Санкт-Петербургский государственный

электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)

Кафедра МО ЭВМ

**­Отчет**

по лабораторной работе №1

«Освоение трансляции, выполнения и отладки программ на

языке Ассемблера процессора Intel X86»

по дисциплине «Архитектура компьютера»

**Выполнил**: Виноградов К. А.

**Группа**: 6304

**Факультет**: КТИ

**Преподаватель**: Кириянчиков В. А.

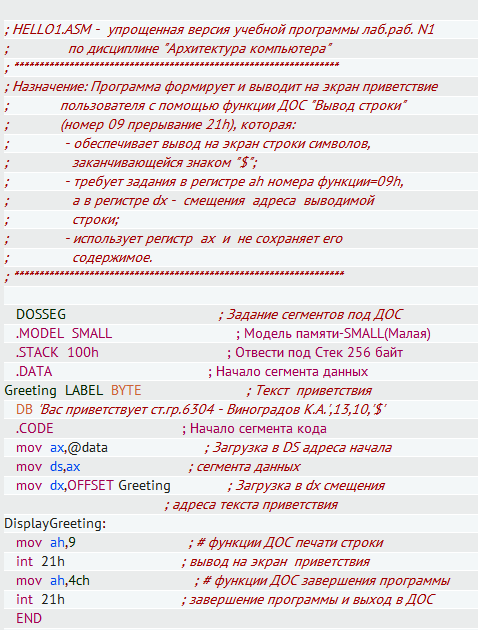
*Санкт-Петербург*

*2017*

**Цель работы:** освоить трансляцию и отладку простейших программ на языке Ассемблера.

**Ход работы:**

**Часть 1.**

1. Загрузим файл hello1.asm и просмотрим программу в режиме редактирования в программе AkelPad. Строку-приветствие преобразуем в соответствии со своими личными данными.: 

2. Запустим программу DOSBox 0.74, смонтируем диск Z: как директорию C:\MASM командой

> mount C C:\MASM

и перейдя в C, протранслируем программу командой

> masm hello1.asm

c созданием объектного файла hello1.obj и файла листинга hello1.lst



Содержание файла листинга hello1.lst:

Microsoft (R) Macro Assembler Version 5.10 9/10/17 19:08:50

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 82 A0 E1 20 AF E0 DB 'Вас приветствует ст.гр.6304 - Виноградов

К.А.',13,10,'$'

A8 A2 A5 E2 E1 E2

A2 E3 A5 E2 20 E1

E2 2E A3 E0 2E 36

33 30 34 20 2D 20

82 A8 AD AE A3 E0

A0 A4 AE A2 20 8A

2E 80 2E 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

Microsoft (R) Macro Assembler Version 5.10 9/10/17 19:08:50

Page 1-2

; 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

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 9/10/17 19:08:50

Symbols-1

Segments and Groups:

N a m e Length Align Combine Class

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

\_DATA . . . . . . . . . . . . 0030 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

Командой

> link hello1.obj

cкомпануем загрузочный модуль hello1.exe и файл карты памяти hello1.map :



Содержание файла карты памяти hello1.map:

Start Stop Length Name Class

00000H 0001FH 00020H \_TEXT CODE

00020H 0004FH 00030H \_DATA DATA

00050H 0014FH 00100H STACK STACK

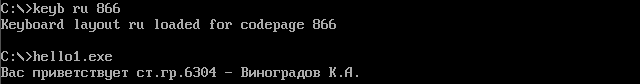
Origin Group

0002:0 DGROUP

3. Запустим программу с помощью команды

>hello1.exe

Для того, чтобы получить текст вывода на кириллице введем команду

> keyb ru 866 ,

4. Создадим версию hello1\_2 программы, дополненную еще одним оператором вывода другой строки приветствия.



5. Протранслируем программу hello1\_2, скомпонуем загрузочный модуль, выполним ее в автоматическом режиме и убедимся в правильной работе.





6. Выполним программу hello1 в пошаговом режиме под управлением

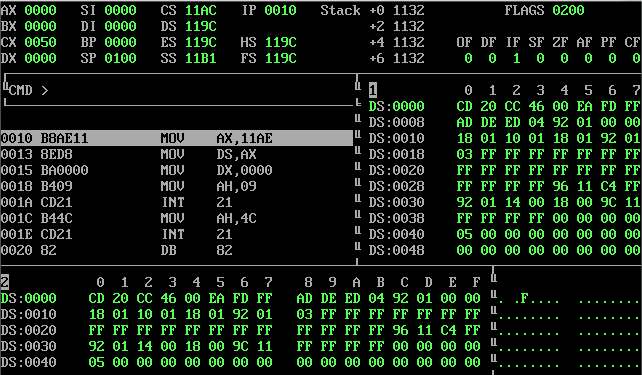
отладчика:

> afd hello1.exe

с фиксацией содержимого используемых регистров и ячеек памяти до и после выполнения команды:

До выполнения первой команды:

|  |  |  |  |
| --- | --- | --- | --- |
| AX = 0000 | SI = 0000 | CS = 11AC | IP = 0010 |
| BX = 0000 | DI = 0000 | DS = 119C |  |
| CX = 0050 | BP = 0000 | ES = 119C | HS = 119C |
| DX = 0000 | SP = 0100 | SS = 11B1 | FS = 119C |

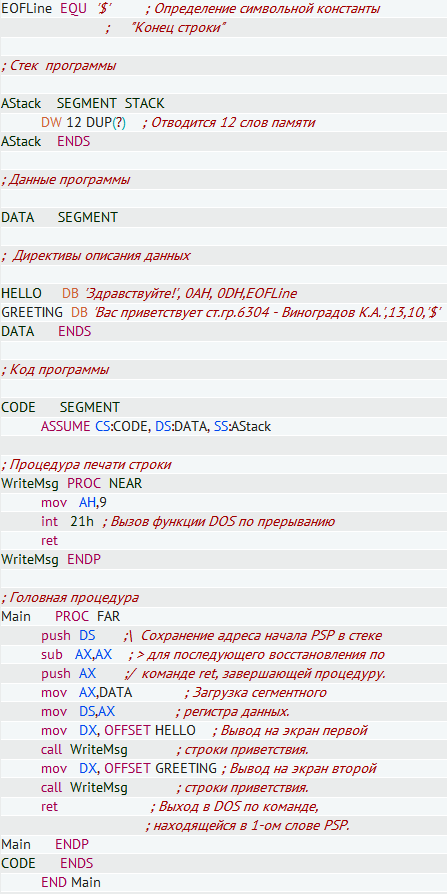


7. Обычные команды выполняем по F1, а прерывания Int - по F2, чтобы не входить внутрь обработчика прерываний.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Адрес команды | Символический код команды | 16-ричный код команды | Содержимое рег-ов и яч. пам. до вып. команд | Содержимое рег-ов и яч. пам. после вып. команд |
| 0010 | MOV AX,11AE | B8AE11 | AX = 0000  IP = 0010 | AX = 11AE  IP = 0013 |
| 0013 | MOV DS,AX | 8ED8 | AX = 11AE  DS = 119C  IP = 0013 | AX = 11AE  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 | AX = 4CAE  DS = 11AE  IP = 001E | AX = 0000  DS = 119C  IP = 0010 |

8. Выходим из отладчика - по команде quit.

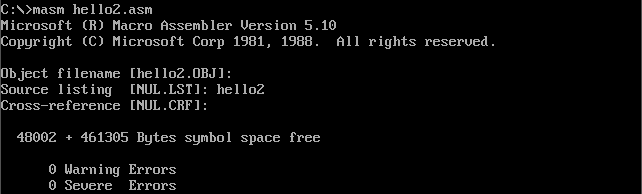
**Часть 2.**

1.Загрузим файл hello1\_2.asm и просмотрим программу в режиме редактирования

2.Протранслируем программу с помощью команды

> masm hello2.asm

c созданием объектного файла hello2.obj и файла листинга hello2.lst.



Содержание файла листинга hello2.lst:

Microsoft (R) Macro Assembler Version 5.10 9/12/17 18:07:41

Page 1-1

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

дисциплине "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.6304 - Вин

оградов К.А.',13,10,'$' A8 A2 A5 E2 E1 E2

A2 E3 A5 E2 20 E1

E2 2E A3 E0 2E 34

33 35 30 20 2D 20

88 A2 A0 AD AE A2

20 88 2E 88 2E 24

003A 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

; Головная процедура

Microsoft (R) Macro Assembler Version 5.10 9/12/17 18:07:41

Page 1-2

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 9/12/17 18:07:41

Symbols-1

Segments and Groups:

N a m e Length Align Combine Class

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

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

DATA . . . . . . . . . . . . . . 003A 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

3. C помощью строки

> link hello2.obj

cкомпануем загрузочный модуль hello2.exe и файл карты памяти hello2.map.



Содержание файла карты памяти hello2.map:

Start Stop Length Name Class

00000H 00017H 00018H ASTACK

00020H 00059H 0003AH DATA

00060H 0007AH 0001BH CODE

Program entry point at 0006:0005

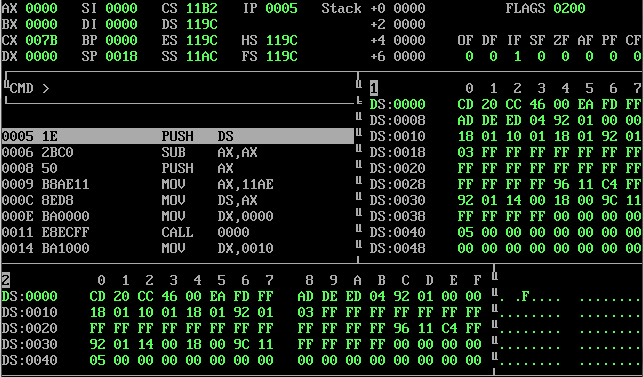
4. Выполним программу в автоматическом режиме и убедимся в ее работоспособности:



4. Выполним программу hello2.exe в пошаговом режиме под управлением отладчика afd с фиксацией содержимого используемых регистров и ячеек памяти до и после выполнения команд:

До выполнения первой команды:

|  |  |  |  |
| --- | --- | --- | --- |
| AX = 0000 | SI = 0000 | CS = 11B2 | IP = 0005 |
| BX = 0000 | DI = 0000 | DS = 119C |  |
| CX = 007B | BP = 0000 | ES = 119C | HS = 119C |
| DX = 0000 | SP = 0018 | SS = 11AC | FS = 119C |



Обычные команды выполняем по F1, а прерывания Int - по F2, чтобы не входить внутрь обработчика прерываний.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Адрес ком. | Символический код команды | 16-ричный код | Содержимое рег-ов и яч. пам. до вып-я ком. | | Содержимое рег-ов и яч. пам. после… | |
| 0005 | PUSH DS | 1E | SP = 0018  IP = 0005  DS = 119C | +0 0000  +2 0000  +4 0000  +6 0000 | SP = 0016  IP = 0006  DS = 119C | +0 119C  +2 0000  +4 0000  +6 0000 |
| 0006 | SUB AX,AX | 2BC0 | IP = 0006  AX = 0000 | | IP = 0008  AX = 0000 | |
| 0008 | PUSH AX | 50 | SP = 0016  IP = 0008  AX = 0000 | +0 119C  +2 0000  +4 0000  +6 0000 | SP = 0014  IP = 0009  AX = 0000 | +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 | 8EDE | DS = 119C  IP = 000C | | DS = 11AE  IP = 000E | |
| 000E | MOV DX,0000 | BA0000 | IP = 000E  DX = 0000 | | IP = 0011  DX = 0000 | |
| 0011 | CALL 0000 | E8ECFF | AX = 11AE  DS = 11AE  SP = 0014  IP = 0011 | +0 0000  +2 119C  +4 0000  +6 0000 | AX = 11AE  DS = 11AE  SP = 0012  IP = 0000 | +0 0014  +2 0000  +4 119C  +6 0000 |
| 0000 | MOV AH,09 | B409 | AX = 11AE  IP = 0000 | | AX = 09AE  IP = 0002 | |
| 0002 | INT 21 | CD21 | IP = 0002 | | IP = 0004 | |
| 0004 | RET | C3 | AX = 09AE  DS = 11AE  SP = 0012  IP = 0004 | +0 0014  +2 0000  +4 119C  +6 0000 | AX = 09AE  DS = 11AE  SP = 0014  IP = 0014 | +0 0000  +2 119C  +4 0000  +6 0000 |
| 0014 | MOV DX,0010 | BA1000 | DX = 0000  IP = 0014 | | DX = 0010  IP = 0017 | |
| 0017 | CALL 0000 | E8E6FF | SP = 0014  IP = 0017 | +0 0000  +2 119C  +4 0000  +6 0000 | SP = 0012  IP = 0000 | +0 001A  +2 0000  +4 119C  +6 0000 |
| 0000 | MOV AH,09 | B409 | AX = 09AE  IP = 0000 | | AX = 09AE  IP = 0002 | |
| 0002 | INT 21 | CD21 | IP = 0002 | | IP = 0004 | |
| 0004 | RET | C3 | SP = 0012  IP = 0004 | +0 001A  +2 0000  +4 119C  +6 0000 | SP = 0014  IP = 001A | +0 0000  +2 119C  +4 0000  +6 0000 |
| 001A | RET Far | CB | SP = 0014  IP = 001A | +0 0000  +2 119C  +4 0000  +6 0000 | SP = 0012  IP = 0000 | +0 0000  +2 0000  +4 0000  +6 0000 |
| 0000 | INT 20 | CD20 | AX = 09AE  DS = 11AE  IP = 0000 | | AX = 0000  DS = 119C  IP = 0005 | |

6. Выходим из отладчика - по команде quit.