredAccess: DWORD,

dwShareMode: DWORD,

lpSecurityAttributes: DWORD,

dwCreationDisposition: DWORD,

dwFlagsAndAttributes: DWORD,

hTemplateFile: DWORD

WriteConsole PROTO,

hConsoleOutput: HANDLE,

lpBuffer: PTR BYTE,

nNumberOfCharsToWrite: DWORD,

lpNumberOfCharsWritten: PTR DWORD,

lpReserved: DWORD

ReadFile PROTO,

hFile: HANDLE,

lpBuffer: PTR BYTE,

nNumberOfBytesToRead: DWORD,

lpNumberOfBytesRead: PTR DWORD,

lpOverlapped: PTR DWORD

CloseHandle PROTO,

hObject:DWORD

.data

handle dword 0

buffer dword 512 dup(?)

bytesRead dword 0

.code

ReadDrive PROC C driveName: DWORD

invoke CreateFile, driveName, GENERIC\_READ, FILE\_SHARE\_READ+FILE\_SHARE\_WRITE, NULL, OPEN\_EXISTING, NULL, NULL

mov handle,eax

invoke ReadFile, handle, ADDR buffer, 512, ADDR bytesRead, 0

invoke CloseHandle,handle

mov eax, offset buffer

ret

ReadDrive ENDP

END