

**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. 1303

Попандопуло А. Г.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

## **Цель работы.**

Изучение работы простейших программ на языке Ассемблера, практика в работе с отладчиком.

## **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` 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`.

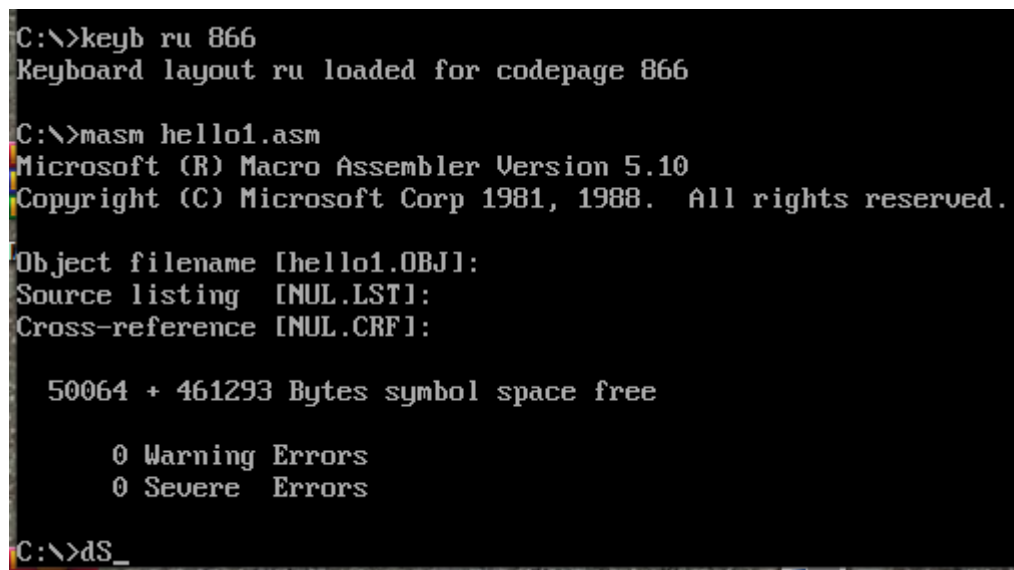
6. Выполнить программу в автоматическом режиме путем набора строки `> hello1.exe` убедиться в корректности ее работы и зафиксировать результат выполнения в протоколе.
7. Запустить выполнение программы под управлением отладчика с помощью команды `> afd hello1.exe`

## 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.

Ход работы.

Трансляция `hello1.asm` с созданием объектного файла `hello1.obj` – рис.1



```
C:\>keyb ru 866
Keyboard layout ru loaded for codepage 866

C:\>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 + 461293 Bytes symbol space free

    0 Warning Errors
    0 Severe Errors

C:\>dS_
```

Рис.1

Скомпоновка загрузочного модуля с созданием исполняемого файла `hello1.exe` и его запуск: (рис.2)

```

C:\>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]:

C:\>hello1
Вас приветствует ст.гр.1303 – Попандопуло А.Г.

C:\>S_

```

Рис. 2

Аналогичные действия с файлом hello2.asm представлены на рис.3 и рис.4

```

C:\>masm hello2.asm
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988.  All rights reserved.

Object filename [hello2.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

    50072 + 461285 Bytes symbol space free

    0 Warning Errors
    0 Severe  Errors

C:\>S_

```

Рис. 3

```

C:\>link hello2.obj

Microsoft (R) Overlay Linker  Version 3.64
Copyright (C) Microsoft Corp 1983-1988.  All rights reserved.

Run File [HELLO2.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:

C:\>hello2
Hello Worlds!
Student from 1303 – Popandopulo Aleksandr
C:\>SS

```

Рис. 4

Посредством отладчика, программы были выполнены пошагово, в таблицах представлены соответствующие протоколы.

Файл hello1.asm:

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0000	mov ax,1a07	b8 07 1a	(ax) = 0000 (ip) = 0010	(ax) = 1a07 (ip) = 0013
0003	mov ds,ax	8e d8	(ds) = 19f5 (ip) = 0013	(ds) = 1a07 (ip) = 0015
0005	mov dx,0000	ba 00 00	(ip) = 0015 (dx) = 0000	(ip) = 0018 (dx) = 0000
0008	mov ah,09	b4 09	(ax) = 1a07 (ip) = 0018	(ax) = 0907 (ip) = 001a
000a	int 21	cd 21	(ip) = 001a	(ip) = 001c
000c	mov ah,4c	b4 4c	(ax) = 0907 (ip) = 001c	(ax) = 4c07 (ip) = 001e
000e	int 21	cd 21	(ax) = 4c07 (ds) = 1a07 (cx) = 006b (ip) = 001e	(ax) = 0000 (cx) = 0000 (ds) = 19f5 (ip) = 0010

Таблица 1

Файл hello2.asm:

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0005	push ds	1e	stack(+0) = 0000 (ip) = 0005 (sp) = 0018	stack(+0) = 19f5 (ip) = 0006 (sp) = 0016
0006	sub ax,ax	2b c0	(ip) = 0006 (ax) = 0000	(ip) = 0008 (ax) = 0000
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	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	mov dx,0000	ba 00 00	(ip) = 000e (dx) = 0000	(ip) = 0011 (dx) = 0000
0011	call 0000	e8 ec ff	(ax) = 1a07 (ip) = 0011 (sp) = 0014 stack(+0) = 0000 stack(+2) = 19f5 stack(+4) = 0000	(ax) = 0907 (ip) = 0000 (sp) = 0012 stack(+0) = 0014 stack(+2) = 0000 stack(+4) = 19f5
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 stack(+2) = 0000 stack(+4) = 19f5	(ip) = 0014 (sp) = 0014 stack(+0) = 0000 stack(+2) = 19f5 stack(+4) = 0000
0014	mov dx,0010	ba 10 00	(ip) = 0014 (dx) = 0000	(ip) = 0017 (dx) = 0010
0017	call 0000	e8 e6 ff	(ip) = 0017 (dx) = 0010 (sp) = 0014 stack(+0) = 0000 stack(+2) = 19f5 stack(+4) = 0000	(ip) = 0000 (dx) = 0010 (sp) = 0012 stack(+0) = 001a stack(+2) = 0000 stack(+4) = 19f5
0000	mov ah, 9	b4 09	(ip) = 0000	(ip) = 0002
0002	int 21	cd 21	(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 (cs) = 1a0a (sp) = 0014 stack(+0) = 0000 stack(+2) = 19f5	(ip) = 0000 (cs) = 19f5 (sp) = 0018 stack(+0) = 0000 stack(+2) = 0000
0000	int 20	cd 20	(ax) = 0907 (dx) = 0010 (cs)=19f5 (ds) = 1a07 (ip) = 0000 (cx) = 006b	(ax) = 0000 (dx) = 0000 (cs)=1a0a (ds) = 19f5 (ip) = 0005 (cx) = 0000

Таблица 2

Вывод: в ходе лабораторной работы была изучена структура простейших программ языке Ассемблер. Выполнены трансляция, линковка и выполнение программы, а также ее изучение в отладчике.

# 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.1303 - Попандопуло А.Г.',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 1303 - Popandopulo Aleksandr$'
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

```