**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. 4344 |  | Байдаков Г.И. |
| Преподаватель |  | Фирсов М.А. |

Санкт-Петербург

2025

## Цель работы

Освоение трансляцию, способы выполнения и отладки программ на языке Ассемблера процессора Intel X86.

## Задание

Часть 1

1. Загрузить файл hello1.asm каталога \лаборат\_работы 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).

1. 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.

1. Выполнить программу в автоматическом режиме путём набора строки

> имя\_файла.exe

и убедиться в её работоспособности (результат выполнения просмотреть в режиме отображения экрана пользователя, получаемого набором клавиш Ctrl^O).

1. 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 первой команды. Результаты прогона программы под управлением отладчика должны быть представлены в виде, показанном на примере одной команды в таблице 1.

Таблица 1 ‒ Запуск программы под управлением отладчика

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Адрес команды | Символический код команды | 16-ричный код команды | Содержимое регистров и ячеек памяти | |
| до вып. команды | после вып. команды |
| 0003 | MOV DS, AX | 8E D8 | AX = 2D87  DS = 2D75  IP = 0003 | AX = 2D87  DS = 2D87  IP = 0005 |

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 и убедиться в её 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 первой команды записать содержимое сегментных регистров процессора.
6. Результаты прогона программы под управлением отладчика должны быть подписаны преподавателем.

## Выполнение работы

Часть 1

1. Запуск DOSBox, монтирование диска и переключение на него (см. Рис. 1):



Рисунок 1 – Монтирование диска

1. Просмотр файла в режиме редактирования и преобразование строки-приветствия (см. Рис. 2):



Рисунок 2 – Изменённая строка-приветствие.

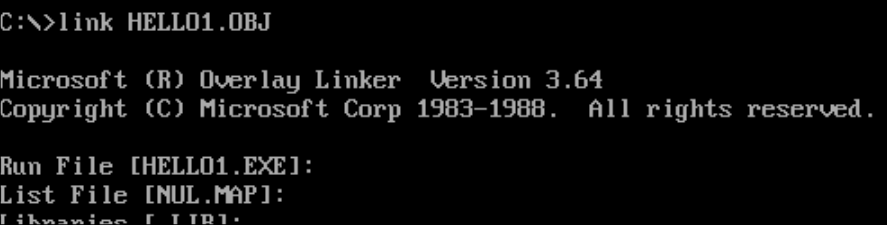
1. Трансляция программы с созданием объектного файла и файла диагностических сообщений (см. Рис. 3):



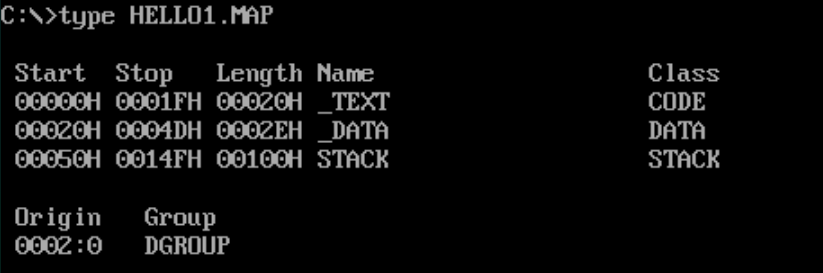
Рисунок 3 – Трансляция программы

В ходе трансляции синтаксических ошибок обнаружено не было.

1. Компоновка загрузочного модуля с созданием файла карты памяти (см. Рис. 4):

Рисунок 4 – Компоновка загрузочного модуля

Получившуюся карту памяти см. на Рис. 5.

 Рисунок 5 – Карта памяти

Сегменты программы расположены в следующем порядке: сегмент кода, сегмент данных, сегмент стека. Сегмент кода занимает 32 байта, сегмент данных – 46 байт, сегмент сетка – 256 байт.

1. Выполнение программы в автоматическом режиме (см. Рис. 6):



Рисунок 6 – Выполнение программы в автоматическом режиме

1. Выполнение программы в пошаговом режиме под управлением отладчика afd.

можно выполнить программу в пошаговом режиме под управлением отладчика. Содержимое регистров перед выполнением первой команды см. на Рис. 7. Результаты прогона программы под управлением отладчика см в Табл. 2.

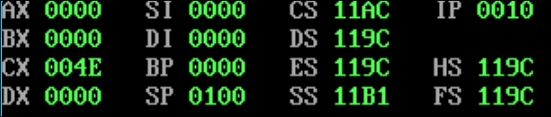


Рисунок 7 – Содержимое регистров перед выполнением программы

Таблица 2 ‒ Изменение регистров после выполнения команд

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Адрес команды | Символический код команды | 16-ричный код команды | Содержимое регистров и ячеек памяти | |
| до вып. команды | после вып. команды |
| 0010 | MOV AX,11AE | B8AE11 | AX = 0000  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 = 0000  IP = 0015 | DX = 0000  IP = 0018 |
| 0018 | MOV AH,09 | B409 | AX = 11AE  IP = 0018 | AX = 09AE  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 | AX = 4CAE  CX = 004E  DS = 11AE  IP = 001E | AX = 0000  CX = 0000  DS = 119C  IP = 0010 |

Исходный код и файл листинга hello1 см. в приложении А.

Часть 2

1. Просмотр файла в режиме редактирования и преобразование строки-приветствия (см. Рис. 8):

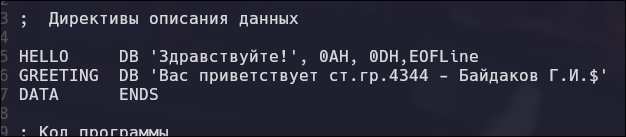


Рисунок 8 – Измененная строка-приветствие

1. Трансляция программы с созданием объектного файла и файла диагностических сообщений (см. Рис. 9):

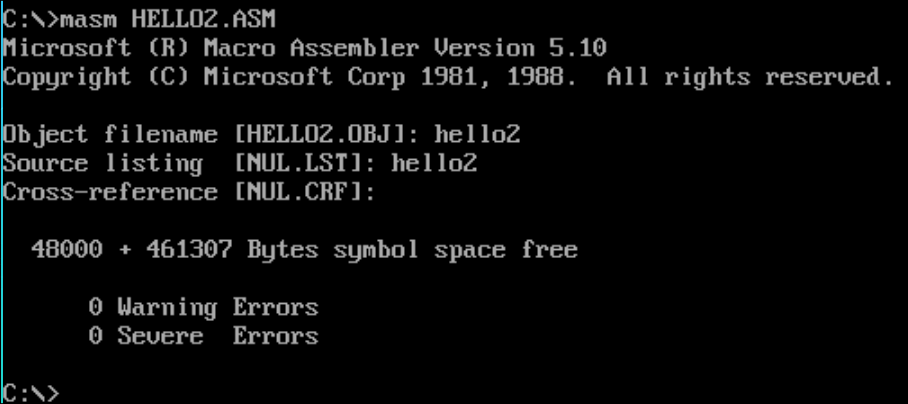


Рисунок 9 – Трансляция программы

В ходе трансляции синтаксических ошибок обнаружено не было.

1. Компоновка загрузочного модуля с созданием файла карты памяти (см. Рис. 10):

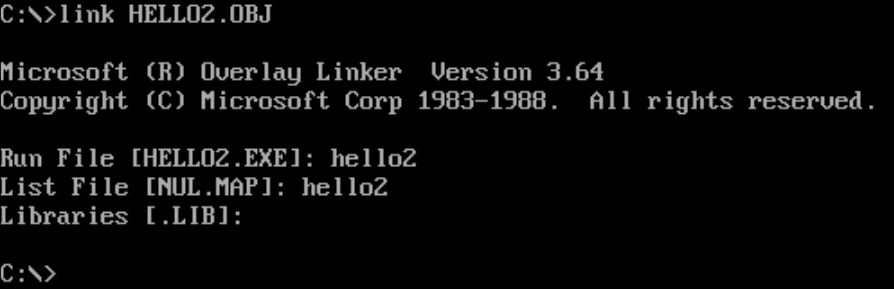


Рисунок 10 – Компоновка загрузочного модуля

1. Выполнение программы в автоматическом режиме (см. Рис. 11):

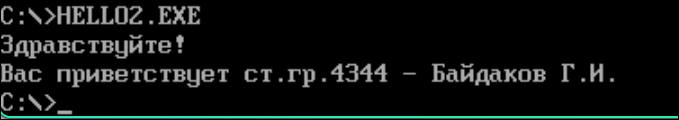


Рисунок 11 – Выполнение программы

1. Выполнение программы в пошаговом режиме под управлением отладчика afd. Результаты прогона программы под управлением отладчика см. в Табл. 3. Состояние регистров и ячеек памяти перед началом выполнения программы см. на Рис. 12.

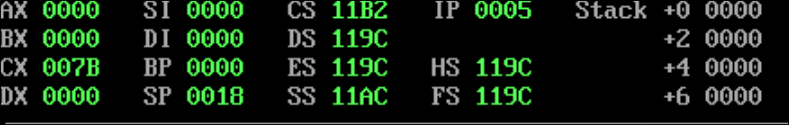
Рисунок 12 ‒ Содержимое регистров перед выполнением программы

Таблица 3 ‒ Изменение регистров и ячеек памяти после выполнения команд

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Адрес команды | Символический код команды | 16-ричный код команды | Содержимое регистров и ячеек памяти | |
| до вып. команды | после вып. команды |
| 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 | AX = 11AE  DS = 119C  IP = 000C | AX = 11AE  DS = 11AE  IP = 000E |
| 000E | MOV DX,0000 | BA0000 | DX = 0000  IP = 000E | DX = 0000  IP = 0011 |
| 0011 | CALL 0000 | 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,09 | B409 | AX = 11AE  SP = 0012  IP = 0000 | AX = 09AE  SP = 0012  IP = 0002 |
| 0002 | INT 21 | CD21 | IP = 0002 | IP = 0004 |
| 0004 | RET | C3 | IP = 0004  SP = 0012  Stack +0 0014  +2 0000  +4 119C  +6 0000 | IP = 0014  SP = 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 | E8E6FF | 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, 09 | B409 | AX = 09AE  IP = 0000 | AX = 09AE  IP = 0002 |
| 0002 | INT 21 | CD21 | IP = 0002 | IP = 0004 |
| 0004 | RET | C3 | SP = 0012  IP = 0004  Stack +0 001A  +2 0000  +4 119C  +6 0000 | SP = 0014  IP = 001A  Stack +0 0000  +2 119C  +4 0000  +6 0000 |
| 001A | RET FAR | CB | SP = 0014  IP = 001A  Stack +0 0000  +2 119C  +4 0000  +6 0000 | SP = 0019  IP = 0000  Stack +0 0000  +2 0000  +4 0000  +6 0000 |

Исходный код и файл листинга hello2 см. в приложении А.

## Вывод

В ходе выполнения лабораторной работы были освоены трансляция, выполнение и отладка программ на языке Ассемблера процессора Intel X86.

# 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.4344 – Байдаков Г.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

Название файла: hello1.lst

Microsoft (R) Macro Assembler Version 5.10 9/6/25 18:15:09

Page 1-1

; HELLO1.ASM - упрощенная версия учебной прогр

аммы лаб.раб. N1

; по дисциплине "Архитектура комп

ьютера"

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

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

; Назначение: Программа формирует и выводит на

экран приветствие

; пользователя с помощью функции ДО

С "Вывод строки"

; (номер 09 прерывание 21h), котора

я:

; - обеспечивает вывод на экран ст

роки символов,

; заканчивающейся знаком "$";

; - требует задания в регистре ah

номера функции=09h,

; а в регистре dx - смещения а

дреса 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

0000 Greeting LABEL BYTE

; Текст приветствия

0000 82 A0 E1 20 AF E0 DB 'Вас приветствует ст.гр.4344 — Байдаков Г.И.',13,10,'$'

A8 A2 A5 E2 E1 E2

A2 E3 A5 E2 20 E1

E2 2E A3 E0 2E 34

33 34 35 20 2D 20

81 AE A4 A0 E0 A5

A2 A0 20 91 2E 80

2E 0D 0A 24

.CODE ; Начал

о сегмента кода

0000 B8 ---- R mov ax,@data ; Загру

зка в DS адреса начала

0003 8E D8 mov ds,ax ; сегме

нта данных

0005 BA 0000 R mov dx,OFFSET Greeting ; Загру

зка в dx смещения

Microsoft (R) Macro Assembler Version 5.10 9/6/25 18:15:09

Page 1-2

; адрес

а текста приветствия

0008 DisplayGreeting:

0008 B4 09 mov ah,9 ; # фун

кции ДОС печати строки

000A CD 21 int 21h ; вывод

на экран приветствия

000C B4 4C mov ah,4ch ; # фун

кции ДОС завершения программы

000E CD 21 int 21h ; завер

шение программы и выход в ДОС

END

Microsoft (R) Macro Assembler Version 5.10 9/6/25 18:15:09

Symbols-1

Segments and Groups:

N a m e Length Align Combine Class

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

\_DATA . . . . . . . . . . . . 002E 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

47992 + 461315 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 0004DH 0002EH \_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.4344 – Байдаков Г.И.$'

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 9/6/25 21:58:26

Page 1-1

; HELLO2 -Учебная программа N2 лаб.раб.#1 по ди

сциплине "Архитектура компьютера"

; Программа использует процедуру для п

ечати строки

;

; ТЕКСТ ПРОГРАММЫ

= 0024 EOFLine EQU '$' ; Определение символь

ной константы

; "Конец строки"

; Стек программы

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 87 A4 E0 A0 A2 E1 HELLO DB 'Здравствуйте!', 0AH, 0DH,EOFLine

E2 A2 E3 A9 E2 A5

21 0A 0D 24

0010 82 A0 E1 20 AF E0 GREETING DB 'Вас приветствует ст.гр.4344 – Байдаков Г.И.$'

A8 A2 A5 E2 E1 E2

A2 E3 A5 E2 20 E1

E2 2E A3 E0 2E 34

33 34 35 20 2D 20

81 AE A4 A0 E0 A5

A2 A0 20 91 2E 80

2E 24

003C DATA ENDS

; Код программы

0000 CODE SEGMENT

ASSUME CS:CODE, DS:DATA, SS:AStack

; 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

Microsoft (R) Macro Assembler Version 5.10 9/6/25 21:58:26

Page 1-2

; 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

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 9/6/25 21:58:26

Symbols-1

Segments and Groups:

N a m e Length Align Combine Class

ASTACK . . . . . . . . . . . . . 0018 PARA STACK

CODE . . . . . . . . . . . . . . 001B PARA NONE

DATA . . . . . . . . . . . . . . 003C 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

48000 + 461307 Bytes symbol space free

0 Warning Errors

0 Severe Errors

Название файла: hello2.map

Start Stop Length Name Class

00000H 00017H 00018H ASTACK

00020H 0005BH 0003CH DATA

00060H 0007AH 0001BH CODE

Program entry point at 0006:0005