

Лабораторная работа № 8. ФАЙЛОВЫЙ ВВОД-ВЫВОД В WIN32

Для работы с файлами существует большой набор API-функций. Вот несколько таких функций:

```
HANDLE CreateFile(  
    LPCTSTR lpFileName,           // имя файла  
    DWORD dwDesiredAccess,       // режим  
доступа  
    DWORD dwShareMode,           // совместный  
доступ  
    LPSECURITY_ATTRIBUTES lpSecurityAttributes,  
//структура правил наследования процессов  
    DWORD dwCreationDisposition,  
// действия в случае, если объект уже есть  
    DWORD dwFlagsAndAttributes, // атрибуты  
файла  
    HANDLE hTemplateFile //указатель на  
настроечный //файл  
);
```

Эта функция – одна из самых мощных и сложных. С ее помощью можно открывать и создавать не только файлы, но и каналы, консоли, коммуникационные ресурсы и многое другое. Настоятельно рекомендуется ознакомиться с ее полной спецификацией.

Пара функций для поиска файлов – соответственно первого и следующего:

```
HANDLE FindFirstFile(  
    LPCTSTR lpFileName,  
// имя или маска имени файла  
    LPWIN32_FIND_DATA lpFindFileData  
//указатель на структуру с параметрами  
//найденного файла  
);
```

```
HANDLE FindNextFile(  
    HANDLE hFindFile,  
    // ссылка на предыдущий найденный файл  
    LPWIN32_FIND_DATA lpFindFileData  
    //указатель на структуру с параметрами  
    //найденного файла  
    );
```

Чтение и установка атрибутов файла:

```
DWORD GetFileAttributes(  
    LPCTSTR lpFileName  
    // указатель на строку с именем файла  
    );  
BOOL SetFileAttributes(  
    LPCTSTR lpFileName,  
    // указатель на строку с именем файла  
    DWORD dwFileAttributes  
    // атрибуты  
    );
```

Чтение из файла и запись в файл:

```
BOOL ReadFile(  
    HANDLE hFile, // ссылка на файл  
    LPVOID lpBuffer, // указатель на буфер данных  
    DWORD nNumberOfBytesToRead,  
    //число байт для чтения  
    LPDWORD lpNumberOfBytesRead,  
    // указатель на переменную с числом реально  
    // считанных байт  
    LPOVERLAPPED lpOverlapped  
    // указатель на структуру «перекрытий»  
    );
```

```
BOOL WriteFile(  
    HANDLE hFile,  
    // ссылка на файл  
    LPCVOID lpBuffer,
```

```

// указатель на буфер данных
    DWORD nNumberOfBytesToWrite,
// число байт для записи
    LPDWORD lpNumberOfBytesWritten,
// указатель на переменную с числом реально
// записанных байт
    LPOVERLAPPED lpOverlapped
// указатель на структуру «перекрытий»
    );

```

Приведенный список далеко не полный, но его достаточно для простейших файловых операций. Описание функций WinAPI для работы с файлами можно найти в [2, 3, 4].

Следующий пример иллюстрирует вывод содержимого файла, имя которого указано в параметре командной строки, на консоль. Стоит обратить внимание, что консоль тоже открывается как файл.

typefile.inc:

```

includelib      import32.lib

; имена используемых функций из kernel32.dll
                extrn    ExitProcess:near
                extrn    GetCommandLineA:near
                extrn    CreateFileA:near
                extrn    CloseHandle:near
                extrn    ReadFile:near
                extrn    WriteFile:near

; присваивания для облегчения читаемости кода
                GetCommandLine    equ
GetCommandLineA
                CreateFile        equ    CreateFileA

; определения констант и типов
STD_OUTPUT_HANDLE    equ    -11
GENERIC_READ         equ    80000000h

```

GENERIC_WRITE	equ	40000000h
OPEN_EXISTING	equ	3

typefile.asm:

```

; Консольное приложение, выводящее на консоль
; файл
include typefile.inc
.386
.model FLAT, STDCALL
.data
hcons          dd      ?
hfile          dd      ?
buf            db      100 dup(0)
bufer         db      300 dup(0)
numb          dd      ?
numw          dd      ?
nameout       db      'CONOUT$'
.code
_start:  call CreateFile, offset nameout, \
GENERIC_READ+GENERIC_WRITE, 0, 0, OPEN_EXISTING, \
0, 0
        mov hcons, eax; получение ссылки на консоль
; как на файл
        call GetCommandLine; в EAX указатель на
; командную строку
        mov esi, eax
        xor ecx, ecx          ;счетчик
        mov edx, 1           ;признак
n1: cmp byte ptr [esi], 0;конец строки
    je end_                  ;нет параметра
    cmp byte ptr [esi], 32;пробел
    je n3
    add ecx, edx
    cmp ecx, 2;Первый параметр - имя программы.
; Второй - имя файла.
    je n4
    xor edx, edx
    jmp n2

```

```

n3: or edx,1
n2: inc esi
    jmp n1
n4: call CreateFile, esi,\
    GENERIC_READ+GENERIC_WRITE,\
    0,0,OPEN_EXISTING,0,0
    mov hfile,eax;открытие файла, имя которого
; указано в командной строке
10: call ReadFile, hfile,offset bufer,300,\
    offset numb,0 ;чтение в буфер
    call WriteFile,hcons,offset bufer,numb,\
    offset numw,0;вывод на консоль как в файл
    cmp numb,300; numb<300 - файл закончился
    je 10
end_ : call ExitProcess,0
end_start

```

>tasm32 /ml typefile.asm -> console.obj (657b)

>tlink32 /Tpe /ap /c /x typefile.obj -> console.exe (4096b)

Результат:

```

D:\PC>console typefile.inc
includelib      import32.lib

; имена используемых функций из kernel32.dll
                extrn  ExitProcess:near
                extrn  GetCommandLineA:near
                extrn  CreateFileA:near
                extrn  CloseHandle:near
                extrn  ReadFile:near
                extrn  WriteFile:near

; присваивания для облегчения читаемости кода
GetCommandLine equ      GetCommandLineA
CreateFile     equ      CreateFileA

; определения констант и типов
STD_OUTPUT_HANDLE equ      -11
GENERIC_READ   equ      80000000h
GENERIC_WRITE  equ      40000000h
OPEN_EXISTING  equ      3

```

Задание к лабораторной работе

Составить программу – консольное приложение Windows, для варианта задания, соответствующего порядковому номеру студента в группе.

1. Удалить файл, имя которого задано в командной строке.

2. Переименовать файл, старое и новое имя задаются в командной строке.

3. Копировать файл из одной папки в другую. Имена файла и папок задаются в командной строке.

4. Вывести на экран размер файла, имя которого задано в командной строке.

5. Вывести на экран атрибуты файла, имя которого задано в командной строке.

6. Установить атрибут «только для чтения» файлу, имя которого задано в командной строке.

7. Установить атрибут «скрытый» файлу, имя которого задано в командной строке.

8. Вывести на экран содержимое папки, имя которой задано в командной строке. Выводить только имена (и типы) файлов и папок.

9. Создать заданную папку в текущей. Имя папки задается в командной строке.

10. Переместить файл из одной папки в другую. Имена файла и папок задаются в командной строке.

Порядок выполнения работы

1. Изучить основные сведения.

2. Изучить необходимые для выполнения задания функции Win API.

3. Разработать алгоритм и программу решения задачи на языке ассемблера, подготовить тестовые примеры.

4. Выполнить ввод, трансляцию, построение кода программы и получить результаты ее работы.

Содержание отчета

1. Цель работы.
2. Текст задания.
3. Описание используемых API-функций.
4. Текст программы.
5. Результаты работы программы.
6. Выводы.

Список рекомендуемой литературы

1. Андреева А.А. и др. Программирование на языке ассемблера в операционной системе Windows: лаб. практикум. Чебоксары: Изд-во Чуваш. ун-та, 2006. 104 с.
2. Зубков С.В. Ассемблер для DOS, Windows и UNIX. М.: ДМК Пресс, 2015. 638 с.
3. Пирогов В.Ю. Ассемблер для Windows. СПб.: БХВ-Петербург, 2007. 896 с.
4. Юров В. И. Assembler: практикум. СПб.: Питер, 2007. 400 с.
5. Аблязов Р. Программирование на ассемблере на платформе x86-64. М.: ДМК Пресс, 2016. 302 с.