

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

Национальный исследовательский университет «МЭИ»

Отчет по лабораторной работе №3

Дисциплина: Операционные системы

Выполнила:

Студентка группы А-05-18

Поиленкова А.А

Проверил: Чернецов А.М.

Вариант 18

Москва, 2019

Цель работы

Освоение работы с функциями win32api в программах на языке Ассемблер.

Задание

Разработать dll библиотеку на языке Ассемблер с функциями обработки строк. На языке C++ написать программу, статически подключающую разработанную библиотеку, и программу, динамически подключающую разработанную библиотеку.

Индивидуальное задание

Сформировать строку из слов исходной строки, содержащих заданную букву.

————— СОЗДАНИЕ —————

Первым делом создаем файл **lab.ass**

Из библиотеки мы будем вызывать функцию **process** с тремя параметрами

Код библиотеки

```
.386
.model flat, stdcall
option casemap:none
.code

DllMain proc hInstDLL:DWORD, reason:DWORD, reserved1:DWORD
    mov eax, -1
    ret
DllMain endp

process proc src:DWORD, res:DWORD, len:DWORD
    push ebp
    push esi
    push edi
    push ecx
    mov esi, [src] ; Входная строка
    mov edi, [res] ; Выходная строка
    mov ecx, [len] ; Длина
    xor eax, eax

L1: mov al, [esi]
    test eax, eax
    jz EX
    mov edx, 0
    mov ebx, esi

L2: mov al, [ebx]
    test eax, eax
    jz L3 ; флаг 0
    cmp byte ptr [ebx], ' ' ; Количество символов до следующего пробела пишем в edx (персылаем байт)
    je L3
    inc edx
    inc ebx
    jmp L2

L3:
```

```

L3: cmp edx, ecx ; Если длина слова подходит, то копируем в итог
    je L4
    mov al, [ebx]
    test eax, eax
    jz EX
    inc ebx
    mov esi, ebx
    jmp L1

```

```

L4: test edx, edx ; копирование слова в итог
    jz L5
    dec edx
    mov al, [esi]
    mov [edi], al
    inc esi
    inc edi
    jmp L4

```

```

L5: mov al, [esi] ; конец копирования, вставляем пробел
    test eax, eax
    jz EX
    mov [edi], al
    inc edi
    inc esi
    jmp L1

```

```

EX: xor eax, eax
    mov [edi], al
    pop ecx
    pop edi ; Выход
    pop esi
    pop ebp
    ret

```

```
process endp
```

End DllMain ;является обязательной для создания библиотеки.

----- СБОРКА -----

Для того, чтобы проект мог быть собран, как библиотека dll, требуется указать инициализирующую функцию библиотеки.

Далее определяются экспортируемые функции и их аргументы. Стоит обратить внимание на то, что указание параметров является важной частью создания библиотеки, так как на этапе состыковки библиотеки с проектом на C++ могут возникнуть проблемы компиляции проекта из-за неправильно указанных аргументов (например, если в библиотеке и в основном проекте указано разное количество аргументов для одной экспортируемой функции).

- Создаем файл **lab.des**

```

LIBRARY lab           //название
EXPORTS process       //экспортируемая процедура

```

- Устанавливаем на компьютер MASM
- В консоле пишем

```
\masm32\bin\ml /c /coff lab.asm
```

Производится компиляция проекта и создается файл lab.obj

```
\masm32\bin\Link /SUBSYSTEM:WINDOWS /DLL /DEF:lab.def lab.obj
```

В текущей папке появляются три файла с расширениями .exp .lib и .dll. В данной лабораторной работе будут использованы .lib и .dll для статического подключения и .dll для динамического подключения.

Удобно создать бат-файл, который позволит автоматизировать выполнение данной процедуры

```
@echo off
```

```
if exist main.obj del main.obj
```

```
if exist labdllhelpme.dll del labdllhelpme.dll
```

```
\masm32\bin\ml /c /coff lab.asm
```

```
\masm32\bin\Link /SUBSYSTEM:WINDOWS /DLL /DEF:lab.def lab.obj
```

```
dir main.*
```

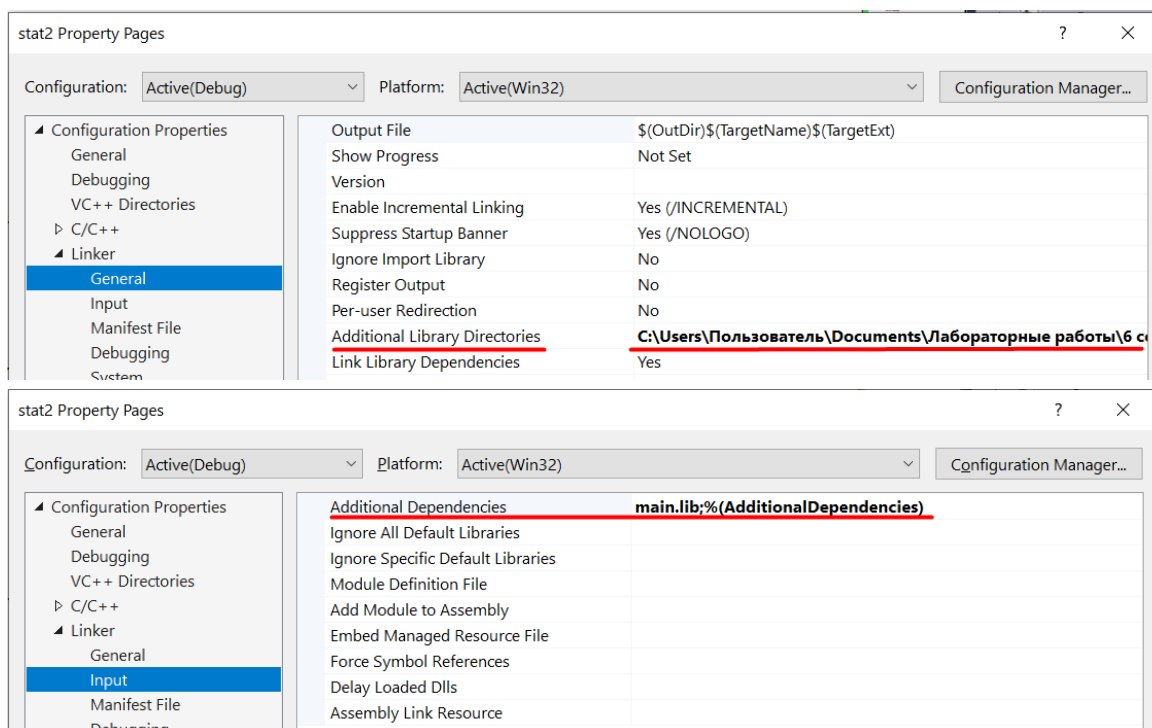
```
pause
```

Библиотека создана.

----- ПОДКЛЮЧЕНИЕ -----

Статическая библиотека

Указываем путь на библиотеку в проекте



- файл Header.h

```
#pragma once
extern "C" void __stdcall process(char*, char*, int);
```

- файл Lab3_ass_static.cpp

```
#include "pch.h"
#include <iostream>
#include "Header.h"
using namespace std;
char str1[256];
int main()
{
    int len = 0;
    cout << "Input str:";
    cin.getline(str1, 256);
    cout << "Len: ";
    cin >> len;
    process(str1, str1, len);
    cout << "Result: " << str1 << endl;
    system("pause");
    return 0;
}
```

Динамическая библиотека

lab3_ass_dinam.cpp

```
#include <cstdio>
#include <windows.h>
#include <iostream>
using namespace std;
#define _CRT_SECURE_NO_WARNINGS
```

```
typedef void(__stdcall* LibraryFunction)(char*, char*, int); // Объявляем тип для указателя на библиотечную функцию
```

```
int main() {
    HMODULE hLib;
    LibraryFunction f;
    TCHAR res[100];
```

```
    hLib = LoadLibraryEx(TEXT("C:\\Users\\Ann\\source\\repos\\lab3_ass_dinam\\lab.dll"), NULL, DONT_RESOLVE_DLL_REFERENCES);
```

```

if (hLib == NULL) // Проверяем результат загрузки библиотеки {
    wsprintf(res, L"Unable to load library");
    cout << res << endl;
}
else {
    f = (LibraryFunction)GetProcAddress(hLib, "process");
    if (!f) // Проверяем полученный указатель {
        wsprintf(res, L"Unable to load function");
        cout << res << endl;
    }
    else {
        char str1[256];
        int len = 0;
        cout << "Input string:";
        cin.getline(str1, 256);
        cout << "Len: ";
        cin >> len;
        f(str1, str1, len);

        cout << "Result: " << str1 << endl;
    }
    FreeLibrary(hLib); // Освобождаем библиотеку
}

system("pause");
return 0;

}

```