

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. 1381

Тарасов К.О.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Задание

Просмотреть программу hello1.asm, которая формирует и выводит на экран приветствие пользователя.

Разобраться в структуре и реализации каждого сегмента программы. Строку-приветствие преобразовать в соответствии со своими личными данными.

Протранслировать программу с созданием объектного файла и файла диагностических сообщений (файла листинга).

Скомпоновать загрузочный модуль с созданием карты памяти и исполняемого файла hello1.exe.

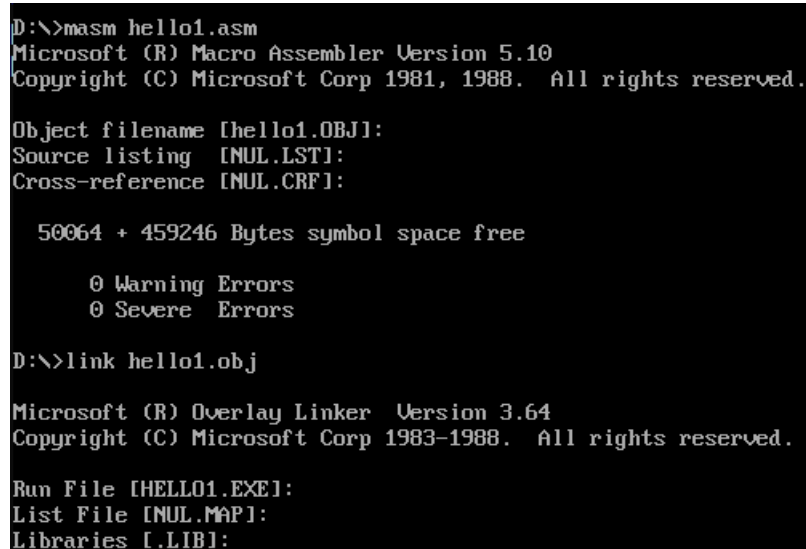
Выполнить программу в автоматическом режиме путем набора строки убедиться в корректности ее работы и зафиксировать результат выполнения в протоколе.

Запустить выполнение программы под управлением отладчика.

Ход работы

Каталог смонтирован в эмулятор DosBox с указанием пути командой mount

Трансляция программы, компоновка загрузочного модуля показаны на рис. 1.



```
D:\>masm hello1.asm
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [hello1.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

    50064 + 459246 Bytes symbol space free

    0 Warning Errors
    0 Severe Errors

D:\>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]:
Libraries [.LIB]:
```

Рис 1. – Получение исполняемого файла

Результат работы показан на рисунке 2

```
D:\>hello1.exe
You are welcomed by student from group 0383 – Tarasov K.O.
```

Рис. 2 — Результат работы программы

Запуск отладчика командой:

afopro hello1.exe

Начальное содержимое сегментных регистров:

CS = 1A05

DS = 19F5

ES = 19F5

SS = 1A0B

Результат прогона программы представлена в таблице 1:

Табл. 1. Результат прогона программы.

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	IP = 0018 DX = 0000 DS = 1A07 AX = 1A07	IP = 001A DX = 0000 DS = 1A07 AX = 0907
001A	INT 21	CD21	DX = 0000 DS = 1A07 AX = 0907	DX = 0000 DS = 1A07 AX = 0907

			IP = 001A	IP = 001C
001C	MOV AH, 4C	B44C	DX = 0000 DS = 1A07 IP = 001C AX = 0907	DX = 0000 DS = 1A07 IP = 001E AX = 4C07
001E	INT 21	CD21	IP = 001E DX = 0000 DS = 1A07 AX = 4C07	Программа завершилось

Трансляция программы, компоновка загрузочного модуля у hello2 выполнена аналогично. Результат работы программы:

```
D:\>hello2.exe
Hello Worlds!
Student from 0383 - Tarasov Konstantin
```

Рис. 3 — Результат работы программы

Начальное содержание сегментных регистров hello2.exe:

CS = 1A0B

DS = 19F5

ES = 19F5

SS = 1A05

Результат работы программы под управлением отладчика:

Табл. 2. Результат работы программы.

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 = 1A0B SP = 0018 IP = 0005 Stack +0 000	AX = 0000 DX = 0000 DS = 19F5 CS = 1A0B SP = 0016 IP = 0006 Stack +0 19F5
0006	SUB AX, AX	2BC0	AX = 0000 DX = 0000 DS = 19F5 CS = 1A0B SP = 0016 IP = 0006 Stack +0 19F5	AX = 0000 DX = 0000 DS = 19F5 CS = 1A0B SP = 0016 IP = 0008 Stack +019F5

0008	PUSH AX	50	AX = 0000 DX = 0000 DS = 19F5 CS = 1A0B SP = 0016 IP = 0008 Stack +0 19F5	AX = 0000 DX = 0000 DS = 19F5 CS = 1A0B SP = 0014 IP = 0009 Stack +0 0000 Stack +2 19F5
0009	MOV AX, 1A07	B8071A	IP = 0009 AX = 0000 DX = 0000 DS = 19F5 CS = 1A0B SP = 0014 Stack +0 0000 Stack +2 19F5	IP = 000C AX = 1A07 DX = 0000 DS = 19F5 CS = 1A0B SP = 0014 Stack +0 0000 Stack +2 19F5
000C	MOV DS, AX	8ED8	IP = 000C AX = 1A07 DX = 0000 DS = 19F5 CS = 1A0B SP = 0014 Stack +0 0000 Stack +2 19F5	IP = 000E AX = 1A07 DX = 0000 DS = 1A07 CS = 1A0B SP = 0014 Stack +0 0000 Stack +2 19F5
000E	MOV DX, 0000	BA0000	IP = 000E AX = 1A07 DX = 0000 DS = 1A07 CS = 1A0B SP = 0014 Stack +0 0000 Stack +2 19F5	IP = 0011 AX = 1A07 DX = 0000 DS = 1A07 CS = 1A0B SP = 0014 Stack +0 0000 Stack +2 19F5
0011	CALL 0000	E8ECFF	IP = 0011 AX = 1A07 DX = 0000 DS = 1A07 CS = 1A0B SP = 0014 Stack +0 0000 Stack +2 19F5	IP = 0000 AX = 1A07 DX = 0000 DS = 1A07 CS = 1A0B SP = 0012 Stack +0 0014 Stack +2 0000 Stack +4 19F5
0000	MOV AH, 09	B409	IP = 0000 AX = 1A07 DX = 0000 DS = 1A07 CS = 1A0B SP = 0012 Stack +0 0014 Stack +2 0000	AX = 0907 IP = 0002 DX = 0000 DS = 1A07 CS = 1A0B SP = 0012 Stack +0 0014 Stack +2 0000

			Stack +4 19F5	Stack +4 19F5
0002	INT 21	CD21	AX = 0907 IP = 0002 DX = 0000 DS = 1A07 CS = 1A0B SP = 0012 Stack +0 0014 Stack +2 0000 Stack +4 19F5	AX = 0907 IP = 0004 DX = 0000 DS = 1A07 CS = 1A0B SP = 0012 Stack +0 0014 Stack +2 0000 Stack +4 19F5
0004	RET	C3	AX = 0907 IP = 0004 DX = 0000 DS = 1A07 CS = 1A0B SP = 0012 Stack +0 0014 Stack +2 0000 Stack +4 19F5	AX = 0907 IP = 0014 DX = 0000 DS = 1A07 CS = 1A0B SP = 0014 Stack +0 0000 Stack +2 19F5 Stack +4 0000
0014	MOV DX, 0010	BA1000	AX = 0907 IP = 0014 DX = 0000 DS = 1A07 CS = 1A0B SP = 0014 Stack +0 0000 Stack +2 19F5	AX = 0907 IP = 0017 DX = 0010 DS = 1A07 CS = 1A0B SP = 0014 Stack +0 0000 Stack +2 19F5
0017	CALL 0000	E8E6FF	AX = 0907 IP = 0017 DX = 0010 DS = 1A07 CS = 1A0B SP = 0014 Stack +0 0000 Stack +2 19F5	AX = 0907 IP = 0000 DX = 0010 DS = 1A07 CS = 1A0B SP = 0012 Stack +0 001A Stack +2 0000 Stack +4 19F5
0000	MOV AH, 09	B409	AX = 0907 IP = 0000 DX = 0010 DS = 1A07 CS = 1A0B SP = 0012 Stack +0 001A Stack +2 0000 Stack +4 19F5	AX = 0907 IP = 0002 DX = 0010 DS = 1A07 CS = 1A0B SP = 0012 Stack +0 001A Stack +2 0000 Stack +4 19F5
0002	INT 21	CD21	AX = 0907 IP = 0002 DX = 0010 DS = 1A07	AX = 0907 IP = 0004 DX = 0010 DS = 1A07

			CS = 1A0B SP = 0012 Stack +0 001A Stack +2 0000 Stack +4 19F5	CS = 1A0B SP = 0012 Stack +0 001A Stack +2 0000 Stack +4 19F5
0004	RET	C3	AX = 0907 IP = 0004 DX = 0010 DS = 1A07 CS = 1A0B SP = 0012 Stack +0 001A Stack +2 0000 Stack +4 19F5	AX = 0907 IP = 001A DX = 0010 DS = 1A07 CS = 1A0B SP = 0014 Stack +0 0000 Stack +2 19F5
001A	RET Far	CB	AX = 0907 IP = 001A DX = 0010 DS = 1A07 CS = 1A0B SP = 0014 Stack +0 0000 Stack +2 19F5	AX = 0907 IP = 0000 DX = 0010 DS = 1A07 CS = 19F5 SP = 0018 Stack +0 0000
0000	INT 20	CD20	AX = 0907 IP = 0000 DX = 0010 DS = 1A07 CS = 19F5 SP = 0018 Stack +0 0000	Программа завершилась

Выводы.

В ходе выполнения работы были изучены основы ассемблера, созданы исполняемые файлы для двух программ на языке ассемблер с помощью эмулятора DosBox.

ПРИЛОЖЕНИЕ А

Текст 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 'You are welcomed by student from group 0383 - Tarasov K.O.',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 ДОО печати строки
int 21h                  ; вывод на экран приветствия
mov ah, 4ch              ; # функции ДОО завершения программы
int 21h                  ; завершение программы и выход в ДОО
END
```

hello1.lst:

#Microsoft (R) Macro Assembler Version 5.10

9/14/21 22:50:45

Page 1-1

PIPμCЪCГРë

P»P°P±.CЪP

CГC

; HELLO1.ASM - CГPиCЪPsC%oPμPSPSP°CЦ

CЦ CГC‡PμP±PSPsPNo PиCЪPsPиCЪP°PjPjC<

°P±. N1

; PиPs PГPëCГC‡PëPиP»PëPSPμ "PђCЪ

C...PëC,PμPëC,CГCЪP° PëPsPjPиCЪCЪC,PμCЪP°"

; *****

; PќP°P·PSP°C‡PμPSPëPμ: PμCЪPsPиCЪP°PjPjP° C,,Ps

CЪPjPëCЪCГPμC, Pë PIC<PIPsPГPëC, PSP° CќPëCЪP°P

S PиCЪPëPIPμC,CГC,PiPëPμ

; PиPsP»CЪP·PsPIP°C,PμP»CЦ CГ PиPsP

jPsC%oCЪCЪ C,,CГPSPëC‡PëPë P”PђPŸ "P’C<PIPsPГ

,CЪPsPëPë"

; (PSPsPjPμCЪ 09 PиCЪPμCЪC<PIP°PSPë

Pμ 21h), PëPsC,PсCЪP°CЦ:

```

;          - PsP±PμCΓPřPμC‡PëPIP°PμC, PIC<P
IPsPr PSP° CKPeCḂP°PS CΓC,CḂPsPePë CΓPëPjPIPsP»
PsPI,
;          P·P°PeP°PSC‡PëPIP°CḂC%PμPNᵒCΓCЦ
P·PSP°PePsPj "$";
;          - C,CḂPμP±CřPμC, P·P°PrP°PSPëCЦ
PI CḂPμPiPëCΓC,CḂPμ ah PSPsPjPμCḂP°

C,,CřPSPeC‡P

ëPë=09h,
;          P° PI CḂPμPiPëCΓC,CḂPμ dx - C
ÍPjPμC%PμPSPëCЦ P°PrCḂPμCΓP° PIC<PIPsPrPëPjPs
PNᵒ
;          CΓC,CḂPsPePë;
;          - PëCΓPřPsP»CHḂP·CřPμC, CḂPμPiPëC
ÍC,CḂ ax Pë PSPμ CΓPsC...CḂP°PSCЦPμC, PμPiPs
;          CΓPsPrPμCḂP¶PëPjPsPμ.
; *****
*****

DOSSEG
; P—P°PrP°PSPëPμ CΓPμPiPjPμPSC,PsPI PřPsPr P”Ph
PŸ
.MODEL SMALL
; PḂPsPrPμP»CHḂ PřP°PjCЦC,Pë-SMALL(PḂP°P»P°CЦ)
.STACK 100h
; PhC,PIPμCΓC,Pë PřPsPr PŸC,PμPe 256 P±P°PNᵒC,
.DATA
; K°P°C‡P°P»Ps CΓPμPiPjPμPSC,P° PrP°PSPSC<C...
0000 Greeting LABEL BYTE
; PŸPμPeCΓC, PřCḂPëPIPμC,CΓC,PIPëCЦ

```

0000 59 6F 75 20 61 72 DB 'You are welcomed by student from group 0
383 - Tarasov K.O.',13,10,'\$'

65 20 77 65 6C 63
6F 6D 65 64 20 62
79 20 73 74 75 64
65 6E 74 20 66 72
6F 6D 20 67 72 6F
75 70 20 30 33 38
33 20 2D 20 54 61

#Microsoft (R) Macro Assembler Version 5.10

9/14/21 22:50:45

Page 1-2

72 61 73 6F 76 20
4B 2E 4F 2E 0D 0A
24

```
.CODE ; PkP°C
;P°P»Ps CÍPµPiPjPµPSC,P° PePsPrP°
0000 B8 ---- R      mov ax, @data ; P—P°
PiCᵀCÍP·PeP° PI DS P°PrCᵀPµCÍP° PSP°C‡P°P»P°
0003 8E D8          mov ds, ax ; CÍPµ
PiPjPµPSC,P° PrP°PSPSC<C...
0005 BA 0000 R      mov dx, OFFSET Greeting ; P—P°
PiCᵀCÍP·PeP° PI dx CÍPjPµC%oPµPSPëCİİ
; P°PrC
ᵀPµCÍP° C,PµPeCÍC,P° PİCᵀPëPIPµC,CÍC,PIPëCİİ
0008 DisplayGreeting:
0008 B4 09          mov ah, 9 ; # C,,
CÍPSPeC†PëPë P”PhPŸ PiPµC‡P°C,Pë CÍC,CᵀPsPePë
000A CD 21          int 21h ; PIC<P
```

```

IPsPr PSP° CKPeCṪP°PS PṛCṪPṛPIPμC,CṼC,PṚPṛCṼ
000C B4 4C          mov ah,4ch          ; # C,,
CṼPSPeCṼPṛPṛ PṛPhPṼ P·P°PIPμCṪCṼPμPSPṛCṼ
PṛCṪPs
PṛCṪP°PjPjCṼ
000E CD 21          int 21h          ; P·P°P
IPμCṪCṼPμPSPṛPμ PṛCṪPsPṛCṪP°PjPjCṼ Pṛ PICṼC...
PsP
r PI PṛPhPṼ
END

```

#Microsoft (R) Macro Assembler Version 5.10 9/14/21 22:50:45

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine	Class
DGROUP	GROUP			
_DATA	003D	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

47994 + 459266 Bytes symbol space free

ПРИЛОЖЕНИЕ Б

Текст компонентов программы hello2.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 - Tarasov Konstantin \$'

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

9/14/21 22:51:42

Page 1-1

; HELLO2 - PJC‡PμP±PSP°CЦ PïCЪPsPiCЪP°PjPjP° N2
P»P°P±.CЪP°P±.#1 PïPs PrPëCГC†PëPïP»PëPSPμ "P
ђCЪC...PëC,PμPëC,CѓCЪP° PëPsPjPïCЪCЪC,PμCЪP°"
; PцCЪPsPiCЪP°PjPjP° PëCГPïPsP»CЪP·CѓP
μC, PïCЪPsC†PμPrCѓCЪCѓ PrP»CЦ PïPμC‡P°C,Pë

CГC,

```

CЉPsPePë
;
; PŷP•PљPŸPŷ PцP PhP“P PhPњPњP«

= 0024          EOFLine EQU '$'      ; PhPiCЉPμPrPμP»PμPSP
                ěPμ CЃPëPjPIPsP»CHBPSPsPNĚ PePsPSCЃC,P°PSC,C<
                ; "PљPsPSPμC† CЃC
                ,CЉPsPePë"

; PŸC,PμPe PiCЉPsPiCЉP°PjPjC<

ASSUME CS:CODE, SS:AStack

0000          AStack SEGMENT STACK
0000 000C[                DW 12 DUP('!' ) ; PhC,PIPsPrPëC,CЃC
                Ъ 12 CЃP»PsPI PïP°PjCЃC,Pë

                0021

                ]

0018          AStack ENDS

; P”P°PSPSC<Pμ PiCЉPsPiCЉP°PjPjC<

0000          DATA SEGMENT

; P”PëCЉPμPeC,PëPIC< PsPiPëCЃP°PSPëCЃ PrP°PSPS
                C<C...

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 0383 - Tarasov Konst
      antin $'
```

```
74 20 66 72 6F 6D
```

```
20 30 33 38 33 20
```

```
2D 20 54 61 72 61
```

```
73 6F 76 20 4B 6F
```

```
6E 73 74 61 6E 74
```

```
69 6E 20 24
```

```
0038          DATA    ENDS
```

```
; PЉPsPr PïCтPsPiCтP°PjPjC<
```

```
0000          CODE     SEGMENT
```

```
; PұCтPsC†PμPrCrCтP° PïPμC‡P°C,Pë CÍC,CтPsPePë
```

```
0000          WriteMsg PROC NEAR
```

```
0000 B4 09                      mov  AH,9
```

```
0002 CD 21                      int  21h ; P'C<P·PsPI C,,CrPSPeC†PëP
```

```
ë DOS PïPs PïCтPμCтC<PIP°PSPëCт
```

```

0004 C3                ret
0005                WriteMsg ENDP

; P“PsP»PsPIPSP°CЦ PïCтPsC†PμPrCfCтP°
0005                Main    PROC FAR
0005 1E                push DS    ;\ PŸPsC...CтP°PSPμPSPë
Pμ P°PrCтPμCГP° PSP°C‡P°P»P° PSP PI CГC,PμPePμ
0006 2B C0            sub  AX,AX    ; > PrP»CЦ PïPsCГP»PμP
rCfCтC%PμPiPs PIPsCГCГC,P°PSPsPIP»PμPSPëCЦ
PïPs
0008 50                push AX    ;/ PePsPjP°PSPrPμ ret
,P·P°PIPμCтC€P°CтC%PμPNë PïCтPsC†PμPrCfCтCГ.
0009 B8 ---- R        mov  AX,DATA    ; P—P°PiCтC
fP·PeP° CГPμPiPjPμPSC,PSPsPiPs
000C 8E D8            mov  DS,AX    ; CтPμPiPëC
ГC,CтP° PrP°PSPSC<C....
000E BA 0000 R        mov  DX, OFFSET HELLO    ; P’C<PIPsP
r PSP° CКPeCтP°PS PïPμCтPIPsPNë
0011 E8 0000 R        call WriteMsg    ; CГC,CтPsP
ePë PïCтPëPIPμC,CГC,PIPëCЦ.
0014 BA 0010 R        mov  DX, OFFSET GREETING ; P’C<PIPsP
r PSP° CКPeCтP°PS PIC,PsCтPsPNë
0017 E8 0000 R        call WriteMsg    ; CГC,CтPsP
ePë PïCтPëPIPμC,CГC,PIPëCЦ.
001A CB                ret            ; P’C<C...PsP
r PI DOS PïPs PePsPjP°PSPrPμ,

```

; PSP°C...PsP

$$rC\dot{U}C\%_0P_{\mu}PN_{\circ}C\dot{I}C\dot{U}PI\ 1-PsP_j\ C\dot{I}P\gg P_sPIP_{\mu}\ PSP.$$

```
001B                                Main    ENDP
```

```
001B                                CODE    ENDS
```

END Main

```
#Microsoft (R) Macro Assembler Version 5.10
```

9/14/21 22:51:42

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK.....	0018	PARA		STACK
CODE.....	001B	PARA		NONE
DATA.....	0038	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

47986 + 459271 Bytes symbol space free

0 Warning Errors

0 Severe Errors