**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.**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 1303 |  | Хабибуллина А.М |
| Преподаватель |  | Ефремов М.А. |

Санкт-Петербург

2022

## Цель работы.

Изучение основных принципов трансляции, отладки и выполнения 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 c 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

### 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. Подключение в DOSBOX папки MASM, где хранятся исходные файлы программ.
2. Трансляция программы >masm hello1.asm. Создание объектного файла и листинга: hello1.obj, lst1.lst
3. Компонуем загрузочный модуль >link hello1.obj
4. Запускаем программу >hello1.exe
5. С помощью отладчика afd проходимся пошагово по программе.
6. >afdpro hello1.exe

Таблицы с пошаговым рассмотрением программ представлены ниже: для hello1.asm и для hello2.asm.

hello1.asm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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  (DS) = 19F5  (IP) = 0010 | (AX) = 1A07 (DS) = 19F5 (IP) = 0013 |
| 0013 | MOV DS, AX | 8ED8 | (AX) = 1A07 (DS) = 19F5 (IP) = 0013 | (AX) = 1A07 (DS) = 1A07 (IP) = 0015 |
| 0015 | MOV DX, 0000 | BA0000 | (AX) = 1A07 (DS) = 1A07 (IP) = 0015  (CX) = 0000 | (AX) = 1A07 (DS) = 1A07 (IP) = 0018  (CX) = 004B |
| 0018 | MOV AH, 09 | B409 | (AX) = 1A07 (DS) = 1A07 (IP) = 0018 | (AX) = 0907 (DS) = 1A07 (IP) = 001A |
| 001A | INT 21 | B44C | (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  (CX) = 004B | (AX) = 0000  (DS) = 19F5  (IP) = 0010  (CX) = 0000 |

hello2.asm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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) = 19F5  (IP) = 0005  (SP) = 0018 | (AX) = 0000 (DS) = 19F5 (IP) = 0006  (SP) = 0016  STACK+0=19F5 |
| 0006 | SUB AX, AX | 2BC0 | (IP) = 0006  (AX) = 0000  STACK+0=19F5 | (IP) = 0008  (AX) = 0000  STACK+0=19F5 |
| 0008 | PUSH AX | 50 | (IP) = 0008  (SP) = 0016  STACK+0=19F5  STACK+2=0000 | (IP) = 0009  (SP) = 0014  STACK+0=0000  STACK+2=19F5 |
| 0009 | MOV AX, 1A07 | B8071A | (AX) = 0000 (DS) = 19F5  (IP) = 0009 | (AX) = 1A07 (DS) = 19F5  (IP) = 000C |
| 000C | MOV DS, AX | 8ED8 | (AX) = 1A07 (DS) = 19F5  (IP) = 000C | (AX) = 1A07 (DS) = 1A07 (IP) = 000E |
| 000E | MOV DX, 0000 | BA0000 | (AX) = 1A07 (DS) = 1A07  (IP) = 000E | (AX) = 1A07 (DS) = 1A07 (IP) = 0011 |
| 0011 | CALL 0000 | E8ECFF | (IP) = 0011  (SP) = 0014  STACK+0=0000  STACK+2=19F5  STACK+4=0000 | (IP) = 0000  (SP) = 0012  STACK+0=0014  STACK+2=0000  STACK+4=19F5 |
| 0000 | MOV AH, 09 | B409 | (AX) = 1A07 (DS) = 1A07  (IP) = 0000 | (AX) = 0907 (DS) = 1A07 (IP) = 0002 |
| 0002 | INT 21 | CD21 | (IP) = 0002 | (IP) = 0004 |
| 0004 | RET | C3 | (AX) = 0907 (DS) = 1A07  (IP) = 0004 | (AX) = 0907 (DS) = 1A07 (IP) = 0014 |
| 0014 | MOV DX, 0010 | BA1000 | (DX) = 0000  (IP) = 0014 | (DX) = 0010 (IP) = 0017 |
| 0017 | CALL 0000 | E8E6FF | (IP) = 0017  (SP) = 0014  STACK+0=0000  STACK+2=19F5  STACK+4=0000 | (IP) = 0000  (SP) = 0012  STACK+0=001A  STACK+2=0000  STACK+4=19F5 |
| 0000 | MOV AH, 09 | B409 | (IP) = 0000 | (IP) = 0002 |
| 0002 | INT 21 | CD21 | (IP) = 0002 | (IP) = 0004 |
| 0004 | RET | C3 | (IP) = 0004  (SP) = 0012  STACK+0=001A  STACK+2=0000  STACK+4=19F5 | (IP) = 001A  (SP) = 0014  STACK+0=0000  STACK+2=19F5  STACK+4=0000 |
| 001A | RET FAR | CB | (IP) = 001A  (SP) = 0014  (CS) = 1A0A  STACK+0=0000  STACK+2=19F5 | (IP) = 0000  (SP) = 0018  (CS) = 19F5  STACK+0=0000  STACK+2=0000 |
| 0000 | INT 20 | CD20 | (AX) = 0907 (DX) = 0010  (IP) = 0000  (CX) = 007B  (DS) = 1A07 | (AX) = 0000 (DX) = 0000  (IP) = 0005  (CX) = 0000  (DS) = 19F5 |

## Выводы.

В ходе лабораторной работы было изучено, как работает программа на языке Ассемблер. Была произведена трансляция, линковка и выполнение программы.

# Приложение А Исходный код программы

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.7303 - Иванов И.И.',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: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 4350 - $'

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. Листинг успешной трансляции**

**программами.**

hello1.lst

Microsoft (R) Macro Assembler Version 5.10 9/22/22 00:13:08

Page 1-1

; HELLO1.ASM - СѓРїСЂРѕС‰РµРЅРЅР°СЏ РІРµСЂСЃРё

СЏ СѓС‡РµР±РЅРѕР№ РїСЂРѕРіСЂР°РјРјС‹ Р»Р°Р±.СЂР

°Р±. N1

; РїРѕ РґРёСЃС†РёРїР»РёРЅРµ "РђСЂ

С…РёС‚РµРєС‚СѓСЂР° РєРѕРјРїСЊСЋС‚РµСЂР°"

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; РќР°Р·РЅР°С‡РµРЅРёРµ: РџСЂРѕРіСЂР°РјРјР° С„Рѕ

СЂРјРёСЂСѓРµС‚ Рё РІС‹РІРѕРґРёС‚ РЅР° СЌРєСЂР°Р

Ѕ РїСЂРёРІРµС‚СЃС‚РІРёРµ

; РїРѕР»СЊР·РѕРІР°С‚РµР»СЏ СЃ РїРѕР

јРѕС‰СЊСЋ С„СѓРЅРєС†РёРё Р”РћРЎ "Р’С‹РІРѕРґ СЃС

‚СЂРѕРєРё"

; (РЅРѕРјРµСЂ 09 РїСЂРµСЂС‹РІР°РЅРё

Рµ 21h), РєРѕС‚РѕСЂР°СЏ:

; - РѕР±РµСЃРїРµС‡РёРІР°РµС‚ РІС‹Р

ІРѕРґ РЅР° СЌРєСЂР°РЅ СЃС‚СЂРѕРєРё СЃРёРјРІРѕР»

РѕРІ,

; Р·Р°РєР°РЅС‡РёРІР°СЋС‰РµР№СЃСЏ

Р·РЅР°РєРѕРј "$";

; - С‚СЂРµР±СѓРµС‚ Р·Р°РґР°РЅРёСЏ

РІ СЂРµРіРёСЃС‚СЂРµ ah РЅРѕРјРµСЂР° С„СѓРЅРєС†Р

ёРё=09h,

; Р° РІ СЂРµРіРёСЃС‚СЂРµ dx - С

ЃРјРµС‰РµРЅРёСЏ Р°РґСЂРµСЃР° РІС‹РІРѕРґРёРјРѕ

Р№

; СЃС‚СЂРѕРєРё;

; - РёСЃРїРѕР»СЊР·СѓРµС‚ СЂРµРіРёС

ЃС‚СЂ ax Рё РЅРµ СЃРѕС…СЂР°РЅСЏРµС‚ РµРіРѕ

; СЃРѕРґРµСЂР¶РёРјРѕРµ.

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DOSSEG

; Р—Р°РґР°РЅРёРµ СЃРµРіРјРµРЅС‚РѕРІ РїРѕРґ Р”Рћ

РЎ

.MODEL SMALL

; РњРѕРґРµР»СЊ РїР°РјСЏС‚Рё-SMALL(РњР°Р»Р°СЏ)

.STACK 100h

; РћС‚РІРµСЃС‚Рё РїРѕРґ РЎС‚РµРє 256 Р±Р°Р№С‚

.DATA

; РќР°С‡Р°Р»Рѕ СЃРµРіРјРµРЅС‚Р° РґР°РЅРЅС‹С…

0000 Greeting LABEL BYTE

; РўРµРєСЃС‚ РїСЂРёРІРµС‚СЃС‚РІРёСЏ

0000 D0 92 D0 B0 D1 81 DB 'Р’Р°СЃ РїСЂРёРІРµС‚СЃС‚РІСѓРµС‚ СЃС‚.РіС

Ђ.7303 - РРІР°РЅРѕРІ Р.Р.',13,10,'$'

20 D0 BF D1 80 D0

B8 D0 B2 D0 B5 D1

82 D1 81 D1 82 D0

B2 D1 83 D0 B5 D1

82 20 D1 81 D1 82

2E D0 B3 D1 80 2E

37 33 30 33 20 2D

Microsoft (R) Macro Assembler Version 5.10 9/22/22 00:13:08

Page 1-2

20 D0 98 D0 B2 D0

B0 D0 BD D0 BE D0

B2 20 D0 98 2E D0

98 2E 0D 0A 24

.CODE ; РќР°С

‡Р°Р»Рѕ СЃРµРіРјРµРЅС‚Р° РєРѕРґР°

0000 B8 ---- R mov ax, @data ; Р—Р°

РіСЂСѓР·РєР° РІ DS Р°РґСЂРµСЃР° РЅР°С‡Р°Р»Р°

0003 8E D8 mov ds, ax ; СЃРµ

РіРјРµРЅС‚Р° РґР°РЅРЅС‹С…

0005 BA 0000 R mov dx, OFFSET Greeting ; Р—Р°

РіСЂСѓР·РєР° РІ dx СЃРјРµС‰РµРЅРёСЏ

; Р°РґС

ЂРµСЃР° С‚РµРєСЃС‚Р° РїСЂРёРІРµС‚СЃС‚РІРёСЏ

0008 DisplayGreeting:

0008 B4 09 mov ah, 9 ; # С„

СѓРЅРєС†РёРё Р”РћРЎ РїРµС‡Р°С‚Рё СЃС‚СЂРѕРєРё

000A CD 21 int 21h ; РІС‹Р

ІРѕРґ РЅР° СЌРєСЂР°РЅ РїСЂРёРІРµС‚СЃС‚РІРёСЏ

000C B4 4C mov ah, 4ch ; # С„

СѓРЅРєС†РёРё Р”РћРЎ Р·Р°РІРµСЂС€РµРЅРёСЏ РїСЂРѕ

РіСЂР°РјРјС‹

000E CD 21 int 21h ; Р·Р°Р

ІРµСЂС€РµРЅРёРµ РїСЂРѕРіСЂР°РјРјС‹ Рё РІС‹С…РѕР

ґ РІ Р”РћРЎ

END

Microsoft (R) Macro Assembler Version 5.10 9/22/22 00:13:08

Symbols-1

Segments and Groups:

N a m e Length Align Combine Class

DGROUP . . . . . . . . . . . . . GROUP

\_DATA . . . . . . . . . . . . 0047 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

48000 + 459260 Bytes symbol space free

0 Warning Errors

0 Severe Errors

hello2.lst

Microsoft (R) Macro Assembler Version 5.10 9/22/22 00:17:11

Page 1-1

; HELLO2 - РЈС‡РµР±РЅР°СЏ РїСЂРѕРіСЂР°РјРјР° N2

Р»Р°Р±.СЂР°Р±.#1 РїРѕ РґРёСЃС†РёРїР»РёРЅРµ "Р

ђСЂС…РёС‚РµРєС‚СѓСЂР° РєРѕРјРїСЊСЋС‚РµСЂР°"

; РџСЂРѕРіСЂР°РјРјР° РёСЃРїРѕР»СЊР·СѓР

µС‚ РїСЂРѕС†РµРґСѓСЂСѓ РґР»СЏ РїРµС‡Р°С‚Рё СЃС‚

СЂРѕРєРё

;

; РўР•РљРЎРў РџР РћР“Р РђРњРњР«

= 0024 EOFLine EQU '$' ; РћРїСЂРµРґРµР»РµРЅР

ёРµ СЃРёРјРІРѕР»СЊРЅРѕР№ РєРѕРЅСЃС‚Р°РЅС‚С‹

; "РљРѕРЅРµС† СЃС

‚СЂРѕРєРё"

; РЎС‚РµРє РїСЂРѕРіСЂР°РјРјС‹

ASSUME CS:CODE, SS:AStack

0000 AStack SEGMENT STACK

0000 000C[ DW 12 DUP('!') ; РћС‚РІРѕРґРёС‚СЃС

Џ 12 СЃР»РѕРІ РїР°РјСЏС‚Рё

0021

]

0018 AStack ENDS

; Р”Р°РЅРЅС‹Рµ РїСЂРѕРіСЂР°РјРјС‹

0000 DATA SEGMENT

; Р”РёСЂРµРєС‚РёРІС‹ РѕРїРёСЃР°РЅРёСЏ РґР°РЅРЅ

С‹С…

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

; РљРѕРґ РїСЂРѕРіСЂР°РјРјС‹

0000 CODE SEGMENT

; РџСЂРѕС†РµРґСѓСЂР° РїРµС‡Р°С‚Рё СЃС‚СЂРѕРєРё

0000 WriteMsg PROC NEAR

0000 B4 09 mov AH,9

0002 CD 21 int 21h ; Р’С‹Р·РѕРІ С„СѓРЅРєС†РёР

ё DOS РїРѕ РїСЂРµСЂС‹РІР°РЅРёСЋ

0004 C3 ret

0005 WriteMsg ENDP

; Р“РѕР»РѕРІРЅР°СЏ РїСЂРѕС†РµРґСѓСЂР°

Microsoft (R) Macro Assembler Version 5.10 9/22/22 00:17:11

Page 1-2

0005 Main PROC FAR

0005 1E push DS ;\ РЎРѕС…СЂР°РЅРµРЅРё

Рµ Р°РґСЂРµСЃР° РЅР°С‡Р°Р»Р° PSP РІ СЃС‚РµРєРµ

0006 2B C0 sub AX,AX ; > РґР»СЏ РїРѕСЃР»РµР

ґСѓСЋС‰РµРіРѕ РІРѕСЃСЃС‚Р°РЅРѕРІР»РµРЅРёСЏ РїРѕ

0008 50 push AX ;/ РєРѕРјР°РЅРґРµ ret

, Р·Р°РІРµСЂС€Р°СЋС‰РµР№ РїСЂРѕС†РµРґСѓСЂСѓ.

0009 B8 ---- R mov AX,DATA ; Р—Р°РіСЂС

ѓР·РєР° СЃРµРіРјРµРЅС‚РЅРѕРіРѕ

000C 8E D8 mov DS,AX ; СЂРµРіРёС

ЃС‚СЂР° РґР°РЅРЅС‹С….

000E BA 0000 R mov DX, OFFSET HELLO ; Р’С‹РІРѕР

ґ РЅР° СЌРєСЂР°РЅ РїРµСЂРІРѕР№

0011 E8 0000 R call WriteMsg ; СЃС‚СЂРѕР

єРё РїСЂРёРІРµС‚СЃС‚РІРёСЏ.

0014 BA 0010 R mov DX, OFFSET GREETING ; Р’С‹РІРѕР

ґ РЅР° СЌРєСЂР°РЅ РІС‚РѕСЂРѕР№

0017 E8 0000 R call WriteMsg ; СЃС‚СЂРѕР

єРё РїСЂРёРІРµС‚СЃС‚РІРёСЏ.

001A CB ret ; Р’С‹С…РѕР

ґ РІ DOS РїРѕ РєРѕРјР°РЅРґРµ,

; РЅР°С…РѕР

ґСЏС‰РµР№СЃСЏ РІ 1-РѕРј СЃР»РѕРІРµ PSP.

001B Main ENDP

001B CODE ENDS

END Main

Microsoft (R) Macro Assembler Version 5.10 9/22/22 00:17:11

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

47990 + 459267 Bytes symbol space free

0 Warning Errors

0 Severe Errors