

**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. 1383

\_\_\_\_\_

Кошкин Е.А.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2022

### **Цель работы.**

Получение навыков трансляции, выполнения и отладки программ на языке Ассемблера.

### **Задание.**

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. Обычные команды выполняются по F1 (Step), а вызовы обработчиков прерываний (Int) - по F2 (StepProc), чтобы не входить внутрь обработчика прерываний. Продвижение по сегментам экранной формы отладчика выполняется с помощью клавиш F7 – F10 (up, down, left, right). Перезапуск программы в отладчике выполняется клавишей F3 (Retrieve). Выход из отладчика - по команде Quit.

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. hello1.ams

Рисунок 1 — Начальное состояние регистров hello1

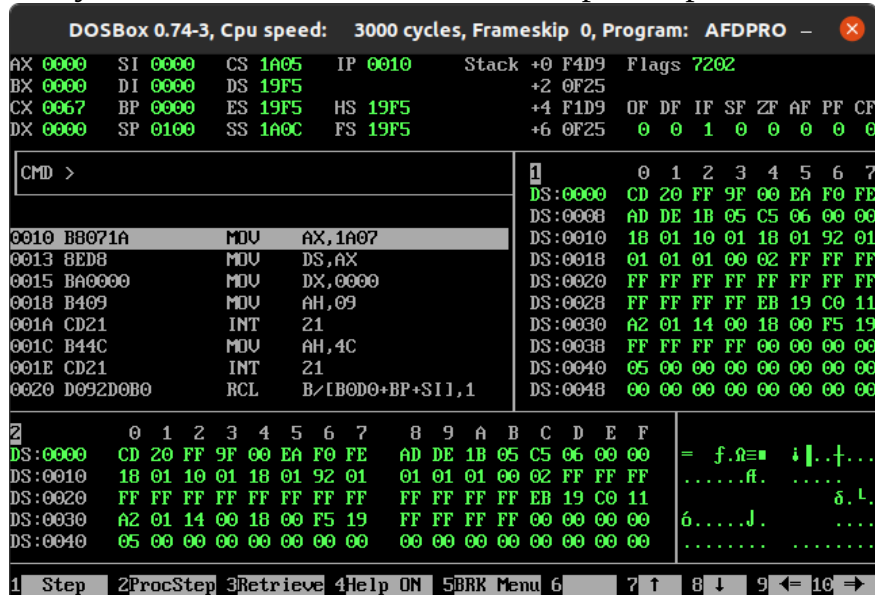


Таблица 1. Протокол hello1

Адрес с Команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0010	Mov ax, @data	B8 07 1A	AX = 0000 IP = 0010	AX = 1A07 IP = 0013
0013	Mov ax, dx	8E D8	DS = 19F5 IP = 0013	DS = 1A07 IP = 0015
0015	Mov dx, OFFSET Greeting	BA 0000	DX = 0000 IP = 0015	DX = 0000 IP = 0018
0018	Mov ah, 09	B4 09	AX = 1A07 IP = 0018	AX = 0907 IP = 001A
001A	Int 21h	CD 21	IP = 001A	IP = 001C
001C	Mov ah, 4ch	B4 4C	AX = 0907 IP = 001C	AX = 4C07 IP = 001E
001E	Int 21h	CD 21	IP = 001E AX = 4C07	IP = 0010 AX = 0000

			DS = 1A07	DS = 19F5
--	--	--	-----------	-----------

## 2. hello2.asm

Рисунок 2 — Начальное состояние регистров hello2

Таблица 2. Протокол hello2

Адрес Команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0005	Push ds	1E	IP = 0005  SP = 0018  Stack: +0 0000 +2 0000 +4 0000 +6 0000	IP = 0006  SP = 0016  Stack: +0 19F5 +2 0000 +4 0000 +6 0000
0006	Sub AX, AX	2B C0	AX = 0000 IP = 0006	AX = 0000 IP = 0008
0008	Push AX	50	IP = 0008  SP = 0016  Stack: +0 19F5	IP = 0009  SP = 0014  Stack: +0 0000

			+2 0000 +4 0000 +6 0000	+2 19F5 +4 0000 +6 0000
0009	Mov AX, DATA	B8 07 1A	AX = 0000 IP = 0009	AX = 1A07 IP = 000C
000C	Mov DS, AX	8E D8	DS = 19F5 IP = 000C	DS = 1A07 IP = 000E
000E	Move DX, OFFSET HELLO	BA 00 00	DX = 0000 IP = 000E	DX = 0000 IP = 0011
0011	Call WriteMsg	E8 EC FF	IP = 0011 SP = 0014 Stack: +0 0000 +2 19F5 +4 0000 +6 0000	IP = 0000 SP = 0012 Stack: +0 0014 +2 0000 +4 19F5 +6 0000
0000	Mov AH, 09	B4 09	AX = 1A07 IP = 0000	AX = 0907 IP = 0002
0002	Int 21h	CD 21	IP = 0002	IP = 0004
0004	Ret	C3	IP = 0004 SP = 0012 Stack: +0 0014 +2 0000 +4 19F5 +6 0000	IP = 0014 SP = 0014 Stack: +0 0000 +2 19F5 +4 0000 +6 0000
0014	Mov DX, OFFSET GREETING	BA 10 00	DX = 0000 IP = 0014	DX = 0010 IP = 0017
0017	Call WriteMsg	E8 E6 FF	IP = 0017 SP = 0014 Stack: +0 0000 +2 19F5 +4 0000	IP = 0000 SP = 0012 Stack: +0 001A +2 0000 +4 19F5

			+6 0000	+6 0000
0000	Mov AH, 09	B4 09	AX = 0907 IP = 0000	AX = 0907 IP = 0002
0002	Int 21h	CD 21	IP = 0002	IP = 0004
0004	Ret	C3	IP = 0004 SP = 0012 Stack: +0 001A +2 0000 +4 19F5 +6 0000	IP = 001A SP = 0014 Stack: +0 0000 +2 19F5 +4 0000 +6 0000
001A	Ret	CB	IP = 001A SP = 0014 CS = 1A0A Stack: +0 0000 +2 19F5 +4 0000 +6 0000	IP = 0000 SP = 0018 CS = 19F5 Stack: +0 0000 +2 0000 +4 0000 +6 0000
0000	Int 20	CD 20	AX = 0907	AX = 0000

**Выводы.**

Получены навыки выполнения, трансляции и отладки программ на языке Ассемблера.

Изучены исходные коды программ `hello1.asm` и `hello2.asm`, составлены протоколы работы программ с помощью отладчика.



## 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
hello1.lst:

```

HELLO1.ASM - ̃)Đ;̃(ĐŸ̃)ĐμĐœĐœĐ°̃  
Đ²Đμ̃(̃)Đž  
̃  
̃)̃ĐμĐ±ĐœĐŸĐ¹  
Đ;̃(ĐŸĐ³̃(Đ°ĐœĐœ̃) Đ»Đ°Đ±.̃(Đ  
°Đ±. N1  
;  
Đ;ĐŸ ĐžĐž̃)̃ĐžĐ;Đ»ĐžĐœĐμ  
"Đ ̃  
̃)Đž̃(ĐμĐ°̃(̃)̃(Đ°  
Đ°ĐŸĐœĐ;̃{̃ ̃(Đμ̃(Đ°"  
;  
\*\*\*\*\*  
\*\*\*\*\*  
;  
Đ Đ°Đ·ĐœĐ°̃ĐμĐœĐžĐμ:  
Đ ̃(ĐŸĐ³̃(Đ°ĐœĐœĐ° ̃(ĐŸ  
̃(ĐœĐž̃(̃)Đμ̃( Đž Đ²̃)Đ²ĐŸĐžĐž̃(  
ĐœĐ° ̃ Đ°̃ Đ°Đ  
œ Đ;̃(ĐžĐ²Đμ̃(̃)̃(Đ²ĐžĐμ  
;  
Đ;ĐŸĐ»̃{Đ·ĐŸĐ²Đ°̃(ĐμĐ»̃ ̃)  
Đ;ĐŸĐ  
œĐŸ̃)̃{̃ ̃(̃)ĐœĐ°̃ĐžĐž  
Đ Đ Đ; "Đ ̃ )Đ²ĐŸĐž̃ ̃)̃  
̃(ĐŸĐ°Đž"  
;  
(ĐœĐŸĐœĐμ̃( 09  
Đ;̃(Đμ̃(̃)Đ²Đ°ĐœĐž

$\mathbb{D}_\mu$  21h),  $\mathbb{D}^\circ \mathbb{D} \ddot{Y} \tilde{N} (\mathbb{D} \ddot{Y} \tilde{N} (\mathbb{D}^\circ \tilde{N} :$   
 $;$  -  $\mathbb{D} \ddot{Y} \mathbb{D} \pm \mathbb{D}_\mu \tilde{N}) \mathbb{D}_\zeta \mathbb{D}_\mu \tilde{N} \gg \mathbb{D} \check{Z} \mathbb{D}^2 \mathbb{D}^\circ \mathbb{D}_\mu \tilde{N} ($   
 $\mathbb{D}^2 \tilde{N}) \mathbb{D}$   
 $^2 \mathbb{D} \ddot{Y} \mathbb{D} \check{Z} \mathbb{D} \alpha \mathbb{D}^\circ \tilde{N} \mathbb{D}^\circ \tilde{N} (\mathbb{D}^\circ \mathbb{D} \alpha \tilde{N}) \tilde{N} (\tilde{N} (\mathbb{D} \ddot{Y} \mathbb{D}^\circ \mathbb{D} \check{Z}$   
 $\tilde{N}) \mathbb{D} \check{Z} \mathbb{D} \mathbb{C} \mathbb{D}^2 \mathbb{D} \ddot{Y} \mathbb{D} \gg$   
 $\mathbb{D} \ddot{Y} \mathbb{D}^2,$   
 $;$   
 $\mathbb{D} \cdot \mathbb{D}^\circ \mathbb{D}^\circ \mathbb{D}^\circ \mathbb{D} \alpha \tilde{N} \gg \mathbb{D} \check{Z} \mathbb{D}^2 \mathbb{D}^\circ \tilde{N} \tilde{N} \mathbb{D}_\mu \mathbb{D}^1 \tilde{N}) \tilde{N}$   
 $\mathbb{D} \cdot \mathbb{D} \alpha \mathbb{D}^\circ \mathbb{D}^\circ \mathbb{D} \ddot{Y} \mathbb{D} \mathbb{C} "\$";$   
 $;$  -  $\tilde{N} (\tilde{N} (\mathbb{D}_\mu \mathbb{D} \pm \tilde{N}) \mathbb{D}_\mu \tilde{N} ($   
 $\mathbb{D} \cdot \mathbb{D}^\circ \mathbb{D} \check{Z} \mathbb{D}^\circ \mathbb{D} \alpha \mathbb{D} \check{Z} \tilde{N}$   
 $\mathbb{D}^2 \tilde{N} (\mathbb{D}_\mu \mathbb{D}^3 \mathbb{D} \check{Z} \tilde{N}) \tilde{N} (\tilde{N} (\mathbb{D}_\mu \text{ ah } \mathbb{D} \alpha \mathbb{D} \ddot{Y} \mathbb{D} \mathbb{C} \mathbb{D}_\mu \tilde{N} (\mathbb{D}^\circ$   
 $\tilde{N} (\tilde{N}) \mathbb{D} \alpha \mathbb{D}^\circ \tilde{N} \ll \mathbb{D}$   
 $\check{Z} \mathbb{D} \check{Z} = 09h,$   
 $;$   $\mathbb{D}^\circ \mathbb{D}^2 \tilde{N} (\mathbb{D}_\mu \mathbb{D}^3 \mathbb{D} \check{Z} \tilde{N}) \tilde{N} (\tilde{N} (\mathbb{D}_\mu \text{ dx} - \tilde{N}$   
 $) \mathbb{D} \mathbb{C} \mathbb{D}_\mu \tilde{N} \gg \mathbb{D}_\mu \mathbb{D} \alpha \mathbb{D} \check{Z} \tilde{N} \mathbb{D}^\circ \mathbb{D} \check{Z} \tilde{N} (\mathbb{D}_\mu \tilde{N}) \mathbb{D}^\circ$   
 $\mathbb{D}^2 \tilde{N}) \mathbb{D}^2 \mathbb{D} \ddot{Y} \mathbb{D} \check{Z} \mathbb{D} \check{Z} \mathbb{D} \mathbb{C} \mathbb{D} \ddot{Y}$   
 $\mathbb{D}^1$   
 $;$   $\tilde{N} \tilde{N} (\tilde{N} (\mathbb{D} \ddot{Y} \mathbb{D}^\circ \mathbb{D} \check{Z};$   
 $;$  -  $\mathbb{D} \check{Z} \tilde{N}) \mathbb{D}_\zeta \mathbb{D} \ddot{Y} \mathbb{D} \gg \tilde{N} (\mathbb{D} \cdot \tilde{N}) \mathbb{D}_\mu \tilde{N} ($   
 $\tilde{N} (\mathbb{D}_\mu \mathbb{D}^3 \mathbb{D} \check{Z} \tilde{N}$   
 $) \tilde{N} (\tilde{N} (\text{ ax } \mathbb{D} \check{Z} \mathbb{D} \alpha \mathbb{D}_\mu \tilde{N}) \mathbb{D} \ddot{Y} \tilde{N}) \tilde{N} (\mathbb{D}^\circ \mathbb{D} \alpha \tilde{N} \mathbb{D}_\mu \tilde{N} ($   
 $\mathbb{D}_\mu \mathbb{D}^3 \mathbb{D} \ddot{Y}$   
 $;$   $\tilde{N}) \mathbb{D} \ddot{Y} \mathbb{D} \check{Z} \mathbb{D}_\mu \tilde{N} (\mathbb{D} \ll \mathbb{D} \check{Z} \mathbb{D} \mathbb{C} \mathbb{D} \ddot{Y} \mathbb{D}_\mu.$   
 $;$

\*\*\*\*\*

\*\*\*\*\*

DOSSEG

```

;                                     Đ Đ°ĐŽĐ°ĐœĐŽĐµ
Ñ)ĐµĐ³ĐƎĐµĐœÑ(ĐŸĐ² ĐġĐŸĐŽ Đ Đ
Đi
.MODEL SMALL
; Đ ĐŸĐŽĐµĐ»Ñ{ ĐġĐ°ĐƎÑ Ñ(ĐŽ-
SMALL(Đ Đ°Đ»Đ°Ñ )
.STACK 100h
; Đ Ñ (Đ²ĐµÑ)Ñ (Đž ĐġĐŸĐŽ ĐiÑ (ĐµĐ°
256 Đ±Đ°Đ¹Ñ(
.DATA
; Đ Đ°Ñ»Đ°Đ»ĐŸ Ñ)ĐµĐ³ĐƎĐµĐœÑ(Đ°
ĐŽĐ°ĐœĐœÑ\Ñ)
0000 Greeting LABEL BYTE
; ĐƒĐµĐ°Ñ)Ñ( ĐġÑ(ĐžĐ²ĐµÑ(Ñ)Ñ(Đ²ĐžÑ
0000 D0 92 D0 B0 D1 81 DB 'Đ Đ°Ñ)
ĐġÑ(ĐžĐ²ĐµÑ(Ñ)Ñ(Đ²Ñ)ĐµÑ( Ñ)Ñ(.Đ³Ñ
(.7303 - Đ Đ²Đ°ĐœĐŸĐ² Đ .Đ .',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

```

20 D0 98 D0 B2 D0  
 B0 D0 BD D0 BE D0  
 B2 20 D0 98 2E D0  
 98 2E 0D 0A 24

```

                                .CODE                ; Ð Ð°Ñ
                                »Ð°Ð»ÐŸ Ñ)ÐµÐ³ÐŒÐµÐœÑ(Ð° Ð°ÐŸÐŸÐŸÐ°
0000 B8 ---- R                mov ax, @data          ; Ð Ð°
                                Ð³Ñ(Ñ)Ð·Ð°Ð° Ð² DS Ð°ÐŸÑÑ(ÐµÑ)Ð°
ÐœÐ°Ð°Ñ»Ð°Ð»Ð°
0003 8E D8                    mov ds, ax              ; Ñ)Ðµ
                                Ð³ÐŒÐµÐœÑ(Ð° ÐŸÐ°ÐœÐœÑÑ)
0005 BA 0000 R                mov dx, OFFSET Greeting ; Ð Ð°
                                Ð³Ñ(Ñ)Ð·Ð°Ð° Ð² dx Ñ)ÐŒÐµÑ)ÐµÐœÐŸÑ
                                ; Ð°ÐŸÑ
                                (ÐµÑ)Ð°                Ñ(ÐµÐ°Ñ)Ñ(Ð°
Ð¿Ñ(ÐŸÐ²ÐµÑÑÑÑ(Ð²ÐŸÑ
0008                          DisplayGreeting:
0008 B4 09                    mov ah, 9                ; # Ñ(
                                Ñ)ÐœÐ°Ñ«ÐŸÐŸ Ð Ð Ð¿ Ð¿ÐµÑ»Ð°Ñ(ÐŸ
ÑÑÑÑ(ÐŸÐ°ÐŸ
000A CD 21                    int 21h                  ; Ð²Ñ)Ð
                                ²ÐŸÐŸ ÐœÐ°Ð°                ÑÐ°Ñ(Ð°Ðœ
Ð¿Ñ(ÐŸÐ²ÐµÑÑÑÑ(Ð²ÐŸÑ
000C B4 4C                    mov ah, 4ch              ; # Ñ(

```



$\tilde{N})D_{\alpha}D^{\circ}\tilde{N}\langle D\check{z}D\check{z}$   
 $D \quad D \quad D_i \quad D \cdot D^{\circ}D^2D_{\mu}\tilde{N}(\tilde{N}\langle D_{\mu}D_{\alpha}D\check{z}\tilde{N} \quad D_{\check{z}}\tilde{N}(D\check{Y}$   
 $D^3\tilde{N}(D^{\circ}D\mathbb{C}D\mathbb{C}\tilde{N})$   
 $000E \quad CD \quad 21 \quad \quad \quad \text{int } 21h \quad \quad \quad ; D \cdot D^{\circ}D$   
 $^2D_{\mu}\tilde{N}(\tilde{N}\langle D_{\mu}D_{\alpha}D\check{z}D_{\mu} \quad D_{\check{z}}\tilde{N}(D\check{Y}D^3\tilde{N}(D^{\circ}D\mathbb{C}D\mathbb{C}\tilde{N})$   
 $D\check{z} \quad D^2\tilde{N})\tilde{N})D\check{Y}D$   
 $\check{Z} \quad D^2 \quad D \quad D \quad D_i$   
 $END$

## 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

47992 + 459268 Bytes symbol space free

0 Warning Errors

0 Severe Errors

**hello2.lst:**

```

; HELLO2 - DŁŃ»DµD±DœD°Ń
DŁŃ(DŸD³Ń(D°DœDœD° N2
D»D°D±.Ń(D°D±.#1 DŁDŸ
DŽDžŃ)Ń«DžDŁD»DžDœDµ "D
Ń(Ń)DžŃ(DµD°Ń(Ń)Ń(D°
D°DŸDœDŁŃ{Ń Ń(DµŃ(D°"
; D Ń(DŸD³Ń(D°DœDœD°
DžŃ)DŁDŸD»Ń{D·Ń)D
µŃ( DŁŃ(DŸŃ«DµDžŃ)Ń(Ń) DžD»Ń
DŁDµŃ»D°Ń(Dž Ń)Ń(
Ń(DŸD°Dž
;
; D¢D D DŁD¢
D D D D D D D D«
= 0024 EOFLine EQU '$' ;
D DŁŃ DµDžDµD»DµDœD
žDµ Ń)DžDœD²DŸD»Ń{DœDŸD¹
D°DŸDœŃ)Ń(D°DœŃ(Ń)
; "D DŸDœDµŃ«Ń)Ń
(Ń(DŸD°Dž"
; DŁŃ(DµD° DŁŃ(DŸD³Ń(D°DœDœŃ)
ASSUME CS:CODE, SS:AStack

```

```

0000          AStack  SEGMENT STACK
0000 000C[                               DW  12  DUP('!')          ;
Đ  Ń (Đ²ĐŸĐŽĐžŃ (Ń )Ń
                                12 Ń )Đ»ĐŸĐ² ĐġĐ°ĐĈŃ  Ń (Đž
0021
                                ]

0018          AStack  ENDS

                                ; Đ  Đ°ĐœĐœŃ)Đµ ĐġŃ(ĐŸĐ³Ń(Đ°ĐĈĐĈŃ)

0000          DATA   SEGMENT

                                ;                                Đ  ĐžŃ(ĐµĐ°Ń(ĐžĐ²Ń)
ĐŸĐġĐžŃ)Đ°ĐœĐžŃ  ĐŽĐ°ĐœĐœ
                                Ń)Ń)

0000 48 65 6C 6C 6F 20      HELLO      DB  'Hello  Worlds!', 0AH,
0DH,EOFLine
0000 57 6F 72 6C 64 73
0000 21 0A 0D 24
0010 53 74 75 64 65 6E      GREETING DB  'Student from 4350 - $'
0010 74 20 66 72 6F 6D
0010 20 34 33 35 30 20
0010 2D 20 24
0025          DATA   ENDS

                                ; Đ  ĐŸĐž ĐġŃ(ĐŸĐ³Ń(Đ°ĐĈĐĈŃ)

```

0000	CODE SEGMENT
	;   Đ   Ń(ĐŸŃ«ĐµĐŽŃ)Ń(Đ°   ĐġĐµŃ»Đ°Ń(Đž
Ń)Ń(Ń(ĐŸĐ°Đž	
0000	WriteMsg PROC NEAR
0000 B4 09	mov AH,9
0002 CD 21	int 21h ;   Đ   Ń)Đ·ĐŸĐ²
Ń(Ń)ĐœĐ°Ń«ĐžĐ	
	ž DOS ĐġĐŸ ĐġŃ(ĐµŃ(Ń)Đ²Đ°ĐœĐžŃ
0004 C3	ret
0005	WriteMsg ENDP
	;                                   Đ   ĐŸĐ»ĐŸĐ²ĐœĐ°Ń
ĐġŃ(ĐŸŃ«ĐµĐŽŃ)Ń(Đ°	

```

0005          Main    PROC FAR
0005 1E          push DS      ;\ Đ;ĐŸŃ)Ń(Đ°ĐœĐµĐœĐž
Đµ Đ°ĐžŃ(ĐµŃ)Đ° ĐœĐ°Ń»Đ°Đ»Đ° PSP Đ²
Ń)Ń(ĐµĐ°Đµ
0006 2B C0          sub     AX,AX      ; > ĐžĐ»Ń
Đ;ĐŸŃ)Đ»ĐµĐ
                                žŃ)Ń Ń)ĐµĐ³ĐŸ
Đ²ĐŸŃ)Ń)Ń(Đ°ĐœĐŸĐ²Đ»ĐµĐœĐžŃ Đ;ĐŸ
0008 50          push AX      ;/ Đ°ĐŸĐœĐ°ĐœĐžĐµ ret
,                                Đ·Đ°Đ²ĐµŃ(Ń(Đ°Ń Ń)ĐµĐ¹
Đ;Ń(ĐŸŃ«ĐµĐžŃ)Ń(Ń).
0009 B8 ---- R          mov     AX,DATA      ; Đ Đ°Đ³Ń(Ń
)Đ·Đ°Đ° Ń)ĐµĐ³ĐœĐµĐœŃ(ĐœĐŸĐ³ĐŸ
000C 8E D8          mov     DS,AX      ; Ń(ĐµĐ³ĐžŃ
)Ń(Ń(Đ° ĐžĐ°ĐœĐœŃ)Ń).
000E BA 0000 R          mov     DX, OFFSET HELLO      ;
Đ Ń)Đ²ĐŸĐ
                                ž ĐœĐ° Ń)Đ°Ń(Đ°Đœ Đ;ĐµŃ(Đ²ĐŸĐ¹
0011 E8 0000 R          call WriteMsg      ; Ń)Ń(Ń(ĐŸĐ
°Đž Đ;Ń(ĐžĐ²ĐµŃ(Ń)Ń(Đ²ĐžŃ .
0014 BA 0010 R          mov     DX, OFFSET GREETING      ;
Đ Ń)Đ²ĐŸĐ
                                ž ĐœĐ° Ń)Đ°Ń(Đ°Đœ Đ²Ń(ĐŸŃ(ĐŸĐ¹
0017 E8 0000 R          call WriteMsg      ; Ń)Ń(Ń(ĐŸĐ
°Đž Đ;Ń(ĐžĐ²ĐµŃ(Ń)Ń(Đ²ĐžŃ .

```

```

001A CB                ret                ; D  Ń)Ń)ĐŸĐ
                        Ž Đ² DOS Đ;ĐŸ Đ°ĐŸĐŒĐ°ĐœĐŽĐµ,
                                ; ĐœĐ°Ń)ĐŸĐ
                        ŽŃ  Ń)ĐµĐ¹Ń)Ń                Đ²                1-ĐŸĐŒ
Ń)Đ»ĐŸĐ²Đµ PSP.

001B                Main    ENDP

001B                CODE    ENDS

                        END Main

```



## 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

47984 + 459273 Bytes symbol space free

0 Warning Errors

0 Severe Errors