МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА (РУТ(МИИТ))»

Кафедра «Цифровые технологии управления транспортными процессами»

Отчёт

по лабораторной работе №1

по дисциплине «Ассемблер»

Тема: «Средства разработки приложений на языке ассемблер»

студента учебной группы УИС-311

Баклашкина Алексея Андреевича

Проверил: Варфоломеев В. А.

Москва

2021 г.

**Описание полного цикла разработки программы на языке ассемблер в среде MASM32.**

1. Создание и редактирование исходного кода программы с помощью текстового редактора. В результате получается исходный файл с расширением .asm.
2. Ассемблирование программы. В результате создаётся два файла: объектный файл (.obj) и файл с листингом программы (.lst).
3. Выявление синтаксических ошибок. Анализируются сообщения ассемблера и файла листинга. Если обнаружены ошибки, необходимо вернуться к шагу 1.
4. Линковка (редактирование связей). В результате будет получен исполняемый файл с расширением .exe.
5. Выполнение программы (запускается файл с расширением .exe)
6. Использование отладчика. На этом этапе можно выполнить программу пошагово, увидеть содержимое регистров процессора и ячеек памяти (дамп), значения переменных при выполнении.

**Краткое описание используемых ключей программ ассемблирования и редактирования связей**

/c - Транслирует без линковки. Этот ключ используется для создания только объектных файлов.

/coff - Включает MS COFF-формат (MS Common Object File Format). Необходим для создания обычных Win32-программ.

/Sn - Исключает из листинга таблицу символов

/Fl - Создаёт файл листинга трансляции. Создаётся текстовый файл, который помогает понять, во что транслируется каждая строка исходников.

/SUBSYSTEM - Выполняет компоновку исполняемого файла для подсистемы Win32. Необходим для линковки оконных приложений.

**Текст программы hello**

;Лабораторная работа №1

;Выполнил ст. гр. УИС-311 Баклашкин А.А.

;Программа 1 (HELLO.ASM)

.486

.model flat, stdcall ;Тип используемой модели памяти

option casemap: none

.stack 100h ;Определяется сегмент стека размером 256 байт

;Подключение необходимых системных модулей и библиотек

include I:\masm32\include\windows.inc;

include I:\masm32\include\user32.inc;

include I:\masm32\include\kernel32.inc;

includelib I:\masm32\lib\user32.lib;

includelib I:\masm32\lib\kernel32.lib;

.data ;Начало сегмента данных

tit db 'Привет, мир',0 ;Переменная - заголовок окна

hello db'Здравствуйте, я Баклашкин Алексей!',0;Резервируется память

;для переменной Hello

.code ;Начало сегмента кода

main: ;Метка, обозначающая точку входа в программу

push 0 ;Дескриптор окна

push offset tit ;Заголовок окна

push offset hello ;Текст для вывода

push 0 ;Стиль окна (0 – окно с кнопкой ОК)

call MessageBox ;Вызов функции вывода на экран

push 0 ;Код завершения

call ExitProcess ;Завершаем работу программы

end main

**Сегмент данных и сегмент кода из листинга программы hello (hello.lst)**

.data ;Начало сегмента данных (14)

00000000 CF F0 E8 E2 E5 tit db 'Привет, мир',0 ;Переменная - заголовок окна (15)

F2 2C 20 EC

E8 F0 00

0000000C C7 E4 F0 E0 E2 hello db'Здравствуйте, я Баклашкин Алексей!',0;Резервируется память (16)

F1 F2 E2 F3

E9 F2 E5 2C

20 FF 20 C1

E0 EA EB E0

F8 EA E8 ED

20 C0 EB E5

EA F1 E5 E9

21 00

;для переменной Hello

00000000 .code ;Начало сегмента кода (17)

00000000 main: ;Метка, обозначающая точку входа в программу (18

00000000 6A 00 push 0 ;Дескриптор окна (19)

00000002 68 00000000 R push offset tit ;Заголовок окна (20)

00000007 68 0000000C R push offset hello ;Текст для вывода (21)

0000000C 6A 00 push 0 ;Стиль окна (0 – окно с кнопкой ОК) (22)

0000000E E8 00000000 E call MessageBox ;Вызов функции вывода на экран (23)

00000013 6A 00 push 0 ;Код завершения (24)

00000015 E8 00000000 E call ExitProcess ;Завершаем работу программы (25)

end main

**Текст программы data**

;Лабораторная работа №1

;Выполнил ст. гр. УИС-311 Баклашкин А.А.

;Программа 2 (DATA.ASM) Вариант 1

;============================================================

.486

.model flat, stdcall ;Устанавливается тип используемой памяти

option casemap: none

;============================================================

.stack 100h ;Определяется сегмент стека размером 256 байт

;============================================================

;Подключение необходимых модулей

include I:\masm32\include\windows.inc

include I:\masm32\include\user32.inc

include I:\masm32\include\kernel32.inc

includelib I:\masm32\lib\user32.lib

includelib I:\masm32\lib\kernel32.lib

;============================================================

.data ;Начало сегмента данных

A1 db 27 ;Определение однобайтовой переменной

A2 dw 0AE5h ;Определение двубайтной переменной (слова)

A3 dd 123456h ;Определение двойного слова (4 байта)

A4 db -2 ;Определение однобайтового числа со знаком 18

A5 dd A4 ;Определение четырехбайтового указателя на переменную

;============================================================

.code ;Начало сегмента кода

main: ;Метка обозначающая точку входа в программу

mov bl, A1 ;Поместить в регистр BL значение переменной A1

mov bh, A4-2 ;Поместить в регистр BH байт памяти с адресом A4-2

mov ecx, offset A2 ;Поместить в регистр ECX адрес переменной A2

mov ax, A2 ;Поместить в регистр AX значение переменной A2

mov si, A2+1 ;Поместить в регистр SI слово памяти по адресу A2+1

call ExitProcess ;Вызов функции завершения программы

end main ;Конец программы

**Сегмент данных и сегмент кода из листинга программы data (data.lst)**

00000000 .data ;Начало сегмента данных

00000000 1B A1 db 27 ;Определение однобайтовой переменной

00000001 0AE5 A2 dw 0AE5h ;Определение двубайтной переменной (слова)

00000003 00123456 A3 dd 123456h ;Определение двойного слова (4 байта)

00000007 FE A4 db -2 ;Определение однобайтового числа со знаком 18

00000008 00000007 R A5 dd A4 ;Определение четырехбайтового указателя на переменную

;============================================================

00000000 .code ;Начало сегмента кода

00000000 main: ;Метка обозначающая точку входа в программу

00000000 8A 1D 00000000 R mov bl, A1 ;Поместить в регистр BL значение переменной A1

00000006 8A 3D 00000005 R mov bh, A4-2 ;Поместить в регистр BH байт памяти с адресом A4-2

0000000C B9 00000001 R mov ecx, offset A2 ;Поместить в регистр ECX адрес переменной A2

00000011 66| A1 mov ax, A2 ;Поместить в регистр AX значение переменной A2

00000001 R

00000017 66| 8B 35 mov si, A2+1 ;Поместить в регистр SI слово памяти по адресу A2+1

00000002 R

0000001E E8 00000000 E call ExitProcess ;Вызов функции завершения программы

end main ;Конец программы

**Результаты работы в отладчике программы data в виде таблиц**

|  |  |  |  |
| --- | --- | --- | --- |
| **Переменная** | **Тип** | **Длина (байт)** | **Эффективный адрес** |
| A1 | Байт | 1 | 0 |
| A2 | Слово | 2 | 1 |
| A3 | Двойное слово | 4 | 3 |
| A4 | Байт | 1 | 7 |
| A5 | Двойное слово | 4 | 8 |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A1 | A2 |  | A3 |  |  |  | A4 | A5 |  |  |  |
| 1Bh | E5h | 0Ah | 56h | 34h | 12h | 00h | FEh | 07h | 20h | 40h | 00h |

402000h 402001h 402002h 402003h 402004h 402005h 402006h 402007h 402008h 402009h 40200Ah 40200Bh

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | Команда | Код машинной команды (по листингу) | Код машинной команды (по дампу) | Длина машинной команды (байт) | Значения регистров | | | | |
| EAX | EBX | ECX | ESI | EIP |
| 1 | mov bl, A1 | 8A 1D 00000000 R | 8A1D 00204000 | 6 | 0019FFCC | 00282000 | 00401000 | 00401000 | 00401000 |
| 2 | mov bh, A4-2 | 8A 3D 00000005 R | 8A3D 05204000 | 6 | 0019FFCC | 0028201B | 00401000 | 00401000 | 00401006 |
| 3 | mov ecx, offset A2 | B9 00000001 R | B9 01204000 | 5 | 0019FFCC | 0028121B | 00401000 | 00401000 | 0040100C |
| 4 | mov ax, A2 | 66| A1 00000001 R | 66:A1 01204000 | 6 | 00190AE5 | 0028121B | 00402001 | 00401000 | 00401011 |
| 5 | mov si, A2+1 | 66| 8B 35 00000002 R | 66:8B35 02204000 | 7 | 00190AE5 | 0028121B | 00402001 | 00401000 | 00401017 |

**Тексты разработанных командных файлов и их назначение**

runasm.bat - данный командный файл предназначен для компилирования, линкования и выполнения исходной программы.

rem Командный файл для выполнения EXE программ

rem Программист Баклашкин Алексей

@echo off

    if exist "%1.obj" del "%1.obj"

    if exist "%1.exe" del "%1.exe"

echo Ассемблирование

i:\masm32\bin\ml /c /coff /Sn /Fl"%1.lst" "%1.asm"

    if errorlevel 1 goto errasm

echo Редактирование связей

i:\masm32\bin\PoLink /SUBSYSTEM:WINDOWS "%1.obj"

    if errorlevel 1 goto errlink

    dir "%1.\*"

    goto TheEnd

:errlink

    echo \_

    echo Link error

    goto TheEnd

:errasm

    echo \_

    echo Assembly Error

    goto TheEnd

:TheEnd

    echo Выполнение

    "%1.exe"

dbgasm.bat - данный командный файл предназначен не только для компилирования и линкования, но и для запуска загрузочного файла исходной программы в отладчике после предыдущих действий.

rem Командный файл для выполнения EXE программ

rem Программист Баклашкин Алексей

@echo off

    if exist "%1.obj" del "%1.obj"

    if exist "%1.exe" del "%1.exe"

rem Ассемблирование

i:\masm32\bin\ml /c /coff /Sn /Fl"%1.lst" "%1.asm"

    if errorlevel 1 goto errasm

rem Редактирование связей

i:\masm32\bin\polink /SUBSYSTEM:WINDOWS "%1.obj"

    if errorlevel 1 goto errlink

    dir "%1.\*"

rem Дебаггинг

ollydbg "%1.exe"

    goto TheEnd

:errlink

    echo \_

    echo Link error

    goto TheEnd

:errasm

    echo \_

    echo Assembly Error

    goto TheEnd

:TheEnd

    echo Выполнение

    "%1.exe"