

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Организация систем и ЭВМ»
Тема: Освоение трансляции, выполнения и отладки программ на
языке Ассемблера процессора Intel X86

Студент гр. 3388

Еникеев А.А.

Преподаватель

Фирсов М.А.

Санкт-Петербург

2024

Цель работы

Ознакомиться с тем, как происходит трансляция, отладка и выполнение программ на языке Ассемблера.

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`
с 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`
с 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.

Табл. 1

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			до вып. команды	после вып. команды
0003	mov ds,ax	8E D8	(ax)= 2D87 (ds)= 2D75 (ip)= 0003	(ax)= 2D87 *(ds)= 2D87 *(ip)= 0005

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 nasm с 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 6 части 1.

Основные теоретические положения

Всего в процессоре Intel 8086 имеется 14 16-разрядных регистров.

8 регистров общего назначения (AX, BX, CX, DX):

- Регистр AX является основным сумматором и применяется для всех операций ввода-вывода, некоторых операций над строками и некоторых арифметических операций. Например, команды умножения, деления и сдвига предполагают использование регистра AX. Некоторые команды генерируют более эффективный код, если они имеют ссылки на регистр AX.
- Регистр BX является базовым регистром. Это единственный регистр общего назначения, который может использоваться в качестве "индекса" для расширенной адресации. Другое общее применение его - вычисления.
- Регистр CX является счетчиком. Он необходим для управления числом повторений циклов и для операций сдвига влево или вправо. Регистр CX используется также для вычислений.
- Регистр DX является регистром данных. Он применяется для некоторых операций ввода/вывода и тех операций умножения и деления над большими числами, которые используют регистровую пару DX и AX.

4 сегментных регистра (CS, SS, DS, ES):

- Регистр сегмента кода CS содержит начальный адрес сегмента кода. Этот адрес плюс величина смещения в командном указателе (IP) определяет адрес команды, которая должна быть выбрана для

выполнения. Для обычных программ нет необходимости делать ссылки на регистр CS.

- Регистр сегмента данных DS содержит начальный адрес сегмента данных. Этот адрес плюс величина смещения, определенная в команде, указывают на конкретную ячейку в сегменте данных.
- Регистр сегмента стека SS содержит начальный адрес в сегменте стека.
- Некоторые операции над строками используют дополнительный сегментный регистр для управления адресацией памяти. В данном контексте регистр ES связан с индексным регистром DI. Если необходимо использовать регистр ES, ассемблерная программа должна его инициализировать.

2 индексных регистра (SI, DI):

- SI является индексом источника и применяется для некоторых операций над строками. В данном контексте регистр SI связан с регистром DS.
- DI является индексом назначения и применяется также для строковых операций. В данном контексте регистр DI связан с регистром ES.

2 указательных (BP, SP):

- SP указатель стека обеспечивает использование стека в памяти, позволяет временно хранить адреса и иногда данные. Этот регистр связан с регистром SS для адресации стека.
- Указатель базы BP облегчает доступ к параметрам: данным и адресам переданным через стек.

Регистр IP содержит смещение на команду, которая должна быть выполнена. Обычно этот регистр в программе не используется, но он может изменять свое значение при использовании отладчика DOS DEBUG для тестирования программы.

Регистр флагов (FLAGS, включает в себя 9 флагов): девять из 16 битов флагового регистра являются активными и определяют текущее состояние

машины и результатов выполнения. Многие арифметические команды и команды сравнения изменяют состояние флагов.

Выполнение работы

Часть 1.

0. Запускаем DOSBox, монтируем директорию MASM как виртуальный диск C в DOSBox, переключаемся на ранее смонтированный диск C, настраиваем раскладку клавиатуры на русскоую (RU) с кодировкой 866. (см. рис. 1)

```
Z:\>mount c: d:\EVM\MASM
Drive C is mounted as local directory d:\EVM\MASM\

Z:\>c:

C:\>keyb ru 866
Keyboard layout ru loaded for codepage 866
```

Рисунок 1

1. Открываем программу HELLO1 с помощью редактора Geany в режиме редактирования, строку-приветствие преобразовываем. (см. рис. 2)

```
16
17     DOSSEG                                ; Задание сегментов под ДОС
18     .MODEL  SMALL                        ; Модель памяти-SMALL (Малая)
19     .STACK  100h                          ; Отвести под Стек 256 байт
20     .DATA                                  ; Начало сегмента данных
21     Greeting LABEL BYTE                  ; Текст приветствия
22     DB 'Вас приветствует ст.гр.3388 - Еникеев А.А.',13,10,'$'
23     .CODE                                ; Начало сегмента кода
24     mov ax,@data                          ; Загрузка в DS адреса начала
25     mov ds,ax                             ; сегмента данных
26     mov dx,OFFSET Greeting               ; Загрузка в dx смещения
27                                           ; адреса текста приветствия
28     DisplayGreeting:
29     mov ah,9                              ; # функции ДОС печати строки
30     int 21h                              ; вывод на экран приветствия
31     mov ah,4ch                            ; # функции ДОС завершения программы
32     int 21h                              ; завершение программы и выход в ДОС
33     END
34
```

Рисунок 2

2. Транслируем программу с помощью команды `masm HELLO1.ASM` с созданием объектного файла `HELLO1.OBJ` и файла диагностических

сообщений (файла листинга) HELLO1.LST. (см. рис. 3). Ошибок транслятором не обнаружено.

```
C:\>masm HELLO1.ASM
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [HELLO1.OBJ]:
Source listing [NUL.LST]: HELLO1.LST
Cross-reference [NUL.CRF]:

47994 + 461313 Bytes symbol space free

0 Warning Errors
0 Severe Errors
```

Рисунок 3

3. Компануем загрузочный модуль (HELLO1.EXE) с помощью команды link HELLO1.OBJ с созданием загрузочного модуля (HELLO1.EXE) и файла карты памяти (HELLO1.MAP). (см. рис. 4). Оценим по карте памяти (см. рис. 5) размещение и длину сегментов программы:

- Сегмент CODE: длина 20H
- Сегмент DATA: длина 2DH
- Сегмент STACK: длина 100H

```
C:\>link HELLO1.OBJ

Microsoft (R) Overlay Linker Version 3.64
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

Run File [HELLO1.EXE]:
List File [NUL.MAP]: HELLO1.MAP
Libraries [.LIB]:
```

Рисунок 4

	Start	Stop	Length	Name	Class
2	00000H	0001FH	00020H	_TEXT	CODE
3	00020H	0004CH	0002DH	_DATA	DATA
4	00050H	0014FH	00100H	STACK	STACK
5					
6					
7	Origin	Group			
8	0002:0	DGROUP			

Рисунок 5

4. Выполним программу в автоматическом режиме командой HELLO1.EXE (см. рис. 6)

```
C:\>HELLO1.EXE
Вас приветствует ст.гр.3388 – Еникеев А.А.
```

Рисунок 6

5. Выполним программу HELLO1 в пошаговом режиме под управлением отладчика. Значения регистров до выполнения программы см. на рисунке 7. Фиксация используемых регистров на каждом шаге приведена в табл. 2.

```
AX 0000 SI 0000 CS 11AC IP 0010 Stack +0 1132 FLAGS 0200
BX 0000 DI 0000 DS 119C          +2 1132
CX 004D BP 0000 ES 119C HS 119C      +4 1132 OF DF IF SF ZF AF PF CF
DX 0000 SP 0100 SS 11B1 FS 119C      +6 1132 0 0 1 0 0 0 0 0
```

Рисунок 7

Табл. 2

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			до вып. команды	после вып. команды
0010	MOV AX,11AE	B8AE11	(AX) = 0000 (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	(AX) = 4CAE (DS) = 1AE (IP) = 001E (CX) = 004F	(AX) = 0000 (DS) = 119C (IP) = 0010 (CX) = 0000

Часть 2.

1. Открываем программу HELLO2 с помощью редактора Geany в режиме редактирования, строку-приветствие преобразовываем. (см. рис.8). Полный код программы см. в прил. А.

```
21 HELLO      DB 'Здравствуйте!', 0AH, 0DH,E0FLine
22 GREETING  DB 'Вас приветствует ст.гр.3388 - Еникеев А.А.$'
23 DATA     ENDS
```

Рисунок 8

2. Транслируем программу с помощью команды `masm HELLO2.ASM` с созданием объектного файла `HELLO2.OBJ` и файла диагностических сообщений (файла листинга) `HELLO2.LST`. (см. рис. 9). Ошибок транслятором не обнаружено.

```
C:\>masm HELLO2
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [HELLO2.OBJ]:
Source listing [NUL.LST]: HELLO2.LST
Cross-reference [NUL.CRF]:

48002 + 461305 Bytes symbol space free

    0 Warning Errors
    0 Severe Errors
```

Рисунок 9

3. Компануем загрузочный модуль (`HELLO2.EXE`) с помощью команды `link HELLO2.OBJ`. (см. рис. 10).

```
C:\>link HELLO2.OBJ

Microsoft (R) Overlay Linker Version 3.64
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

Run File [HELLO2.EXE]:
List File [NUL.MAP]: HELLO2.MAP
Libraries [.LIB]:
```

Рисунок 10

4. Запускаем программу в автоматическом режиме командой HELLO2.EXE (см. рис. 11).

```
C:\>HELLO2.EXE
Здравствуйте!
Вас приветствует ст.гр.4350 – Еникеев А.А.
```

Рисунок 11

5. Выполним программу HELLO2 в пошаговом режиме под управлением отладчика. Значения регистров до выполнения программы см. на рисунке 12. Фиксация используемых регистров на каждом шаге приведена в табл. 3.

```
AX 0000 SI 0000 CS 11B2 IP 0005 Stack +0 0000 FLAGS 0200
BX 0000 DI 0000 DS 119C      +2 0000
CX 007B BP 0000 ES 119C HS 119C  +4 0000  DF DF IF SF ZF AF PF CF
DX 0000 SP 0018 SS 11AC FS 119C  +6 0000  0 0 1 0 0 0 0 0
```

Рисунок 12

Табл. 3

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			до вып. команды	после вып. команды
0005	PUSH DS	1E	Stack(+0) = 0000 (IP) = 0005 (SP) = 0018	Stack(+0) = 119C (IP) = 0006 (SP) = 0016
0006	SUB AX,AX	2BC0	(IP) = 0006 (AX) = 0000	(IP) = 0008 (AX) = 0000
0008	PUSH AX	50	(IP) = 0008 (SP) = 0016 Stack(+0) = 119C Stack(+2) = 0000	(IP) = 0009 (SP) = 0014 Stack(+0) = 0000 Stack(+2) = 119C
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	(IP) = 000E	(IP) = 0011

			(DX) = 0000	(DX) = 0000
0011	CALL 0000	E8ECFF	(IP) = 0011 (SP) = 0014 Stack(+0) = 0000 Stack(+2) = 119C Stack(+4) = 0000	(IP) = 0000 (SP) = 0012 Stack(+0) = 0014 Stack(+2) = 0000 Stack(+4) = 119C
0000	MOV AH, 09	B409	(AX)=11AE (IP) = 0000	(AX)=09AE (IP) = 0002
0002	INT 21H	CD21	(IP) = 0002	(IP) = 0004
0004	RET	C3	(IP) = 0004 (SP) = 0012 Stack(+0) = 0014 Stack(+2) = 0000 Stack(+4) = 119C	(IP) = 0014 (SP) = 0014 Stack(+0) = 0000 Stack(+2) = 119C Stack(+4) = 0000
0014	MOV DX,0010	BA1000	(IP) = 0014 (DX) = 0000	(IP) = 0017 (DX) = 0010
0017	CALL 0000	E8E6FF	(IP) = 0017 (SP) = 0014 Stack(+0) = 0000 Stack(+2) = 119C Stack(+4) = 0000	(IP) = 0000 (SP) = 0012 Stack(+0) = 001A
0000	MOV AH, 9	B409	(IP) = 0000 (AX) = 09AE	(IP) = 0002 (AX) = 09AE
0002	INT 21	CD21	(IP) = 0002	(IP) = 0004
0004	RET	C3	(IP) = 0004 (SP) = 0012 Stack(+0) = 001A Stack(+2) = 0000 Stack(+4) = 119C	(IP) = 001A (SP) = 0014 Stack(+0) = 0000 Stack(+2) = 119C Stack(+4) = 0000
001A	RET Far	CB	(IP) = 001A (CS) = 11B2 (SP) = 0014 Stack(+0) = 0000 Stack(+2) = 119C	(IP) = 0000 (CS) = 119C (SP) = 0018 Stack(+0) = 0000 Stack(+2) = 0000
0000	INT 20	CD20	(AX) = 09AE (CS)= 119C (DS) = 11AE (IP) = 0000 (CX) = 007B	(AX) = 0000 (CS)=11B2 (DS) = 119C (IP) = 0005 (CX) = 0000

Выводы

В ходе выполнения лабораторной работы была освоена трансляция, изучено, как происходит выполнение и отладка программ на языке Ассемблер, а также разобраны структуры приведенных в работе программ.

ПРИЛОЖЕНИЕ А

Файл HELLO1.ASM:

```
; HELLO1.ASM - упрощенная версия учебной программы лаб.раб. N1
; по дисциплине "Архитектура компьютера"
; *****
; Назначение: Программа формирует и выводит на экран приветствие
; пользователя с помощью функции ДОС "Вывод строки"
; (номер 09 прерывание 21h), которая:
; - обеспечивает вывод на экран строки символов,
; заканчивающейся знаком "$";
; - требует задания в регистре ah номера функции=09h,
; а в регистре dx - смещения адреса 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
Greeting LABEL BYTE ; Текст приветствия
DB 'Вас приветствует ст.гр.3388 - Еникеев А.А.',13,10','$'
.CODE ; Начало сегмента кода
mov ax,@data ; Загрузка в DS адреса начала
mov ds,ax ; сегмента данных
mov dx,OFFSET Greeting ; Загрузка в dx смещения
; адреса текста приветствия

DisplayGreeting:
mov ah,9 ; # функции ДОС печати строки
int 21h ; вывод на экран приветствия
mov ah,4ch ; # функции ДОС завершения программы
int 21h ; завершение программы и выход в ДОС
END
```

Файл HELLO1.LST:

```
#Microsoft (R) Macro Assembler Version 5.10 9/19/24 11:39:56
Page 1-1
```

```
; HELLO1.ASM - упрощенная версия учебной прог р
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.3388 - Еникеев А.
А.',13,10,'$'

A8 A2 A5 E2 E1 E2
A2 E3 A5 E2 20 E1
E2 2E A3 E0 2E 33
33 38 38 20 2D 20
85 AD A8 AA A5 A5
A2 20 80 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

```

```

a 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      9/19/24 11:39:56
           Symbols-1

```

Segments and Groups:

N a m e	Length	Align	Combine	Class
DGROUP	GROUP			
_DATA	002D	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

```


Файл HELLO1.MAP:

Start	Stop	Length	Name	Class
00000H	0001FH	00020H	_TEXT	CODE
00020H	0004CH	0002DH	_DATA	DATA
00050H	0014FH	00100H	STACK	STACK

Origin Group
0002:0 DGROUP
0001:0 DGROUP

Файл HELLO2.ASM:

; HELLO2 -Учебная программа N2 лаб.раб.#1 по дисциплине "Архитектура компьютера"
;
; Программа использует процедуру для печати строки
;
; ТЕКСТ ПРОГРАММЫ

EOFLine EQU '\$' ; Определение символьной константы
; "Конец строки"

; Стек программы

AStack SEGMENT STACK
DW 12 DUP(?) ; Отводится 12 слов памяти
AStack ENDS

; Данные программы

DATA SEGMENT

; Директивы описания данных

HELLO DB 'Здравствуйте!', 0AH, 0DH,EOFLine
GREETING DB 'Вас приветствует ст.гр.4350 - Еникеев А.А.\$'
DATA ENDS

; Код программы

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

; Процедура печати строки

WriteMsg PROC NEAR
mov AH,9
int 21h ; Вызов функции DOS по прерыванию
ret
WriteMsg ENDP

; Головная процедура

Main PROC FAR
push DS ;\ Сохранение адреса начала PSP в стеке

```

sub  AX,AX    ; > для последующего восстановления по
push AX      ;/ команде ret, завершающей процедуру.
mov  AX,DATA  ; Загрузка сегментного
mov  DS,AX    ; регистра данных.
mov  DX, OFFSET HELLO ; Вывод на экран первой
call WriteMsg ; строки приветствия.
mov  DX, OFFSET GREETING ; Вывод на экран второй
call WriteMsg ; строки приветствия.
ret          ; Выход в DOS по команде,
              ; находящейся в 1-ом слове PSP.

Main  ENDP
CODE  ENDS
      END Main

```

Файл HELLO2.LST:

```

#Microsoft (R) Macro Assembler Version 5.10      9/19/24 16:12:26
                                           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 символъ
; 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.4350 - Ени
                        кеев А.А.$'
                        A8 A2 A5 E2 E1 E2
                        A2 E3 A5 E2 20 E1
                        E2 2E A3 E0 2E 34
                        33 35 30 20 2D 20
                        85 AD A8 AA A5 A5
                        A2 20 80 2E 80 2E
                        24
003B                      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          9/19/24 16:12:26
                        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
                        останова 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,

```

```

; находящейся в 1-ом слове PSP.
001B      Main      ENDP
001B      CODE      ENDS
           END Main

#Microsoft (R) Macro Assembler Version 5.10      9/19/24 16:12:26
           Symbols-1

```

Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK	0018	PARA	STACK	
CODE	001B	PARA	NONE	
DATA	003B	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

Файл HELLO2.MAP:

Start Stop Length Name Class

00000H 00017H 00018H ASTACK

00020H 0005AH 0003BH DATA

00060H 0007AH 0001BH CODE

Program entry point at 0006:0005