

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО 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. 3384

Рудаков А.Л.

Преподаватель

Ковалев А.Д.

Санкт-Петербург

2024

Цель работы.

Изучить процесс трансляции, выполнения и отладки программ на языке Ассемблера процессоров Intel x86 для DOS.

Задание.

Часть 1.

1. Загрузить файл `hello1.asm` из каталога \Задания в каталог \MASM.
2. 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.
3. 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).
4. 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.
5. Выполнить программу в автоматическом режиме путем набора строки `> имя_файла.exe` и убедиться в ее работоспособности (результат выполнения просмотреть в режиме отображения экрана пользователя, получаемого набором клавиш `ctrl^O`);
6. 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.

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

Перемещения маркера между полями экрана управляется клавишами: F7(up), F8(down), F9(left), F10(right).

В пределах поля перемещение маркера управляется клавиатурой (->, <-, и др.)

Выход из отладчика - по команде Quit.

Записать содержимое всех регистров процессора, включая сегментные, перед выполнением 1-ой команды. Результаты прогона программы под управлением отладчика должны быть представлены в виде, показанном на примере одной команды в табл.1, и подписаны преподавателем.

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 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, убедиться в ее 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. Перед выполнением 1-ой команды записать содержимое сегментных регистров процессора.

Основные теоретические положения.

Регистры общего назначения (РОН) AX, BX, CX и DX используются для хранения данных и выполнения различных арифметических и логических операций. Кроме того, каждый из этих регистров поделён на 2 части по 8-бит, с которыми можно работать как с 8-битными регистрами (AH, AL, BH, BL, CH, CL, DH, DL). Младшие части регистров имеют в названии букву L (от слова Low), а старшие H (от слова High). Некоторые команды неявно используют определённый регистр, например, CX может выполнять роль счетчика цикла.

Индексные регистры предназначены для хранения индексов при работе с массивами. SI (Source Index) содержит индекс источника, а DI (Destination Index) — индекс приёмника, хотя их можно использовать и как регистры общего назначения.

Регистры-указатели BP и SP используются для работы со стеком. BP (Base Pointer) позволяет работать с переменными в стеке. Его также можно использовать в других целях. SP (Stack Pointer) указывает на вершину стека. Он используется командами, которые работают со стеком.

Сегментные регистры CS (Code Segment), DS (Data Segment), SS (Stack Segment) и ES (Enhanced Segment) предназначены для обеспечения сегментной адресации. Код находится в сегменте кода, данные — в сегменте данных, стек — в сегменте стека и есть еще дополнительный сегмент данных. 3 Реальный физический адрес получается путём сдвига содержимого сегментного регистра на 4 бита влево и прибавления к нему смещения

(относительного адреса внутри сегмента). Подробнее о сегментной адресации рассказывается в части 31.

COM-программа всегда находится в одном сегменте, который является одновременно сегментом кода, данных и стека. При запуске COM-программы сегментные регистры будут содержать одинаковые значения.

Выполнение работы.

Запуск DosBox, монтирование диска.

```
Z:\>SET BLASTER=A220 I7 D1 H5 T6  
  
Z:\>mount f D:\MASM  
Drive F is mounted as local directory D:\MASM
```

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

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

Переход к директории диска

```
Z:\>f:  
  
F:\>
```

Часть 1.

Программа hello1.asm:

1. Изменено приветствие:

```

DOSSEG                                ; Задание сегментов под DOS
.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.3384 - Рудаков А.Л.',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 DOS печати строки
int 21h                              ; вывод на экран приветствия
mov ah,4ch                           ; # функции DOS завершения программы
int 21h                              ; завершение программы и выход в DOS
END

```

2. Трансляция программы hello1.asm:

Команда для запуска трансляции: `masm hello1.asm`.

По команде порождаются объектный файл и файл листинга:

- `hello1.obj`

- `hello1.lst`.

Файл листинга приводится в приложении.

```

F:\>masm hello1.asm
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [hello1.OBJ]: hello1.obj
Source listing [NUL.LST]: hello1.lst
Cross-reference [NUL.CRF]: hello1.crf

47458 + 461849 Bytes symbol space free

0 Warning Errors
0 Severe Errors

F:\>

```

3. Компоновка загрузочного модуля:

Команда для запуска компоновки: `link hello1.obj`.

По команде порождаются исполняемый файл и карта памяти:

- `hello1.exe`

- `hello1.map`

Карта памяти приводится в приложении.

```
F:\>link hello1.obj

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

Run File [HELLO1.EXE]: hello1.exe
List File [NUL.MAP]: hello1.map
Libraries [.LIB]:
```

4. Выполнение программы в автоматическом режиме:

Команда для запуска программы: hello1.exe

```
F:\>hello1.exe
Вас приветствует ст.гр.3384 – Рудаков А.А.
```

5. Выполнение программы в отладчике:

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

```
AX 0000 SI 0000 CS 11AC IP 0010 Stack +0 1132
BX 0000 DI 0000 DS 119C          +2 1132
CX 004D BP 0000 ES 119C HS 119C    +4 1132
DX 0000 SP 0100 SS 11B1 FS 119C    +6 1132
```

Адрес команды	Код команды	16-ый код команды	Содержание регистров и ячеек памяти	
			До выполнения	После выполнения
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 = 0AA2 IP = 0015	DX = 0000 IP = 0018
0018	MOV AH, 09	B409	AX = 11AE	AX = 09AE

			IP = 0018	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:

```

┌ CMD >
└ Program terminated normally

```

Часть 2.

Программа hello2.asm:

1. Изменено приветствие:

```

DATA        SEGMENT

; Директивы описания данных

HELLO       DB 'Здравствуйте!', 0AH, 0DH, EOFLine
GREETING    DB 'Вас приветствует ст.гр.3384 - Рудаков А.Л.$'
DATA        ENDS

; Код программы

```

2. Трансляция программы hello2.asm:

Команда для запуска трансляции: `masm hello2.asm`.

По команде порождаются объектный файл и файл листинга:

- `hello2.obj`

- `hello2.lst`.

Файл листинга приводится в приложении.

```
F:\>masm hello2.asm
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [hello2.OBJ]: hello2.obj
Source listing [NUL.LST]: hello2.lst
Cross-reference [NUL.CRF]: hello2.crf

47466 + 461841 Bytes symbol space free

0 Warning Errors
0 Severe Errors
```

3. Компоновка загрузочного модуля:

Команда для запуска компоновки: `link hello2.obj`.

По команде порождаются исполняемый файл и карта памяти:

- `hello2.exe`

- `hello2.map`

Карта памяти приводится в приложении.

```
F:\>link hello2.obj

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

Run File [HELLO2.EXE]: hello2.exe
List File [NUL.MAP]: hello2.map
Libraries [.LIB]:
```

4. Выполнение программы:

Команда для запуска программы: `hello2.exe`

```
F:\>hello2.exe
Здравствуйте!
Вас приветствует ст.гр.3384 – Рудаков А.Л.
F:\>
```

5. Выполнение программы в отладчике:

Команда для запуска программы в режиме отладки: `afg hello2.exe`

```
AX 0000 SI 0000 CS 11B2 IP 0005 Stack +0 0000
BX 0000 DI 0000 DS 119C          +2 0000
CX 007B BP 0000 ES 119C HS 119C   +4 0000
DX 0000 SP 0018 SS 11AC FS 119C   +6 0000
```

Адрес команды	Код команды	16-ый код команды	Содержание регистров и ячеек памяти	
			До выполнения	После выполнения
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	DS = 11AE

			IP = 000C	IP = 000E
000E	MOV DX, 0000	BA0000	IP = 000E	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	SP = 0014

			IP = 0004 Stack: +0 001A +2 0000 +4 119C +6 0000	IP = 001A Stack: +0 0000 +2 119C +4 0000 +6 0000
001A	RET Far	CB	SP = 0014 CS = 11B2 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:



```

C:\CMD >
Program terminated normally

```

Программный код см. в приложении А.

Выводы.

Был изучен процесс трансляции, выполнения и отладки программ на языке Ассемблера процессоров Intel x86 для DOS. Полученные знания применены на практике.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: 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.3384 - Рудаков А.Л.',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
```

Название файла: hello1.lst

```
CMD > Microsoft (R) Macro Assembler
Version 5.10
9/14/24 15:41:05
Program terminated normally
Page 1-1
```

```
1
2 ; HELLO1.ASM - упрощенная версия учебной
3 программы лаб.раб. N1
4 ; по дисциплине "Архитектура компьютера"
5 ; *****
6 ; Назначение: Программа формирует и выводит на экран приветствие
7 ; пользователя с помощью функции ДОС "Вывод строки"
; (номер 09 прерывание 21h), которая:
```

```

8          ;          - обеспечивает вывод на
          экран строки символов,
9          ;          заканчивающейся знаком
          "$";
10         ;          - требует задания в реги-
          стре ah номера функции=09h,
11         ;          а в регистре dx - сме-
          щения адреса выводимой
12         ;          строки;
13         ;          - использует регистр ax
          и не сохраняет его
14         ;          содержимое.
15         ; *****
          *****
16
17         DOSSEG
          ; Задание сегментов под ДОС
18         .MODEL  SMALL
          ; Модель памяти-SMALL (Малая)
19         .STACK  100h
          ; Отвести под Стек 256 байт
20         .DATA
          ; Начало сегмента данных
21 0000      Greeting LABEL BYTE
          ; Текст приветствия
22 0000  82 A0 E1 20 AF E0      DB 'Вас приветствует ст.гр.3384 -
          Ру
          даков А.Л.',13,10,'$'
23      A8 A2 A5 E2 E1 E2
24      A2 E3 A5 E2 20 E1
25      E2 2E A3 E0 2E 33
26      33 38 34 20 2D 20
27      90 E3 A4 A0 AA AE
28      A2 20 80 2E 8B 2E
29      0D 0A 24
30         .CODE
          ; Начало сегмента кода
31 0000  B8 ---- R      mov  ax,@data
          ; Загрузка в DS адреса начала
32 0003  8E D8      mov  ds,ax
          ; сегмента данных
33 0005  BA 0000 R      mov  dx,OFFSET Greeting

```

```

C:\CMD > Microsoft (R) Macro Assembler
Version 5.10
9/14/24 15:41:05
Program terminated normally

```

Page 1-2

```

34         ; Загрузка в dx смещения
          ; адреса текста приветствия
35 0008      DisplayGreeting:
36 0008  B4 09      mov  ah,9
          ; # функции ДОС печати строки
37 000A  CD 21      int  21h
          ; вывод на экран приветствия
38 000C  B4 4C      mov  ah,4ch
          ; # функции ДОС завершения программы
39 000E  CD 21      int  21h
          ; завершение программы и выход в ДОС
40         END

```

```

Microsoft (R) Macro Assembler
Version 5.10
9/14/24 15:41:05
Program terminated normally

```

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

47458 + 461849 Bytes symbol space free

0 Warning Errors
0 Severe Errors

Название файла: 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			

Название файла: 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.3384 - Рудаков А.Л.$'
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

```

Название файла: hello2.lst



Microsoft (R) Macro Assembler Version
5.10 9/14/24 15:46:07
Program terminated normally

Page 1-1

```

1          ; HELLO2 -Учебная программа N2 лаб.раб.
          #1 по дисциплине "Архитектура компьютер
          а"
2          ;          Программа использует процеду
          ру для печати строки
3          ;
4          ;          ТЕКСТ ПРОГРАММЫ
5
6 = 0024          EOFLine EQU '$'          ; Определение
          символьной константы

```



```

7                                     ; "Конец
                                     строки"
8
9                                     ; Стек программы
10
11 0000                                AStack  SEGMENT  STACK
12 0000  000C[                        DW 12 DUP(?)    ; Отводится 1
                                     2 слов памяти
13      ????
14      ]
15
16 0018                                AStack  ENDS
17
18                                     ; Данные программы
19
20 0000                                DATA      SEGMENT
21
22                                     ; Директивы описания данных
23
24 0000  87 A4 E0 A0 A2 E1  HELLO      DB 'Здравствуйте!', 0AH, 0DH,
                                     EOFLine
25      E2 A2 E3 A9 E2 A5
26      21 0A 0D 24
27 0010  82 A0 E1 20 AF E0  GREETING DB 'Вас приветствует ст.гр.33
                                     84 - Рудаков А.Л.$'
28      A8 A2 A5 E2 E1 E2
29      A2 E3 A5 E2 20 E1
30      E2 2E A3 E0 2E 33
31      33 38 34 20 2D 20
32      90 E3 A4 A0 AA AE
33      A2 20 80 2E 8B 2E
34      24
35 003B                                DATA      ENDS
36
37                                     ; Код программы
38
39 0000                                CODE      SEGMENT
40                                     ASSUME CS:CODE, DS:DATA, SS:A
                                     Stack
41
42                                     ; Процедура печати строки
43 0000                                WriteMsg  PROC  NEAR
44 0000  B4 09                                mov  AH,9
45 0002  CD 21                                int   21h ; Вызов функции DO

```

```

C:\CMD > Microsoft (R) Macro Assembler
Version 5.10
9/14/24 15:46:07
Program terminated normally

```

Page 1-2

```

S по прерыванию
46 0004  C3                                ret
47 0005                                WriteMsg  ENDP
48
49                                     ; Головная процедура
50 0005                                Main      PROC  FAR
51 0005  1E                                push  DS ;\ Сохранение
                                     адреса начала PSP в стеке
52 0006  2B C0                            sub   AX,AX ; > для послед
                                     ущего восстановления по
53 0008  50                                push  AX ;/ команде re

```

```

t, завершающей процедуру.
54 0009 B8 ---- R      mov     AX,DATA          ; 3
агрузка сегментного
55 000C 8E D8          mov     DS,AX            ; p
егистра данных.
56 000E BA 0000 R      mov     DX, OFFSET HELLO   ; B
ывод на экран первой
57 0011 E8 0000 R      call    WriteMsg          ; c
троки приветствия.
58 0014 BA 0010 R      mov     DX, OFFSET GREETING ; B
ывод на экран второй
59 0017 E8 0000 R      call    WriteMsg          ; c
троки приветствия.
60 001A CB              ret                      ; B
ыход в DOS по команде,
61                                ; H
аходящейся в 1-ом слове PSP.
62 001B                Main      ENDP
63 001B                CODE      ENDS
64                                END Main

```

```

Microsoft (R) Macro Assembler
Version 5.10
9/14/24 15:46:07
Program terminated normally

```

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

```

47466 + 461841 Bytes symbol space free

```

0 Warning Errors
0 Severe Errors

```

Название файла: 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