**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра математического обеспечения и применения ЭВМ**

**ОТЧЕТ**

**по лабораторной работе №1**

**по дисциплине «Организация ЭВМ и систем»**

**Тема:**

**Освоение трансляции, выполнения и отладки программ на**

**языке Ассемблера процессора Intel X86.**

|  |  |
| --- | --- |
| Студент гр. 3344 | Тукалкин В.А. |
| Преподаватель | Фирсов М.А. |

Санкт-Петербург

2024

**Цель работы**

Познакомиться со средой DOS и изучить базовые понятия в ассеблере.

**Задание**

**Часть 1.**

0. Подготовить среду для запуска исполняемых файлов DOS. Загрузить файл hello1.asm в каталог \MASM.

1. Просмотреть программу в режиме редактирования, разобраться в структуре и реализации каждого сегмента программы. Непонятные фрагменты прояснить у преподавателя. Строку-приветствие преобразовать в соответствии со своими личными данными.

2. Протранслировать программу с помощью строки

> masm имя\_файла.asm

c созданием объектного файла имя\_файла.obj и файла диагностических сообщений

(файла листинга) имя\_файла.lst. Объяснить и исправить синтаксические ошибки, если они будут обнаружены транслятором и представлены в файле листинга. Повторить трансляцию программы до получения объектного файла (модуля).

3. Скомпоновать загрузочный модуль (имя\_файла.exe) с помощью строки

> link имя\_файла.obj

c созданием загрузочного модуля (имя\_файла.exe) и файла карты памяти (имя\_файла.map). По карте памяти оценить размещение и длину сегментов программы.

4. Выполнить программу в автоматическом режиме путём набора строки

> имя\_файла.exe

и убедиться в её работоспособности (результат выполнения просмотреть в режиме отображения экрана пользователя, получаемого набором клавиш ctrl^O);

5. Выполнить программу hello1 в пошаговом режиме под управлением отладчика:

> afd имя\_файла.exe

с фиксацией содержимого используемых регистров и ячеек памяти до и после выполнения команды. Записать содержимое всех регистров процессора, включая сегментные, перед выполнением 1-ой команды. Дальнейшие результаты прогона программы под управлением отладчика должны быть представлены в виде, показанном на примере одной команды в табл.1.

**Часть 2.**

1. Просмотреть программу hello2.asm в режиме редактирования, разобраться в структуре и реализации каждого сегмента программы. Обратить внимание на использование полного описания сегментов и на использование процедуры для вывода строки. Непонятные фрагменты прояснить у преподавателя. Строки-приветствия преобразовать в соответствии со своими пожеланиями и личными данными.

2. Протранслировать программу hello2.asm с помощью транслятора masm c созданием объектного файла и файла диагностических сообщений (файла листинга). Объяснить и исправить синтаксические ошибки, если они будут обнаружены транслятором. Повторить трансляцию программы до получения объектного модуля.

3. Скомпоновать загрузочный модуль hello2.exe с помощью компоновщика Link.

4. Выполнить программу в автоматическом режиме и убедиться в её работоспособности.

5. Выполнить программу hello2 в пошаговом режиме под управлением отладчика afd с фиксацией содержимого используемых регистров и ячеек памяти до и после выполнения команд. Аналогично пункту 5 части 1.

**Основные теоретические положения**

Трансляция:

* Процесс трансляции включает преобразование исходного кода на ассемблере в машинный код с помощью ассемблера.
* Основные этапы: лексический анализ, синтаксический анализ и генерация машинного кода.

Выполнение программ: программа на ассемблере выполняется процессором, который интерпретирует машинные команды.

Отладка:

* Отладка - это процесс поиска и исправления ошибок в программе.
* Используются инструменты, такие как отладчики (например, AFD), которые позволяют пошагово выполнять программу и анализировать состояние регистров и памяти.

Регистры:

* Регистры используются для хранения временных данных и адресов.
* Понимание работы с регистрами критически важно для оптимизации кода.

Сегменты:

* Физически сегмент представляет собой область памяти, занятую командами и (или) данными.
* Основные сегменты: CS – сегмент кода, DS – сегмент данных, SS – сегмент стека

Команда ASSUME в ассемблере используется для указания компилятору или ассемблеру, какие сегменты памяти соответствуют определённым логическим именам. Это помогает ассемблеру правильно интерпретировать адреса и указывает, какие сегменты следует использовать для выполнения операций.

Команда DOSSEG в ассемблере используется для установки режима сегментации, который позволяет компилятору автоматически размещать сегменты в памяти так, чтобы они занимали минимально возможное количество места.

Разница между ASSUME И DOSSEG:

- ASSUME управляет тем, какие сегменты используются в коде.

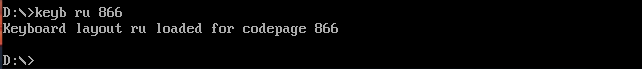
- DOSSEG управляет размещением этих сегментов в памяти.

Файл листинга содержит информацию о структуре программы, такие как исходный код, метки, директивы ассемблера и комментарии. Он включает следующие элементы:

* Командные строки (строки кода).
* Адреса операций.
* Комментарии, которые объясняют назначение отдельных строк кода.
* Метки для обозначения различных участков кода.
* Директивы, определяющие сегменты программы и их расположение в памяти.

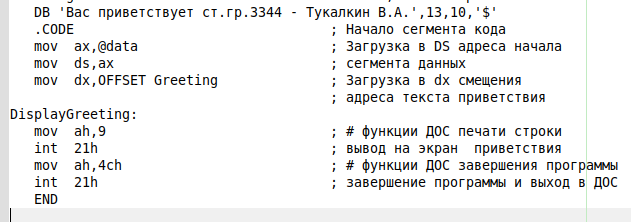
**Выполнение работы**

Подключение русификатора (переключение на кодировку CP866 для работы с буквами русского алфавита).

**Часть 1.**

**Программа hello1.asm:**

1. Изменено приветствие:

**

2. Трансляция программы hello1.asm:

Команда для запуска трансляции: masm hello1.asm.

По команде порождаются объектный файл и файл листинга:

- hello1.obj

- l1.lst.

Файл листинга приводится в приложении.



3. Компоновка загрузочного модуля:

Команда для запуска компоновки: link hello1.obj.

По команде порождаются исполняемый файл и карта памяти:

- hello1.exe

- h1.map

Карта памяти приводится в приложении.



4. Выполнение программы в автоматическом режиме:

Команда для запуска программы: hello1.exe



5. Выполнение программы в отладчике:

Команда для запуска программы в режиме отладки: afd hello1.exe



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Адрес команды | Cимволический код команды | 16-ричный код команды | Содержание регистров и ячеек памяти | |
| До вып. команды | После вып. команды |
| 0010 | MOV AX, 11AE | B8AE11 | AX = 0  IP = 0010 | AX = 11AE  IP = 0013 |
| 0013 | MOV DS, AX | 8ED8 | DS = 119C  IP = 0013 | DS = 11AE  IP = 0015 |
| 0015 | MOV DX, 0000 | BA0000 | DX = 0000  IP = 0015 | DX = 0000  IP = 0018 |
| 0018 | MOV AH, 09 | B409 | AX = 11AE  IP = 0018 | AX = 09AE  IP = 001A |
| 001A | INT 21 | CD21 | IP = 001A | IP = 001C |
| 001C | MOV AH, 4C | B44C | AX = 09AE  IP = 001C | AX = 4CAE  IP = 001E |
| 001E | INT 21 | CD21 | IP = 001E | IP = 0010 |

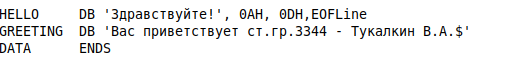
Сообщение отладчика после выполнения программы:



**Часть 2.**

**Программа hello2.asm:**

1. Изменено приветствие:



2. Трансляция программы hello2.asm:

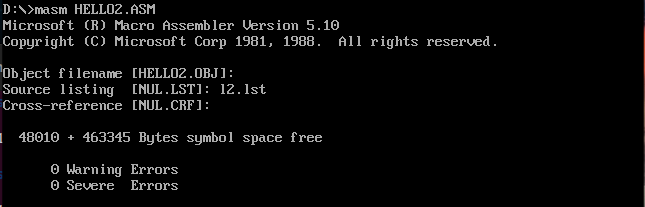
Команда для запуска трансляции: masm hello2.asm.

По команде порождаются объектный файл и файл листинга:

- hello2.obj

- l2.lst.

Файл листинга приводится в приложении.



3. Компоновка загрузочного модуля:

Команда для запуска компоновки: link hello2.obj.

По команде порождаются исполняемый файл и карта памяти:

- hello2.exe

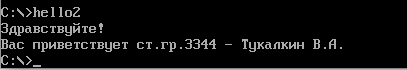
- h2.map

Карта памяти приводится в приложении.



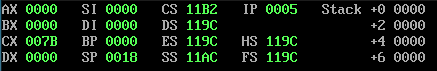
4. Выполнение программы:

Команда для запуска программы: hello2.exe



5. Выполнение программы в отладчике:

Команда для запуска программы в режиме отладки: afd hello2.exe

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Адрес команды | Cимволический код команды | 16-ричный код команды | Содержание регистров и ячеек памяти | |
| До вып. команды | После вып. команды |
| 0005 | PUSH DS | 1E | SP = 0018  IP = 0005  Stack:  +0 0000  +2 0000  +4 0000  +6 0000 | SP = 0016  IP = 0006  Stack:  +0 119C  +2 0000  +4 0000  +6 0000 |
| 0006 | SUB AX, AX | 2BC0 | AX = 0000  IP = 0006 | AX = 0000  IP = 0008 |
| 0008 | PUSH AX | 50 | SP = 0016  IP = 0008  Stack:  +0 119C  +2 0000  +4 0000  +6 0000 | SP = 0014  IP = 0009  Stack:  +0 0000  +2 119C  +4 0000  +6 0000 |
| 0009 | MOV AX, 11AE | B8AE11 | AX = 0000  IP = 0009 | AX = 11AE  IP = 000C |
| 000C | MOV DS, AX | 8ED8 | DS = 119C  IP = 000C | DS = 11AE  IP = 000E |
| 000E | MOV DX, 0000 | BA0000 | DX = 0000  IP = 000E | DX = 0000  IP = 0011 |
| 0011 | CALL 0000  (WriteMsg) | E8ECFF | SP = 0014  IP = 0011  Stack:  +0 0000  +2 119C  +4 0000  +6 0000 | SP = 0012  IP = 0000  Stack:  +0 0014  +2 0000  +4 119C  +6 0000 |
| 0000 | MOV AH, 9 | B409 | AX = 11AE  IP = 0000 | AX = 09AE  IP = 0002 |
| 0002 | INT 21 | CD21 | IP = 0002 | IP = 0004 |
| 0004 | RET | C3 | SP = 0012  IP = 0004  Stack:  +0 0014  +2 0000  +4 119C  +6 0000 | SP = 0014  IP = 0014  Stack:  +0 0000  +2 119C  +4 0000  +6 0000 |
| 0014 | MOV DX, 0010 | BA1000 | DX = 0000  IP = 0014 | DX = 0010  IP = 0017 |
| 0017 | CALL 0000  (WriteMsg) | E8ECFF | SP = 0014  IP = 0017  Stack:  +0 0000  +2 119C  +4 0000  +6 0000 | SP = 0012  IP = 0000  Stack:  +0 001A  +2 0000  +4 119C  +6 0000 |
| 0000 | MOV AH, 9 | B409 | AX = 09AE  IP = 0000 | AX = 09AE  IP = 0002 |
| 0002 | INT 21 | CD21 | IP = 0002 | IP = 0004 |
| 0004 | RET | C3 | SP = 0012  IP = 0004  Stack:  +0 001A  +2 0000  +4 119C  +6 0000 | SP = 0014  IP = 001A  Stack:  +0 0000  +2 119C  +4 0000  +6 0000 |
| 001A | RET Far | CB | SP = 0014  CS = 11B3  IP = 001A  Stack:  +0 0000  +2 119C  +4 0000  +6 0000 | SP = 0018  CS = 119C  IP = 0000  Stack:  +0 0000  +2 0000  +4 0000  +6 0000 |
| 0000 | INT 20 | CD20 | AX = 09AE  CX = 007B  CS = 119C  DS = 11AE | AX = 0000  CX = 0000  CS = 11B2  DS = 119C |

Сообщение отладчика после выполнения программы:

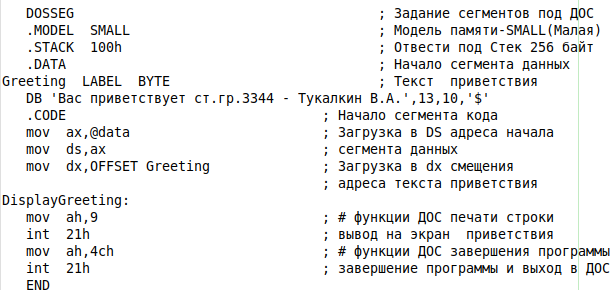


**Вывод**

В ходе данной лабораторной работы были выполнены анализ, трансляция, отладка и выполнение двух программ на языке Ассемблера.

**Приложение**

Код программы hello1.asm:



Листинг l1.lst:

#Microsoft (R) Macro Assembler Version 5.10 9/11/24 14:32:09

Page 1-1

; HELLO1.ASM - упрощенная версия учебной прогр

аммы лаб.раб. N1

; по дисциплине "Архитектура комп

ьютера"

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Назначение: Программа формирует и выводит на

экран приветствие

; пользователя с помощью функции ДО

С "Вывод строки"

; (номер 09 прерывание 21h), котора

я:

; - обеспечивает вывод на экран ст

роки символов,

; заканчивающейся знаком "$";

; - требует задания в регистре ah

номера функции=09h,

; а в регистре dx - смещения а

дреса выводимой

; строки;

; - использует регистр ax и не

сохраняет его

; содержимое.

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DOSSEG

; Задание сегментов под ДОС

.MODEL SMALL

; Модель памяти-SMALL(Малая)

.STACK 100h

; Отвести под Стек 256 байт

.DATA

; Начало сегмента данных

0000 Greeting LABEL BYTE

; Текст приветствия

0000 95 E3 A9 21 21 21 DB ''Вас приветствует ст.гр.3344 - Тук

алкин В.А.',13,10,'$'

0D 0A 24

.CODE ; Начал

о сегмента кода

0000 B8 ---- R mov ax,@data ; Загру

зка в DS адреса начала

0003 8E D8 mov ds,ax ; сегме

нта данных

0005 BA 0000 R mov dx,OFFSET Greeting ; Загру

зка в dx смещения

; адрес

а текста приветствия

0008 DisplayGreeting:

0008 B4 09 mov ah,9 ; # фун

кции ДОС печати строки

000A CD 21 int 21h ; вывод

на экран приветствия

#Microsoft (R) Macro Assembler Version 5.10 9/11/24 14:32:09

Page 1-2

000C B4 4C mov ah,4ch ; # фун

кции ДОС завершения программы

000E CD 21 int 21h ; завер

шение программы и выход в ДОС

END

#Microsoft (R) Macro Assembler Version 5.10 9/11/24 14:32:09

Symbols-1

Segments and Groups:

N a m e Length Align Combine Class

DGROUP . . . . . . . . . . . . . GROUP

\_DATA . . . . . . . . . . . . 0009 WORD PUBLIC 'DATA'

STACK . . . . . . . . . . . . 0100 PARA STACK 'STACK'

\_TEXT . . . . . . . . . . . . . 0010 WORD PUBLIC 'CODE'

Symbols:

N a m e Type Value Attr

DISPLAYGREETING . . . . . . . . L NEAR 0008 \_TEXT

GREETING . . . . . . . . . . . . L BYTE 0000 \_DATA

@CODE . . . . . . . . . . . . . TEXT \_TEXT

@CODESIZE . . . . . . . . . . . TEXT 0

@CPU . . . . . . . . . . . . . . TEXT 0101h

@DATASIZE . . . . . . . . . . . TEXT 0

@FILENAME . . . . . . . . . . . TEXT hello1

@VERSION . . . . . . . . . . . . TEXT 510

33 Source Lines

33 Total Lines

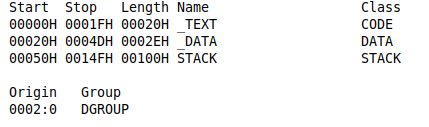
19 Symbols

47994 + 461313 Bytes symbol space free

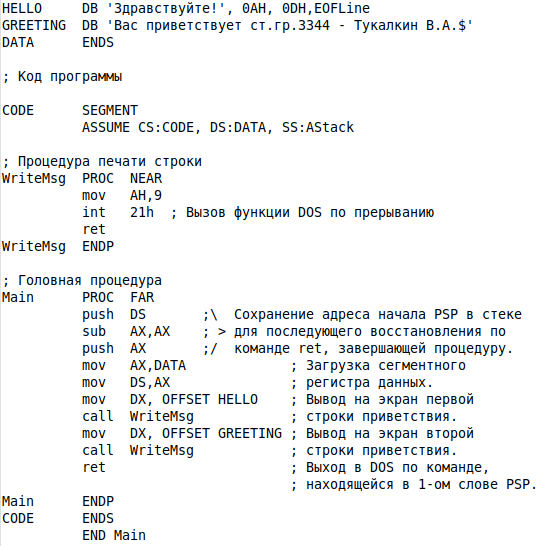
0 Warning Errors

0 Severe Errors

Карта памяти h1.map



Код программы hello2.asm:



Листинг l2.lst:

#Microsoft (R) Macro Assembler Version 5.10 9/11/24 23:21:56

Page 1-1

; HELLO2 -Учебная программа N2 лаб.раб.#1 по ди

сциплине "Архитектура компьютера"

; Программа использует процедуру для п

ечати строки

;

; ТЕКСТ ПРОГРАММЫ

= 0024 EOFLine EQU '$' ; Определение символь

ной константы

; "Конец строки"

; Стек программы

0000 AStack SEGMENT STACK

0000 000C[ DW 12 DUP(?) ; Отводится 12 слов п

амяти

????

]

0018 AStack ENDS

; Данные программы

0000 DATA SEGMENT

; Директивы описания данных

0000 87 A4 E0 A0 A2 E1 HELLO DB 'Здравствуйте!', 0AH, 0DH,EOFLine

E2 A2 E3 A9 E2 A5

21 0A 0D 24

0010 82 A0 E1 20 AF E0 GREETING DB 'Вас приветствует ст.гр.3344 - Тук

алкин В.А.$'

A8 A2 A5 E2 E1 E2

A2 E3 A5 E2 20 E1

E2 2E A3 E0 2E 33

33 34 34 20 2D 20

92 E3 AA A0 AB AA

A8 AD 20 82 2E 80

2E 24

003C DATA ENDS

; Код программы

0000 CODE SEGMENT

ASSUME CS:CODE, DS:DATA, SS:AStack

; Процедура печати строки

0000 WriteMsg PROC NEAR

0000 B4 09 mov AH,9

0002 CD 21 int 21h ; Вызов функции DOS по пре

рыванию

0004 C3 ret

0005 WriteMsg ENDP

#Microsoft (R) Macro Assembler Version 5.10 9/11/24 23:21:56

Page 1-2

; Головная процедура

0005 Main PROC FAR

0005 1E push DS ;\ Сохранение адреса

начала PSP в стеке

0006 2B C0 sub AX,AX ; > для последующего в

осстановления по

0008 50 push AX ;/ команде ret, завер

шающей процедуру.

0009 B8 ---- R mov AX,DATA ; Загрузка

сегментного

000C 8E D8 mov DS,AX ; регистра

данных.

000E BA 0000 R mov DX, OFFSET HELLO ; Вывод на

экран первой

0011 E8 0000 R call WriteMsg ; строки пр

иветствия.

0014 BA 0010 R mov DX, OFFSET GREETING ; Вывод на

экран второй

0017 E8 0000 R call WriteMsg ; строки пр

иветствия.

001A CB ret ; Выход в D

OS по команде,

; находящей

ся в 1-ом слове PSP.

001B Main ENDP

001B CODE ENDS

END Main

#Microsoft (R) Macro Assembler Version 5.10 9/11/24 23:21:56

Symbols-1

Segments and Groups:

N a m e Length Align Combine Class

ASTACK . . . . . . . . . . . . . 0018 PARA STACK

CODE . . . . . . . . . . . . . . 001B PARA NONE

DATA . . . . . . . . . . . . . . 003C PARA NONE

Symbols:

N a m e Type Value Attr

EOFLINE . . . . . . . . . . . . NUMBER 0024

GREETING . . . . . . . . . . . . L BYTE 0010 DATA

HELLO . . . . . . . . . . . . . L BYTE 0000 DATA

MAIN . . . . . . . . . . . . . . F PROC 0005 CODE Length = 0016

WRITEMSG . . . . . . . . . . . . N PROC 0000 CODE Length = 0005

@CPU . . . . . . . . . . . . . . TEXT 0101h

@FILENAME . . . . . . . . . . . TEXT hello2

@VERSION . . . . . . . . . . . . TEXT 510

52 Source Lines

52 Total Lines

13 Symbols

48002 + 461305 Bytes symbol space free

0 Warning Errors

0 Severe Errors

Карта памяти h2.map

