

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. 9383

Пономаренко С. А.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2020

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
```

Тексты исходных файлов программ hello1 и hello2.

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.9383 - Пономаренко С.А.',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

```

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 'Hello Worlds!', 0AH, 0DH,EOFLine
GREETING DB 'Student from 9383 - Ponomarenko Sofya $'
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

Ddvs.

hello1.lst

; 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 ДО
С "Вывод строки"

; (номер 09 прерывание 21h), которая
я:

; - 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
0000    Greeting LABEL BYTE
        ; Текст приветствия
0000    82 A0 E1 20 AF E0    DB 'Вас приветствует ст.гр.9383 - Пономаренк
        о С.А.',13,10,'$'

        A8 A2 A5 E2 E1 E2
        A2 E3 A5 E2 20 E1
        E2 2E A3 E0 2E 39
        33 38 33 20 2D 20
        8F AE AD AE AC A0
        E0 A5 AD AA AE 20
        91 2E 80 2E 0D 0A
        24

```

```

        .CODE                                ; Начало сегмента кода
0000    B8 ---- R    mov ax, @data            ; Загр
        узка в DS адреса начала
0003    8E D8                mov ds, ax        ; сегм
        ента данных
0005    BA 0000 R    mov dx, OFFSET Greeting    ; Загр

```

```

#Microsoft (R) Macro Assembler Version 5.10    10/15/20 00:10:4

```

Page 1-2

```

        узка в dx смещения
        ; адрес
        а текста приветствия
0008    DisplayGreeting:
0008    B4 09                mov ah, 9        ; # фу

```

```

нкции ДОС печати строки
000A CD 21                int 21h                ; вывод
                        на экран приветствия
000C B4 4C                mov ah, 4ch            ; # фу
нкции ДОС завершения программы
000E CD 21                int 21h                ; завер
                        шение программы и выход в ДОС
                        END

#Microsoft (R) Macro Assembler Version 5.10      10/15/20 00:10:4
                        Symbols-1

```

Segments and Groups:

N a m e	Length	Align	Combine	Class
DGROUP	GROUP			
_DATA	0031	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 + 461313 Bytes symbol space free

0 Warning Errors

0 Severe Errors

hello2.lst

#Microsoft (R) Macro Assembler Version 5.10

10/15/20 01:31:2

Page 1-1

; HELLO2 - Учебная программа N2 лаб.раб.#1 по
дисциплине "Архитектура компьютера"

; Программа использует процедуру для п
ечати строки

;

; ТЕКСТ ПРОГРАММЫ

= 0024

EOFLine EQU '\$'

; Определение символъ

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 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 9383 - Ponomarenko S
ofya \$'

74 20 66 72 6F 6D
20 39 33 38 33 20
2D 20 50 6F 6E 6F
6D 61 72 65 6E 6B
6F 20 53 6F 66 79
61 20 24

0037 DATA ENDS

; Код программы

0000 CODE SEGMENT

ASSUME CS:Code DS:DATA SS:AStack

hello2.asm(28): warning A4001: Extra characters on line

; Процедура печати строки

0000 WriteMsg PROC NEAR

0000 B4 09 mov AH,9

0002 CD 21 int 21h ; Вызов функции DOS по прерыванию

0004 C3 ret

0005 WriteMsg ENDP

; Головная процедура

0005 Main PROC FAR

#Microsoft (R) Macro Assembler Version 5.10

10/15/20 01:31:2

Page 1-2

0005 1E push DS ;\ Сохранение адреса начала PSP в стеке

0006 2B C0 sub AX,AX ; > для последующего в останавления по

0008 50 push AX ;/ команде ret, завершающей 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 10/15/20 01:31:2
Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK	0018	PARA		STACK
CODE	001B	PARA		NONE
DATA	0037	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	

51 Source Lines

51 Total Lines

13 Symbols

48002 + 461305 Bytes symbol space free

1 Warning Errors

0 Severe Errors

Протокол работы на компьютере.

1. Просмотрели программу hello1.asm.
2. Разобрались в структуре и реализации каждого сегмента программы. Строку-приветствие преобразовали в соответствии со своими личными данными.

3. Загрузили файл hello1.asm.
4. Протранслировали программу с помощью строки `masm hello1.asm`.
5. Скомпоновали загрузочный модуль с помощью строки `link hello1.obj`.
6. Выполнили программу в автоматическом режиме путем набора строки `hello1.exe` и убедились в корректности ее работы.
7. Запустили выполнение программы под управлением отладчика с помощью команды `afdp Hello1.exe`.
8. Повторили те же действия с файлом hello2.asm.

Протокол пошагового исполнения программы 1.

Адрес	Символический	16-ричный	Содержимое регистров и ячеек памяти	после
Команды	код команды	код команды	до выполнения	выполнения
10	MOV AX, 1A07	B8071A	(AX) = 0000 (IP) = 0010	(AX) = 1A07 (IP) = 0013
13	MOV DS, AX	8ED8	(DS) = 19F5 (IP) = 0013	(DS) = 1A07 (IP) = 0015
15	MOV DX, 0000	BA0000	(IP) = 0015	(IP) = 0018
18	MOV AH, 09	B409	(AX) = 1A07 (IP) = 0018	(AX) = 0907 (IP) = 001A
001A	INT 21	CD21	(IP) = 001A	(IP) = 001C
001C	MOV AH, 4C	B44C	(AX) = 0907 (IP) = 001C	(AX) = 4C07 (IP) = 001E
001E	INT 21	CD21	(AX) = 4C07 (CX) = 0051 (IP) = 001E	(AX) = 0000 (CX) = 0000 (IP) = 0010

Протокол пошагового исполнения программы 2.

5	PUSH DS	1E	(SP) = 0018 (IP) = 0005	(SP) = 0016 (IP) = 0006
6	SUB AX, AX	2BC0	(IP) = 0006	(IP) = 0008
8	PUSH AX	50	(SP) = 0016 (IP) = 0008	(SP) = 0014 (IP) = 0009
9	MOV AX, 1A07	B8071A	(IP) = 0009 (AX) = 0000	(IP) = 000C (AX) = 1A07
000C	MOV DS, AX	8ED8	(IP) = 000C (DS) = 19F5	(IP) = 000E (DS) = 1A07
000E	MOV DX, 0000	BA0000	(IP) = 000E	(IP) = 0011
11	CALL 0000	E8ECFF	(IP) = 0011 (SP) = 0014	(IP) = 0000 (SP) = 0012
0	MOV AH, 09	B409	(IP) = 0000 (AX) = 1A07	(IP) = 0002 (AX) = 0907
2	INT 21	CD21	(IP) = 0002	(IP) = 0004

4	RET	C3	(SP) = 0012 (IP) = 0004	(SP) = 0014 (IP) = 0014
14	MOV DX, 0010	BA10000	(IP) = 0014	(IP) = 0017
17	CALL 0000	E8ECFF	(SP) = 0014 (IP) = 0017	(SP) = 0012 (IP) = 0000
0	MOV AH, 09	B409	(IP) = 0000	(IP) = 0002
2	INT 21	CD21	(IP) = 0002	(IP) = 0004
4	RET	C3	(SP) = 0012 (IP) = 0004	(SP) = 0014 (IP) = 001A
001A	RET Far	CB	(SP) = 0014 (CS) = 1A0B (IP) = 001A	(SP) = 0018 (CS) = 19F5 (IP) = 0000
0	INT 20	CD20	(AX) = 0907 (CX) = 007B (DX) = 0010 (CS) = 19F5 (DS) = 1A07 (IP) = 0000	(AX) = 0000 (CX) = 0000 (DX) = 0000 (CS) = 1A0B (DS) = 19F5 (IP) = 0005

Вывод.

Ознакомились с трансляцией, отладкой на языке Ассемблера. С помощью dosbox создали файлы .obj, .lst и .exe. Выполнили программу, ошибок не было обнаружено. Изучили изменение регистров и памяти при пошаговом выполнении программы, занесли результаты в протокол. Опробовали некоторые команды в dosbox.