

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: Трансляция, отладка и выполнение программ
на языке Ассемблера

Студент гр. 0383

Самара. Р.Д.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Изучение трансляции, отладки и выполнения программ на языке Ассемблера.

Задание.

Лабораторная работа 1 использует 2 готовых программы на ассемблере: hello1 – составлена с использованием сокращенного описания сегментов и hello2 – составлена с полным описанием сегментов и выводом строки, оформленным как процедура. Выполнение работы состоит из двух частей, по каждой из которых необходимо представить протокол с фиксацией всех выполняемых действий и полученных результатов, и подписать его у преподавателя. Уточнение задания следует посмотреть в файле lr1_comp.txt каталога

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

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.

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.

Часть 1

1. Просмотрена программа `hello1.asm`.

2. Структура и реализация каждого сегмента программы разобраны.

Непонятные фрагменты прояснены у преподавателя. Строка-приветствие преобразована в соответствии со своими личными данными.

3. Файл `hello1.asm` из каталога Задания в каталог Masm загружен.

4. Программа протранслирована с помощью строки

```
> masm hello1.asm
```

с созданием объектного файла и файла диагностических сообщений (файла листинга). Синтаксические ошибки не обнаружены транслятором.

5. Загрузочный модуль скомпонован с помощью строки

```
> link hello1.obj
```

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

6. Программа выполнена в автоматическом режиме путем набора строки

```
> hello1.exe
```

Программа работает корректно. Результат: «Greetings from Samara Roman, 0383»

7. Запущено выполнение программы под управлением отладчика с помощью команды

```
> afd hello1.exe (afdpro hello1.exe)
```

Начальное содержимое сегментных регистров:

(CS) = 1A05, (DS) = 19F5, (ES) = 19F5 и (SS) = 1A0B.

Выполнена программа в пошаговом режиме с фиксацией используемых регистров и ячеек памяти до и после выполнения каждой команды в таблицу 1.

Таблица 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 (DS) = 1A07 (IP) = 001A (DX) = 0000	(AX) = 0907 (DS) = 1A07 (IP) = 001C (DX) = 0000
001C	Mov AH, 4C	B44C	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (IP) = 14A0	(AX) = 4C07 (DX) = 0000 (DS) = 1A07 (IP) = 001E
001E	Int 21	CD21	(AX) = 4C07 (DX) = 0000 (IP) = 001E (DS) = 1A07	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (IP) = 0010

Часть 2. Работа с hello2.asm

Выполнено протранслирование программы, компоновка загрузочного модуля и запуск программы в автоматическом режиме. Программа работает корректно.

Результат выполнения: «Hello worlds!\nStudent from 0383 – Samara R. D.»

Произведен запуск программы под управлением отладчика с фиксацией изменений используемых регистров до и после выполнения команд в таблицу 2.

Начальное содержимое сегментных регистров:

(CS) = 1A0A, (DS) = 19F5, (ES) = 19F5 и (SS) = 1A05

Таблица 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 Stack +0 0000 (IP) = 0005	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (CS) = 1A0A (SP) = 0016 Stack +0 19F5 (IP) = 0006
0006	SUB AX, AX	2BC0	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (CS) = 1A0A (SP) = 0016 Stack +0 19F5 (IP) = 0006	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (CS) = 1A0A (SP) = 0016 Stack +0 19F5 (IP) = 0008
0008	PUSH AX	50	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (CS) = 1A0A (SP) = 0016 Stack +0 19F5	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (CS) = 1A0A (SP) = 0014 Stack +0 0000

			(IP) = 0008	Stack +2 19F5 (IP) = 0009
0009	MOV AX, 1A07	B8071A	(AX) = 0000 (DX) = 0000 (DS) = 19F5 (CS) = 1A0A (SP) = 0014 Stack +0 0000 Stack +2 19F5 (IP) = 0009	(AX) = 1A07 (DX) = 0000 (DS) = 19F5 (CS) = 1A0A (SP) = 0014 Stack +0 0000 Stack +2 19F5 (IP) = 000C
000C	MOV DX, AX	8ED8	(AX) = 1A07 (DX) = 0000 (DS) = 19F5 (CS) = 1A0A (SP) = 0014 Stack +0 0000 Stack +2 19F5 (IP) = 000C	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0014 Stack +0 0000 Stack +2 19F5 (IP) = 000E
000E	MOV DX, 0000	BA0000	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0014 Stack +0 0000 Stack +2 19F5 (IP) = 000E	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0014 Stack +0 0000 Stack +2 19F5 (IP) = 0011
0011	CALL 0000	E8ECFF	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0014 Stack +0 0000 Stack +2 19F5 (IP) = 0011	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 Stack +0 0014 Stack +2 0000 Stack +4 19F5 (IP) = 0000
0000	Mov AH, 09	B409	(AX) = 1A07 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 Stack +0 0014 Stack +2 0000 Stack +4 19F5	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 Stack +0 0014 Stack +2 0000 Stack +4 19F5

			(IP) = 0000	(IP) = 0002
0002	Int 21	CD21	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 Stack +0 0014 Stack +2 0000 Stack +4 19F5 (IP) = 0002	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 Stack +0 0014 Stack +2 0000 Stack +4 19F5 (IP) = 0004
0004	RET	C3	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 Stack + 00014 Stack + 20000 Stack + 419F5 (IP) = 0004	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0014 Stack+00000 Stack + 219F5 (IP) = 0014
0014	Mov DX, 0010	BA1000	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0014 Stack +0 0000 Stack +2 19F5 (IP) = 0014	(AX) = 0907 (DX) = 0000 (DS) = 1A07 (CS) = 1A0A (SP) = 0014 Stack +0 0000 Stack +2 19F5 (IP) = 0017
0017	CALL 0000	E8E6FF	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 1A0A (SP) = 0014 Stack +0 0000 Stack +2 19F5 (IP) = 0017	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 Stack +0 001A Stack +2 0000 Stack +4 19F5 (IP) = 0000
0000	Mov AH, 09	B409	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 Stack +0 001A Stack +2 0000 Stack +4 19F5 (IP) = 0000	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 Stack +0 001A Stack +2 0000 Stack +4 19F5 (IP) = 0002

0002	INT 21	CD21	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 Stack +0 001A Stack +2 0000 Stack +4 19F5 (IP) = 0002	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 Stack +0 001A Stack +2 0000 Stack +4 19F5 (IP) = 0004
0004	RET	C3	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 1A0A (SP) = 0012 Stack +0 001A Stack +2 0000 Stack +4 19F5 (IP) = 0004	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 1A0A (SP) = 0014 Stack +0 0000 Stack +2 0000 (IP) = 001A
001A	RET FAR	CB	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 1A0A (SP) = 0014 Stack +0 0000 Stack +2 0000 (IP) = 001A	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 19F5 (SP) = 0018 Stack +0 0000 (IP) = 0000
0000	Int 20	CD20	(AX) = 0907 (DX) = 0010 (DS) = 1A07 (CS) = 19F5 (SP) = 0018 Stack +0 0000 (IP) = 0000	Программа завершилась.

Тексты исходных файлов программ hello1 и hello2 см. в приложении А.

Тексты файлов диагностических сообщений hello1.lst и hello2.lst см. в приложении В.

Выводы.

Были изучены трансляция, отладка и выполнение программ на языке Ассемблера. Рассмотрен синтаксис программ на этом языке.

ПРИЛОЖЕНИЕ А

ТЕКСТЫ ИСХОДНЫХ ФАЙЛОВ ПРОГРАММ

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 'Greetings from Samara Roman, 0383',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
```

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

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

ASSUME CS:CODE, SS:AStackAStack 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 – Samara Roman \$'DATA

ENDS

; Код программы

CODE SEGMENT

; Процедура печати строки

WriteMsg PROC NEAR

mov AH,9

int 21h ; Вызов функции DOS по прерываниюget

WriteMsg ENDP

; Главная процедура

Main PROC FAR

```
    push DS      ;\ Сохранение адреса начала PSP в стеке sub
    AX,AX        ; > для последующего восстановления порush
    AX           ;/ команде ret, завершающей процедуру. mov
    AX,DATA      ; Загрузка сегментного
    mov DS,AX    ; регистра данных.
    mov DX, OFFSET HELLO ; Вывод на экран первойcall
    WriteMsg     ; строки приветствия.
    mov DX, OFFSET GREETING ; Вывод на экран второйcall
    WriteMsg     ; строки приветствия.
    ret          ; Выход в DOS по команде,
                ; находящейся в 1-ом слове PSP.
```

Main ENDP

CODE ENDS

END Main

ПРИЛОЖЕНИЕ В

ТЕКСТЫ ФАЙЛОВ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: hello1.lst

Microsoft (R) Macro Assembler Version 5.10

9/15/21 21:22:35

Page 1-1

```

1
2      ; HELLO1.ASM - rĭa®йГ-- п ŷГабЁп гзГŷ-
      ®© ĭa®Ja ¬¬л « ŷ.a ŷ. N1
3      ;      ĭ® ºЁбжЁĭ«Ё-Г "БаеЁвГЄв
      га Є®-ĭмовГа "
4      ; *****
      *****
5      ; Ќ §- зГ-ЁГ: Цa®Ja ¬¬ д®a¬ЁaГв Ё ŷлŷ
      ®ºЁв - нЄa - ĭaЁŷГвбвŷЁГ
6      ;      ĭ®«м§®ŷ вГ«п б ĭ®¬®ймо дГ
      -ЄжЁЁ „Ĥ“ „лŷ®º бва®ЄЁ"
7      ;      (-®¬Га 09 ĭaГалŷ -ЁГ 21h)
      , Є®в®a п:
8      ;      - ®ŷГбĭГзЁŷ Гв ŷлŷ®º -
      нЄa - бва®ЄЁ бЁ¬ŷ®«®ŷ,
9      ;      § Є -зЁŷ ойГ©бп §- Є®¬
      "$";
10     ;      - ваГŷГв § º -Ёп ŷ aГJЁ
      бваГ ah -®¬Га дГ-ЄжЁЁ=09h,
11     ;      ŷ aГJЁбваГ dx - б¬Г
      йГ-Ёп ºaГб ŷлŷ®ºЁ¬®©
12     ;      бва®ЄЁ;
13     ;      - Ёбĭ®«м§ГГв aГJЁбва ах
      Ё -Г б®ea -пГв ГJ®
14     ;      б®ºГaĭЁ¬®Г.
15     ; *****
      *****
16
17     DOSSEG
      ; ‡ º -ЁГ бГJ¬Г-в®ŷ ĭ®º „Ĥ“
18     .MODEL SMALL
      ; Ь®ºГ«м ĭ ¬пвЁ-SMALL(Ь « п)
19     .STACK 100h
      ; ĤвŷГбвЁ ĭ®º ‘вГЄ 256 ŷ ©в
20     .DATA
      ; Ќ з «® бГJ¬Г-в º --ле
21 0000      Greeting LABEL BYTE
      ; ’ГЄбв ĭaЁŷГвбвŷЁп
22 0000 47 72 65 65 74 69      DB 'Greetings from Samara Roman, 038
      3',13,10,'$'
23      6E 67 73 20 66 72
24      6F 6D 20 53 61 6D

```

```

25    61 72 61 20 52 6F
26    6D 61 6E 2C 20 30
27    33 38 33 0D 0A 24
28                                     .CODE
                                     ; К з «® ЪГГ-Г-В €®¤
29 0000 B8 ---- R      mov ax, @data
                                     ; ‡ Јар§€ ѓ DS ¢аГб - з «
30 0003 8E D8          mov ds, ax
                                     ; ЪГГ-Г-В ¢ --ле
31 0005 BA 0000 R      mov dx, OFFSET Greeting
                                     ; ‡ Јар§€ ѓ dx б-ГйГ-Ёп
32
__Microsoft (R) Macro Assembler Version 5.10          9/15/21 21:22:35
                                     Page    1-2

                                     ; ¢аГб вГЅбв ЁаЁђГвбвђЁп
33 0008      DisplayGreeting:
34 0008 B4 09          mov ah, 9
                                     ; # дГ-ЅжЁЁ „Г‘ Ёз вЁ бва®ЅЁ
35 000A CD 21          int 21h
                                     ; ѓлђ®¤ - нЅа - ЁаЁђГвбвђЁп
36 000C B4 4C          mov ah, 4ch
                                     ; # дГ-ЅжЁЁ „Г‘ § ѓГайГ-Ёп Ёа®Ја ─л
37 000E CD 21          int 21h
                                     ; § ѓГайГ-ЁГ Ёа®Ја ─л Ё ѓле®¤ ѓ „Г‘
38                                     END
__Microsoft (R) Macro Assembler Version 5.10          9/15/21 21:22:35
                                     Symbols-1

```

Segments and Groups:

N a m e	Length	Align	Combine	Class
DGROUP	GROUP			
_DATA	0024	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: hello2.lst

Microsoft (R) Macro Assembler Version 5.10 9/15/21 22:21:31
Page 1-1

```
1          ; HELLO2 - "зГЎ- п ĩa®Ja ¬¬ N2 « Ў.a
           Ў.#1 ĩ® ¢ĖбжĖĭ«Ė-Г "ѠaeĖБГĖBга €®-ĭмов
           Га "
2          ;   Цa®Ja ¬¬ Ėбĭ®«м$ГГв ĩa®жГ¬Г
           аГ ¢«п ĭГз вĖ бBa®€Ė
3          ;
4          ;   '...Љ' Цђћѓћђђђђ
5
6 = 0024          EOFLine EQU '$'      ; ѠĭaГ¬Г«Г-ĖГ
           бĖ¬ŷ®«м-®© €®-бв -вЛ
7          ;   "Љ®-Гж
           бBa®€Ė"
8
9          ; 'вГ€ ĩa®Ja ¬¬Л
10
11          ASSUME CS:CODE, SS:AStack
12
13 0000          AStack SEGMENT STACK
14 0000 000C[          DW 12 DUP(?) ; Ѡвŷ®¬Ėбп 1
           2 б«®ŷ ĭ ¬пвĖ
15  ???
16          ]
17
18 0018          AStack ENDS
19
20          ; ,, --ЛГ ĩa®Ja ¬¬Л
21
22 0000          DATA SEGMENT
23
24          ; „ĖaГ€вĖŷл ®ĭĖб -Ėп ¢ --ле
25
```

```

26 0000 48 65 6C 6C 6F 20      HELLO    DB 'Hello Worlds!', 0AH, 0DH,
                                EOFLine
27      57 6F 72 6C 64 73
28      21 0A 0D 24
29 0010 53 74 75 64 65 6E      GREETING DB 'Student from 0383 - Samar
                                a R. D.$'

30      74 20 66 72 6F 6D
31      20 30 33 38 33 20
32      2D 20 53 61 6D 61
33      72 61 20 52 2E 20
34      44 2E 24
35 0031                        DATA    ENDS
36
37                        ; Ъ®¤ ĩa®Ja ¬¬л
38
39 0000                        CODE     SEGMENT
40                        ; Цa®жГ¤га Ĩз вĖ бва®СĖ
41 0000                        WriteMsg PROC NEAR
42 0000 B4 09                        mov  AH,9
43 0002 CD 21                        int  21h ; ,л$®ŷ дГ-СжĖĖ DO
                                S Ĩ® ĩaГалŷ -Ėo
44 0004 C3                        ret
45 0005                        WriteMsg ENDP
Microsoft (R) Macro Assembler Version 5.10      9/15/21 22:21:31
Page 1-2

```

```

46
47                        ; ř®«®ŷ- п ĩa®жГ¤га
48 0005                        Main    PROC FAR
49 0005 1E                        push DS ;\ '®ea -Г-ĖГ
                                ¤aГб - з « PSP ŷ бвГСГ
50 0006 2B C0                        sub  AX,AX ; > ¤«п Ĩ®б«Г¤
                                гоЙГЈ® ŷ®ббв -®ŷ«Г-Ėп Ĩ®
51 0008 50                        push AX ;/ С®¬ -¤Г re
                                t, $ ŷГaи ойГ© ĩa®жГ¤гаг.
52 0009 B8 ---- R                mov  AX,DATA ; ‡
                                Јаг$С бГЈ-Г-в-®Ј®
53 000C 8E D8                        mov  DS,AX ; a
                                ГЈĖбва ¤ --ле.
54 000E BA 0000 R                mov  DX, OFFSET HELLO ; ,
                                лŷ®¤ - нСа - Ĩаŷ®©
55 0011 E8 0000 R                call WriteMsg ; б
                                ва®СĖ ĩaĖŷГвбŷĖп.
56 0014 BA 0010 R                mov  DX, OFFSET GREETING ; ,
                                лŷ®¤ - нСа - ŷв®a®©
57 0017 E8 0000 R                call WriteMsg ; б
                                ва®СĖ ĩaĖŷГвбŷĖп.
58 001A CB                        ret ; ,
                                ле®¤ ŷ DOS Ĩ® С®¬ -¤Г,
59                        ; -
                                е®¤пЙГ©бп ŷ 1-®¬ б«®ŷГ PSP.

```



```

60 001B          Main  ENDP
61 001B          CODE  ENDS
62              END Main
Microsoft (R) Macro Assembler Version 5.10      9/15/21 22:21:31
Symbols-1

```

Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK	0018	PARA	STACK	
CODE	001B	PARA	NONE	
DATA	0031	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