

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: Трансляции, отладка и выполнение программ на языке
Ассемблера.

Студент гр. 0383

Куликов А. В.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2021

Цель работы.

Изучить трансляцию, отладку и выполнение программ на языке Ассемблера.

Задание.

Часть 1

1. Просмотреть программу `hello1.asm`, которая формирует и выводит на экран приветствие пользователя с помощью функции ОС MSDOS, вызываемой через прерывание с номером 21H (команда `Int 21h`). Выполняемые функцией действия и задаваемые ей параметры - следующие:

- обеспечивается вывод на экран строки символов, заканчивающейся знаком "\$";
- требуется задание в регистре `ah` номера функции, равного `09h`, а в регистре `dx` - смещения адреса выводимой строки;
- используется регистр `ax` и не сохраняется его содержимое.

2. Разобраться в структуре и реализации каждого сегмента программы. Непонятные фрагменты прояснить у преподавателя. Строку-приветствие преобразовать в соответствии со своими личными данными.

3. Загрузить файл `hello1.asm` из каталога Задания в каталог Masm.

4. Протранслировать программу с помощью строки

```
> masm hello1.asm
```

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.

5. Скомпоновать загрузочный модуль с помощью строки

```
> link hello1.obj
```

с созданием карты памяти и исполняемого файла `hello1.exe`.

6. Выполнить программу в автоматическом режиме путем набора строки

> hello1.exe

убедиться в корректности ее работы и зафиксировать результат выполнения в протоколе.

7. Запустить выполнение программы под управлением отладчика с помощью команды

> afd hello1.exe

4. Записать начальное содержимое сегментных регистров CS, DS, ES и SS. Выполнить программу в пошаговом режиме с фиксацией используемых регистров и ячеек памяти до и после выполнения каждой команды. Обычные команды выполняются по F1 (Step), а вызовы обработчиков прерываний (Int) - по F2 (StepProc), чтобы не входить внутрь обработчика прерываний. Продвижение по сегментам экранной формы отладчика выполняется с помощью клавиш F7 – F10 (up, down, left, right). Перезапуск программы в отладчике выполняется клавишей F3 (Retrieve). Выход из отладчика - по команде Quit.

Результаты прогона программы под управлением отладчика должны быть представлены в виде, показанном на примере одной команды в табл.1, и подписаны преподавателем.

Табл.1

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения.	После выполнения.
0003	MOV DS, AX	8E D8	(AX) = 2D87 (DS) = 2D75 (IP) = 0003	(AX) = 2D87 (DS) = 2D87 (IP) = 0005

Часть 2

Выполнить пункты 1 - 7 части 1 настоящего задания применительно к программе hello2.asm, приведенной в каталоге Задания, которая выводит на экран приветствие пользователя с помощью процедуры WriteMsg, а также использует полное определение сегментов. Сравнить результаты прогона под

управлением отладчика программ hello1 и hello2 и объяснить различия в размещении сегментов.

Выполнение работы.

1. Просмотрена программа hello1.asm, изменено приветственное сообщение.
2. Программа была протранслирована с помощью строки `masm hello1.asm`, создан объектный файл и файл листинга, hello1.obj и hello1.asm соответственно.

```
C:\>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]:

    47992 + 461315 Bytes symbol space free

    0 Warning Errors
    0 Severe Errors
```

(Рис. 1) – Результат трансляции программы

Из рис. 1 видно, что программа была скомпилирована успешно и ьбез ошибок.

3. Был скомпонован загрузочный модуль и файл карты памяти hello1.map с помощью строки `link hello1.obj`

```
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]:
```

(Рис. 2) – Результат компоновки загрузочного модуля

4. Программа была выполнена путем набора строки `hello1.exe`

```
C:\>hello1.exe
Вас приветствует ст.гр. 0383 – Куликов А. В.
```

(Рис. 3) – Результат выполнения программы

Программа была выполнена корректно и вывела измененное приветствие.

5. Программа была запущена в отладчике afd командой afd hello1.exe

Начальное содержимое сегментов

(CS) = 1A05

(DS) = 19F5

(ES) = 19F5

(SS) = 1A0A

Табл. 1

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения.	После выполнения.
0010	MOV AX, 1A07	B0871A	(AX) = 0000 (DS) = 0000 (IP) = 0010	(AX) = 1A07 (DS) = 0000 (IP) = 0013
0013	MOV DS, AX	8ED8	(AX) = 1A07 (DS) = 0000 (IP) = 0013	(AX) = 1A07 (DS) = 1A07 (IP) = 0015
0015	MOV DX, 0000	BA0000	(AX) = 1A07 (DS) = 1A07 (IP) = 0015 (DX) = 0000	(AX) = 1A07 (DS) = 1A07 (IP) = 0018 (DX) = 0000
0018	MOV AH, 09	A409	(AX) = 1A07 (DS) = 1A07 (IP) = 0018	(AX) = 0907 (DS) = 1A07 (IP) = 001A
001A	INT 21	CD21	(AX) = 0907 (DS) = 1A07 (IP) = 001A	(AX) = 0907 (DS) = 1A07 (IP) = 001C
001C	MOV AH, 4C	B44C	(AX) = 0907 (DS) = 1A07 (IP) = 001C	(AX) = 4C07 (DS) = 1A07 (IP) = 001E
001E	INT 21	CD21	(AX) = 4C07 (DS) = 1A07 (IP) = 001E	Завершение работы

6. Аналогично для hello2.asm создаются объектный файл, файл листинг и исполняемый файл, hello2.obj, hello2.lst, hello2.exe соответственно.

7. Программа была выполнена вводом строки hello2.exe

```
C:\>hello2.exe
Hello Worlds!
Student from 0383 - Kulikov A. U.
```

(Рис. 4) – результат работы программы

8. hello2.exe была запущена в отладчике afd командой afd hello2.exe

Начальное содержимое сегментов

(CS) = 1A0B

(DS) = 19F5

(ES) = 18F5

(SS) = 1A05

Табл. 2

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения.	После выполнения.
0005	PUSH DS	1E	(AX) = 0000 (DS) = 0000 (IP) = 0005	(AX) = 0000 (DS) = 0000 (IP) = 0006
0006	SUB AX, AX	2BC0	(AX) = 0000 (DS) = 0000 (IP) = 0006	(AX) = 0000 (DS) = 0000 (IP) = 0008
0008	PUSH AX	50	(AX) = 0000 (DS) = 0000 (IP) = 0008	(AX) = 0000 (DS) = 0000 (IP) = 0008
0009	MOV AX, 1A07	B8071A	(AX) = 0000 (DS) = 0000 (IP) = 0008	(AX) = 1A07 (DS) = 0000 (IP) = 000C
000C	MOV DS, AX	8ED8	(AX) = 1A07 (DS) = 0000 (IP) = 000C	(AX) = 1A07 (DS) = 1A07 (IP) = 000E
000E	MOV DX, 0000	BA0000	(AX) = 1A07 (DS) = 1A07	(AX) = 1A07 (DS) = 1A07

			(IP) = 000E (DX) = 0000	(IP) = 0011 (DX) = 0000
0011	CALL 0000	E8CEFF	(AX) = 1A07 (DS) = 1A07 (IP) = 0011	(AX) = 1A07 (DS) = 1A07 (IP) = 0000
0000	MOV AH, 09	B409	(AX) = 1A07 (DS) = 1A07 (IP) = 0000	(AX) = 0907 (DS) = 1A07 (IP) = 0002
0002	INT 21	CD21	(AX) = 0907 (DS) = 1A07 (IP) = 0002	(AX) = 0907 (DS) = 1A07 (IP) = 0004
0004	RET	C3	(AX) = 0907 (DS) = 1A07 (IP) = 0004	(AX) = 0907 (DS) = 1A07 (IP) = 0014
00014	MOV DX, 0010	BA1000	(AX) = 0907 (DS) = 1A07 (IP) = 0014 (DX) = 0000	(AX) = 0907 (DS) = 1A07 (IP) = 0017 (DX) = 0010
0017	CALL 0000	E8E6FF	(AX) = 0907 (DS) = 1A07 (IP) = 0017	(AX) = 0907 (DS) = 1A07 (IP) = 0000
0000	MOV AH, 09	B409	(AX) = 0907 (DS) = 1A07 (IP) = 0000	(AX) = 0907 (DS) = 1A07 (IP) = 0002
0002	INT 21	CD21	(AX) = 0907 (DS) = 1A07 (IP) = 0002	(AX) = 0907 (DS) = 1A07 (IP) = 0004
0004	RET	C3	(AX) = 0907 (DS) = 1A07 (IP) = 0004	(AX) = 0907 (DS) = 1A07 (IP) = 001A
001A	RET FAR	CB	(AX) = 0907 (DS) = 1A07 (IP) = 001A	(AX) = 0907 (DS) = 1A07 (IP) = 0000

0000	INT 20	CD20	(AX) = 0907 (DS) = 1A07 (IP) = 0000	Завершение работы
------	--------	------	---	-------------------

9. Анализ отличий в работе программ hello1.asm и hello2.asm.

В hello2.asm напрямую задаются смещения до сегментов кода и стека строкой `ASSUME CS:CODE, SS:STACK`, после чего выделяется 12 двухбайтовых слов для стека, когда же в hello1.asm стек длиной в 100 байт и сегмент данных инициализируются при помощи директивы `DOSSEG`. В обеих программах вывод текста, заданного в сегменте данных, происходит посредством вызова прерывания `INT 21H` с кодом 09, который заранее помещается в старший байт регистра `AX`, но в hello2.asm вызов нужного прерывания вынесен в процедуру `WriteMsg` и перед выводом строки происходит вызов процедуры строкой `call WriteMsg`. Также и в первой, и во второй программе в регистр `DX` записывается смещение до необходимых данных. После вывода строк обе программы завершаются, но различными способами, hello1.asm вызывает прерывание `int 21h` с кодом 4ch, что соответствует выходу из программы. В hello2.asm в начале программы на стеке сохраняются четыре байта, изначальное значение `DS`, которое указывает на начало `PSP` и два нулевых байта, которые получаются строкой `SUB AX, AX`, эти четыре байта на стеке при вызове дальнего `RET` переводят `IP` на первый байт `PSP`, где находится прерывание `INT 20`, которое завершает выполнение программы.

Выводы.

В ходе лабораторной работы была изучена трансляция, отладки и работа программ на языке Ассемблера. Были разобраны разные подходы к реализации поставленной задачи и различные способы корректного завершения программы.

ПРИЛОЖЕНИЕ А

Тексты исходных файлов программ

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. 0383 - Куликов А. В.',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
```

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

ASSUME CS:CODE, SS:STACK

STACK  SEGMENT STACK
        DW 12 DUP(?) ; ОТВОДИТСЯ 12 СЛОВ ПАМЯТИ
STACK  ENDS

; ДАННЫЕ ПРОГРАММЫ

DATA   SEGMENT

; ДИРЕКТИВЫ ОПИСАНИЯ ДАННЫХ

HELLO  DB 'HELLO WORLDS!', 0AH, 0DH,EOFLINE
GREETING DB 'STUDENT FROM 0383 - KULIKOV A. V.$'
DATA   ENDS

; КОД ПРОГРАММЫ

CODE   SEGMENT
; 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
```

ПРИЛОЖЕНИЕ Б

Тексты файлов диагностических сообщений программ

hello1.lst

MICROSOFT (R) MACRO ASSEMBLER VERSION 5.10

9/14/21 19:35:27

PAGE 1-1

```
; HELLO1.ASM - УПРОЩЕННАЯ ВЕРСИЯ УЧЕБНОЙ ПРОГР
АММЫ ЛАБ.РАБ. N1
;      ПО ДИСЦИПЛИНЕ "АРХИТЕКТУРА КОМП
БЮТЕРА"
; *****
*****
; НАЗНАЧЕНИЕ: ПРОГРАММА ФОРМИРУЕТ И ВЫВОДИТ НА
ЭКРАН ПРИВЕТСТВИЕ
;      ПОЛЬЗОВАТЕЛЯ С ПОМОЩЬЮ ФУНКЦИИ ДО
С "ВЫВОД СТРОКИ"
;      (НОМЕР 09 ПРЕРЫВАНИЕ 21Н), КОТОРА
Я:
;      - ОБЕСПЕЧИВАЕТ ВЫВОД НА ЭКРАН СТ
РОКИ СИМВОЛОВ,
;      ЗАКАНЧИВАЮЩЕЙСЯ ЗНАКОМ "$";
;      - ТРЕБУЕТ ЗАДАНИЯ В РЕГИСТРЕ АН
НОМЕРА ФУНКЦИИ=09Н,
;      А В РЕГИСТРЕ DX - СМЕЩЕНИЯ А
ДРЕСА ВЫВОДИМОЙ
;      СТРОКИ;
;      - ИСПОЛЬЗУЕТ РЕГИСТР АХ 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. 0383 - КУЛИКОВ
А. В.',13,10,'$'

A8 A2 A5 E2 E1 E2
A2 E3 A5 E2 20 E1
E2 2E A3 E0 2E 20
30 33 38 33 20 2D
20 20 8A E3 AB A8
AA AE A2 20 80 2E
20 82 2E 0D 0A 24
```

```

        .CODE                      ; НАЧАЛ
О СЕГМЕНТА КОДА
0000 B8 ---- R      MOV AX, @DATA          ; ЗАГР
УЗКА В DS АДРЕСА НАЧАЛА
0003 8E D8          MOV DS, AX             ; СЕГМ
ЕНТА ДАННЫХ
0005 BA 0000 R      MOV DX, OFFSET GREETING ; ЗАГР
УЗКА В DX СМЕЩЕНИЯ
MICROSOFT (R) MACRO ASSEMBLER VERSION 5.10      9/14/21 19:35:27
        PAGE    1-2

```

```

; АДРЕС
А ТЕКСТА ПРИВЕТСТВИЯ
0008      DISPLAYGREETING:
0008 B4 09      MOV AH, 9                  ; # ФУ
НКЦИИ ДОС ПЕЧАТИ СТРОКИ
000A CD 21      INT 21H                    ; ВЫВОД
НА ЭКРАН ПРИВЕТСТВИЯ
000C B4 4C      MOV AH, 4CH                ; # ФУ
НКЦИИ ДОС ЗАВЕРШЕНИЯ ПРОГРАММЫ
000E CD 21      INT 21H                    ; ЗАВЕР
ШЕНИЕ ПРОГРАММЫ И ВЫХОД В ДОС
END
MICROSOFT (R) MACRO ASSEMBLER VERSION 5.10      9/14/21 19:35:27
        SYMBOLS-1

```

SEGMENTS AND GROUPS:

N A M E	LENGTH	ALIGN	COMBINE CLASS
DGROUP	GROUP		
_DATA	0030	WORDPUBLIC	'DATA'
STACK	0100	PARA STACK	'STACK'
_TEXT	0010	WORDPUBLIC	'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

```

47992 + 461315 BYTES SYMBOL SPACE FREE

0 WARNING ERRORS
0 SEVERE ERRORS

hello2.lst

MICROSOFT (R) MACRO ASSEMBLER VERSION 5.10

9/2/21 11:22:44

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

ASSUME CS:CODE, SS:STACK

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 48 65 6C 6C 6F 20 HELLO DB 'HELLO WORLDS!', 0AH, 0DH,EOFLINE
      57 6F 72 6C 64 73
      21 0A 0D 24
0010 53 74 75 64 65 6E GREETING DB 'STUDENT FROM 4350 - $'
      74 20 66 72 6F 6D
      20 34 33 35 30 20
      2D 20 24
0025          DATA ENDS

; КОД ПРОГРАММЫ

0000          CODE SEGMENT
; ПРОЦЕДУРА ПЕЧАТИ СТРОКИ
0000          WRITMSG PROC NEAR
0000 B4 09          MOV AH,9
0002 CD 21          INT 21H ; ВЫЗОВ ФУНКЦИИ DOS ПО ПРЕ
```

```

0004 C3          РЫВАНИЮ
0005          RET
0005          WRITEMSG ENDP

; 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
MICROSOFT (R) MACRO ASSEMBLER VERSION 5.10          9/2/21 11:22:44

```

PAGE 1-2

```

0008 50          ОСТАНОВЛЕНИЯ ПО
                     PUSH AX    ;/ КОМАНДЕ RET, ЗАВЕР
0009 B8 ---- R    ШАЮЩЕЙ ПРОЦЕДУРУ.
                     MOV  AX,DATA      ; ЗАГРУЗКА
000C 8E D8        СЕГМЕНТНОГО
                     MOV  DS,AX        ; РЕГИСТРА
000E BA 0000 R    ДАННЫХ.
                     MOV  DX, OFFSET HELLO ; ВЫВОД НА
0011 E8 0000 R    ЭКРАН ПЕРВОЙ
                     CALL WRITEMSG      ; СТРОКИ ПР
0014 BA 0010 R    ИВЕТСТВИЯ.
                     MOV  DX, OFFSET GREETING ; ВЫВОД НА
0017 E8 0000 R    ЭКРАН ВТОРОЙ
                     CALL WRITEMSG      ; СТРОКИ ПР
001A CB          ИВЕТСТВИЯ.
                     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          9/2/21 11:22:44

```

SYMBOLS-1

SEGMENTS AND GROUPS:

N A M E	LENGTH	ALIGN	COMBINE CLASS
ASTACK	0018	PARA	STACK
CODE	001B	PARA	NONE
DATA	0025	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

47978 + 461297 BYTES SYMBOL SPACE FREE

0 WARNING ERRORS
0 SEVERE ERRORS