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 _____ Козлов Т.В.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Основные теоретические положения.

Посмотреть программы hello1.asm и hello2.asm. Разобраться в структуре и реализации каждого сегмента программ, строку-приветствие преобразовать в соответствии со своими личными данными.

Для обоих файлов: протранслировать программу с созданием объектного файла и файла диагностических сообщений (файла листинга), получить объектный модуль, скомпоновать загрузочный модуль с созданием карты памяти и исполняемым файлом. Выполнить программу с фиксацией результата в протоколе.

Запустить программы под управлением отладчика с фиксацией используемых регистров и ячеек памяти до и после выполнения каждой команды. Сравнить результаты прогона под управлением отладчика программ hello1 и hello2 и объяснить различия в размещении сегментов.

Ход работы.

1. Каталог с MASM смонтирован в эмулятор командой mount с C:\Путь (используя встроенную возможность для Windows перетаскивания нужного каталога на иконку приложения DOSBox)

2. Команды для получения исполняемого файла программы предоставлены на рис. 1.

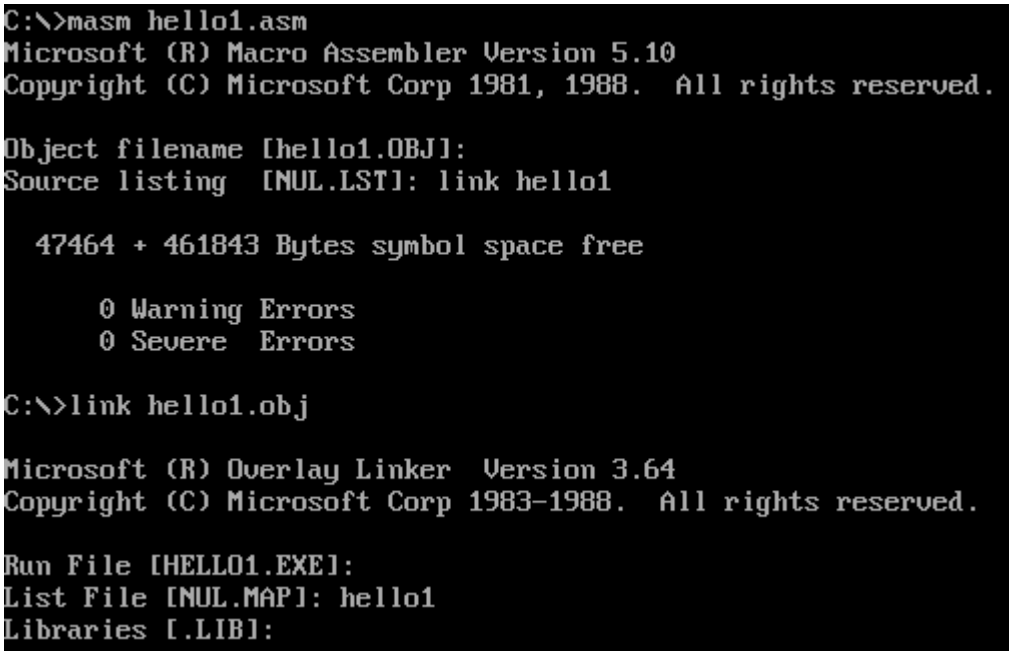

Рис. 1 – Получение исполняемого файла

3. Результат работы исполняемого файла предоставлен на рис.2

Рис. 2 – Результат работы hello1.exe

4. Запуск программы в отладчике командой:

afdpro hello1.exe

5. Начальное содержимое сегментных регистров для hello.exe:

(CS) = 1A05

(DS) = 19F5

(ES) = 19F5

(SS) = 1A0A

6. Результаты прогона hello1.exe под управлением отладчика предоставлены в табл.1

Табл.1:

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0010	mov AX, 1A07	B8071A	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (IP) = 0010	(AX) = 1A07 (DX) = 0000 (DS) = 19F5 (IP) = 0013
0013	mov DS, AX	8ED8	(AX) = 1A07 (DX) = 0000 (DS) = 19F5 (IP) = 0013	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (IP) = 0015
0015	mov DX, 0000	BA0000	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (IP) = 0015	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (IP) = 0018
0018	mov AH, 09	B409	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (IP) = 0018	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (IP) = 001A
001A	int 21	CD21	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (IP) =001A	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (IP) = 001C
001C	mov AH, 4C	B44C	(AX) = 0907	(AX) = 4C07

			(DX) = 0000 (DS) = 1A07 (IP) = 001C	(DX) = 0000 (DS) = 1A07 (IP) = 001E
001E	int 21	CD21	(AX) = 4C07 (DX) = 0000 (DS) = 1A07 (IP) = 001E	Программа корректно завершилась

Компоненты программы см. в приложении А.

7. Аналогично создается hello2.exe. Результат работы исполняемого файла предоставлен на рис. 3:

Рис.3 – Результат работы hello2.exe

8. Начальное содержимое сегментных регистров для hello2.exe:

(CS) = 1A0A

(DS) = 19F5

(ES) = 19F5

(SS) = 1A05

9. Результаты прогона hello2.exe под управлением отладчика предоставлены в табл.2

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 (DX) = 0000 (DS) = 19F5 (CS) = 1A0A (SP) = 0018	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (CS) = 1A0A (SP) = 0016

			(IP) = 0005 Stack +0 0000	(IP) = 0006 Stack +0 19F5
0006	sub AX, AX	2BC0	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (CS) = 1A0A (SP) = 0016 (IP) = 0006 Stack +0 19F5	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (CS) = 1A0A (SP) = 0016 (IP) = 0008 Stack +0 19F5
0008	push AX	50	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (CS) = 1A0A (SP) = 0016 (IP) = 0008 Stack +0 19F5 Stack +2 0000	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (CS) = 1A0A (SP) = 0014 (IP) = 0009 Stack +0 0000 Stack +2 19F5
0009	mov AX, 1A07	B8071 A	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (CS) = 1A0A (SP) = 0016 (IP) = 0008 Stack +0 0000 Stack +2 19F5	(AX) = 1A07 (DX) = 0000 (DS) = 19F5 (CS) = 1A0A (SP) = 0014 (IP) = 000C Stack +0 0000 Stack +2 19F5
000C	mov DS, AX	8ED8	(AX) = 1A07 (DX) = 0000 (DS) = 19F5 (CS) = 1A0A (SP) = 0014 (IP) = 000C	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0014 (IP) = 000E

			Stack +0 0000 Stack +2 19F5	Stack +0 0000 Stack +2 19F5
000E	mov DX, 0000	BA000 0	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0014 (IP) = 000E Stack +0 0000 Stack +2 19F5	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0014 (IP) = 0011 Stack +0 0000 Stack +2 19F5
0011	call 0000	E8ECF F	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0014 (IP) = 0011 Stack +0 0000 Stack +2 19F5 Stack +4 0000	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 (IP) = 0000 Stack +0 0014 Stack +2 0000 Stack +4 19F5
0000	mov AH, 09	B409	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 (IP) = 0000 Stack +0 0014 Stack +2 0000 Stack +4 19F5	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 (IP) = 0002 Stack +0 0014 Stack +2 0000 Stack +4 19F5
0002	int 21	CD21	(AX) = 0907 (DX) = 0000 (DS) = 1A07	(AX) = 0907 (DX) = 0000 (DS) = 1A07

			(CS) = 1A0A (SP) = 0012 (IP) = 0002 Stack +0 0014 Stack +2 0000 Stack +4 19F5	(CS) = 1A0A (SP) = 0012 (IP) = 0004 Stack +0 0014 Stack +2 0000 Stack +4 19F5
0004	ret	C3	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 (IP) = 0004 Stack +0 0014 Stack +2 0000 Stack +4 19F5	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0014 (IP) = 0004 Stack +0 0000 Stack +2 19F5 Stack +4 0000
0014	mov DX, 0010	BA100 0	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0014 (IP) = 0004 Stack +0 0000 Stack +2 19F5	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 1A0A (SP) = 0014 (IP) = 0017 Stack +0 0000 Stack +2 19F5
0017	call 0000	E8E6F F	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 1A0A (SP) = 0014 (IP) = 0017 Stack +0 0000 Stack +2 19F5	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 (IP) = 0000 Stack +0 001A Stack +2 0000

			Stack +4 0000	Stack +4 19F5
0000	mov AH, 09	B409	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 (IP) = 0000 Stack +0 001A Stack +2 0000 Stack +4 19F5	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 (IP) = 0002 Stack +0 001A Stack +2 0000 Stack +4 19F5
0002	int 21	CD21	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 (IP) = 0002 Stack +0 001A Stack +2 0000 Stack +4 19F5	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 (IP) = 0004 Stack +0 001A Stack +2 0000 Stack +4 19F5
0004	ret	C3	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 (IP) = 0004 Stack +0 001A Stack +2 0000 Stack +4 19F5	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 1A0A (SP) = 0014 (IP) = 001A Stack +0 0000 Stack +2 19F5 Stack +4 0000
001A	ret Far	CB	(AX) = 0907 (DX) = 0010 (DS) = 1A07	(AX) = 0907 (DX) = 0010 (DS) = 1A07

			(CS) = 1A0A (SP) = 0014 (IP) = 001A Stack +0 0000 Stack +2 19F5	(CS) = 19F5 (SP) = 0018 (IP) = 0000 Stack +0 0000 Stack +2 0000
0000	int 20	CD20	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 19F5 (SP) = 0018 (IP) = 0000 Stack +0 0000 Stack +2 0000	Программа корректно завершилась

Компоненты программы см. в приложении Б.

10. Анализ работы программ:

hello1:

1) В сегменте данных .DATA определяется метка (LABEL) Greeting типа BYTE, для которой резервируется и сразу инициализируется текст приветствия.
2) Далее в DS (DS – регистр, который должен указывать на начало данных в программе) загружаем адрес начала сегмента данных, но мы не можем сделать это напрямую (а только используя другие регистры), поэтому сначала записываем адрес в AX (mov AX, @data; @data = 1A07, где @data – идентификатор сегмента данных, на место которого после сборки устанавливается реальное смещение данного сегмента), а потом копируем значение AX в DS.
3) Далее в DX записываем смещение адреса выводимой строки приветствия (mov DX, OFFSET Greeting; OFFSET возвращает смещение в соответствующем сегменте выражения)

4) mov AH, 9 – помещаем в старший бит AX номер функции DOS печати строки (09h = 9 в dec, чем объясняется отсутствие h), далее вызывается прерывание, которое исполняет функцию по номеру, переданному в AH. Функция выведет строку по смещению адреса, указанному в DX (п.3).
5) mov AH, 4ch – аналогично помещаем номер функции завершения, после чего вызываем прерывание int 21 и программа корректно завершается.

hello2 (в основном описываются различия от реализации в hello1.asm):

1) В сегменте данных резервируются и инициализируются переменные HELLO и GREETING приветственными строками.
2) В сегменте кода отдельно описаны две процедуры: функция WriteMsg для печати строки и головная процедура – Main, в которой происходит вызов процедуры WriteMsg.
3) Работа процедуры Main начинается с сохранения адреса начала префикса программного сегмента в стеке командой push DS, затем значение AX сбрасывается в 0 командой sub AX, AX. AX сохраняется в стек, чтобы потом процедура WriteMsg командой ret осуществила выход из процедуры (ret извлечет из стека адрес возврата и передаст управление назад в программу, которая вызвала процедуру).
4) Затем происходит загрузка сегментного регистра данных, в DX аналогично hello1 записывается смещение адреса выводимой строки.
5) Командой call вызывается процедура WriteMsg, происходит вывод текста, после чего управление передается в Main
6) Аналогично печатается вторая строка.
7) Программа корректно завершается.

Выводы.

В ходе выполнения работы были изучены некоторые основы ассемблера, созданы исполняемые файлы для двух программ на языке ассемблер, имеющие различные подходы к реализации задания. Подходы были изучены и проанализированы.

Ознакомились с эмулятором DOSBox.

ПРИЛОЖЕНИЕ А

Тексты компонентов программы hello1.exe

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 - Козлов Т.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
```

hello1.lst

Microsoft (R) Macro Assembler Version 5.10 9/13/21 20:23:20

Page 1-1

```
1
2                               ; HELLO1.ASM -  упрощенная версия учебн
```

```
                ой программы лаб.раб. N1
3               ;           по дисциплине "Архитект
                ура компьютера"
4               ; ********************************
                ************************
5               ; Назначение: Программа формирует и выв
                одит на экран приветствие
6               ;           пользователя с помощью фу
                нкции ДОС "Вывод строки"
7               ;           (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.0383 -
Козлов Т.В.',13,10,'$'
23       A8 A2 A5 E2 E1 E2
24       A2 E3 A5 E2 20 E1
25       E2 2E A3 E0 2E 37
26       33 30 33 20 2D 20
27       88 A2 A0 AD AE A2
28       20 88 2E 88 2E 0D
29       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

Microsoft (R) Macro Assembler Version 5.10          9/13/21 20:23:20
                                Page     1-2


                        ; Загрузка в dx смещения
34
                        ; 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          9/13/21 20:23:20
                                                    Symbols-1


Segments and Groups:

                N a m e         Length   AlignCombine Class

DGROUP . . . . . .              GROUP
 _DATA  . . . . . .             002C  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

 47464 + 461843 Bytes symbol space free

     0 Warning Errors
     0 Severe  Errors
```

ПРИЛОЖЕНИЕ Б

Тексты компонентов программы hello1.exe

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:AStack

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

; Данные программы

DATA      SEGMENT

;  Директивы описания данных
```

```
HELLO     DB 'Hello Worlds!', 0AH, 0DH,EOFLine
GREETING  DB 'Student from 0383 - Kozlov T.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
```

hello2.lst

Microsoft (R) Macro Assembler Version 5.10 9/13/21 20:30:40

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:AStack

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 6EGREETING  DB 'Student from 0383 - Kozlov T.V.$'
      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                    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
0005                    Main    PROC  FAR
0005  1E                        push  DS      ;\  Сохранение адреса
                        начала PSP в стеке
0006  2B C0                      sub   AX,AX   ; > для последующего

Microsoft (R) Macro Assembler Version 5.10              9/13/21 20:30:40

                                                        Page     1-2


                        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          9/13/21 20:30:40

                                                    Symbols-1


Segments and Groups:

                N a m e         Length   AlignCombine 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
```