

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 Ассембле-
ра.

Студент гр. 9382

Павлов Р.В.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Изучить основные принципы трансляции, отладки и выполнения программ на языке Ассемблера.

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

> `afid 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.

Результаты прогона программы под управлением отладчика должны быть представлены в виде, показанном на примере одной команды в табл.1, и подписаны преподавателем.

Табл.1

Адрес Команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			до выполнения .	После выполнения
0003	Mov DS, AX	8E D8	(AX) = 2D87 (DS) = 2D75 (IP) = 0003	(AX) = 2D87 (DS) = 2D87 (IP) = 0005

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.

Часть 1.

1. Просмотрена программа hello1.asm.
2. Разобраны сегменты упрощённой записи и директивы, строка изменена на « 'Вас приветствует ст.гр. 9382 - Павлов Р.В.',13,10,'\$' ».
3. Выполнен вход в MS DOS, директория с программой смонтирована как диск D, загружена кодовая таблица crr866.
4. Выполнена трансляция с помощью команды
`> masm hello1.asm ,`
созданы объектный файл (hello1.obj) и листинг (hello1.lst).
5. Скомпонован загрузочный модуль из hello1.obj командой

> link hello1.obj,
создана карта памяти (hello1.map) и исполняемый файл (hello1.exe).

5. Программа выполнена в автоматическом режиме командой

> hello1.exe

и завершена успешно. Вывод программы:

Вас приветствует ст.гр. 9382 - Павлов Р.В.

6. Запущен отладчик afdpro командой

> afdpro hello1.exe.

Начальные значения регистров CS, DS, ES и SS представлены ниже:

CS	DS	ES	SS
1A05	19F5	19F5	1A0A

А также изначально:

AX	DX	IP
0000	0000	0010

Изменение значений регистров в ходе работы программы:

Таблица 1.

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 (IP) = 0010	(AX) = 1A07 (IP) = 0013
0013	MOV DS, AX	8ED8	(AX) = 1A07 (DS) = 19F5 (IP) = 0013	(AX) = 1A07 (DS) = 1A07 (IP) = 0015
0015	MOV DX, 0000	BA0000	(DX) = 0000 (IP) = 0015	(DX) = 0000 (IP) = 0018
0018	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 (DS) = 1A07 (IP) = 001E	(AX) = 0000 (DS) = 19F5 (IP) = 0010
------	--------	------	---	---

Часть 2.

1. Просмотрена программа hello2.asm.

2. Разобраны сегменты расширенной записи, процедуры и директивы, изменены строки:

HELLO >>> « 'Привет, мир!', 0AH, 0DH,EOFLine »

GREETING >>> « 'Студент группы 9382 - Роман Павлов\$' »

3. Выполнена трансляция с помощью команды

> masm hello2.asm ,

созданы объектный файл (hello2.obj) и листинг (hello2.lst). Обнаружено предупреждение, связанное с добавлением транслятором символа перевода строки после выделения памяти для стека (Extra characters on line).

4. Скомпонован загрузочный модуль из hello2.obj командой

> link hello2.obj,

создана карта памяти (hello2.map) и исполняемый файл (hello2.exe).

5. Программа выполнена в автоматическом режиме командой

> hello2.exe

и завершена успешно. Вывод программы:

Привет, мир!

Студент группы 9382 - Роман Павлов

6. Запущен отладчик afdpro командой

> afdpro hello2.exe.

Начальные значения регистров CS, DS, ES и SS представлены ниже:

CS	DS	ES	SS
1A0B	19F5	19F5	1A05

А также изначально:

AX	DX	IP	SP
0000	0000	0005	0018

Изменение значений регистров в ходе работы программы:

Таблица 2.

Адрес Команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0005	PUSH DS	1E	(DS) = 19F5 (SP) = 0018 (IP) = 0005 Stack +0 0000 +2 0000 +4 0000 +6 0000	(DS) = 19F5 (SP)=0016 (IP) = 0006 Stack +0 19F5 +2 0000 +4 0000 +6 0000
0006	SUB AX, AX	2BC0	(AX) = 0000 (IP) = 0006	(AX) = 0000 (IP) = 0008
0008	PUSH AX	50	(AX) = 0000 (SP)=0016 (IP) = 0008 Stack +0 19F5 +2 0000 +4 0000 +6 0000	(AX) = 0000 (SP) = 0014 (IP) = 0009 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0009	MOV AX, 1A07	B8071A	(AX) = 0000 (IP) = 0009	(AX) = 1A07 (IP) = 000C
000C	MOV DS, AX	8ED8	(AX) = 1A07 (DS) = 19F5 (IP) = 000C	(AX) = 1A07 (DS) = 1A07 (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 19F5 +4 0000 +6 0000	(SP) = 0012 (IP) = 0000 Stack +0 0014 +2 0000 +4 19F5 +6 0000
0000	MOV AH, 09	B409	(AX) = 1A07 (IP) = 0000	(AX) = 0907 (IP) = 0002
0002	INT 21	CD21	(IP) = 0002	(IP) = 0004

0004	RET	C3	(SP) = 0012 (IP) = 0004 Stack +0 0014 +2 0000 +4 19F5 +6 0000	(SP) = 0014 (IP) = 0014 Stack +0 0000 +2 119C +4 0000 +6 0000
0014	MOV DX, 000A	BA0A00	(DX) = 0000 (IP) = 0014	(DX) = 000A (IP) = 0017
0017	CALL 0000	E8E6FF	(SP)=0014 (IP) = 0017 Stack +0 0000 +2 19F5 +4 0000 +6 0000	(SP) = 0012 (IP) = 0000 Stack +0 001A +2 0000 +4 119C +6 0000
0000	MOV AH, 09	B409	(AX) = 0907 (IP) = 0000	(AX) = 0907 (IP) = 0002
0002	INT 21	CD21	(IP) = 0002	(IP) = 0004
0004	RET	C3	(SP) = 0012 (IP) = 0004 Stack +0 001A +2 0000 +4 19F5 +6 0000	(SP) = 0014 (IP) = 001A Stack +0 0000 +2 19F5 +4 0000 +6 0000
001A	RET Far	CB	(SP) = 0014 (IP) = 001A Stack +0 0000 +2 19F5 +4 0000 +6 0000	(SP) = 0018 (IP) = 0000 Stack +0 0000 +2 0000 +4 0000 +6 0000
0000	INT 20	CD20	(AX) = 0907 (DS) = 1A07 (IP) = 0000	(AX) = 0000 (DS) = 19F5 (IP) = 0005

Сравнение

Различия в начальных адресах сегментов связаны с тем, что стек занимает разное количество памяти в первом и втором случаях (для сегмента SS), а также сокращённая запись в большей степени автоматизирует процесс связывания сегментов с регистрами, поэтому в первом случае CS и IP имеют большее значение, чем во втором.

Выводы.

В результате выполнения данной лабораторной работы были освоены процедуры трансляции и компоновки программы на языке ассемблера, про-

ведены наблюдения за её выполнением в автоматическом и пошаговом режиме.

ПРИЛОЖЕНИЕ А. ИСХОДНЫЙ КОД

- имя 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. 9382 - Павлов Р.В.',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
```

- имя 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 'Привет, мир!', 0AH, 0DH, EOFLine
GREETING  DB 'Студент группы 9382 - Роман Павлов$'
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

```

ПРИЛОЖЕНИЕ Б. ТЕКСТ ЛИСТИНГОВ

- имя файла: hello1.lst

#Microsoft (R) Macro Assembler Version 5.10

10/13/20 20:38:5

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. 9382 - Павлов Р.
        В.',13,10,'$'
        A8 A2 A5 E2 E1 E2
        A2 E3 A5 E2 20 E1
        E2 2E A3 E0 2E 20
        39 33 38 32 20 2D
        20 8F A0 A2 AB AE
        A2 20 90 2E 82 2E
        0D 0A 24

        .CODE                               ; Начал
о 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
узка в dx смещения
```

```

; адрес
а текста приветствия
0008      DisplayGreeting:
0008  B4 09      mov  ah, 9                ; # фy
           нкции ДОС печати строки
000A  CD 21      int  21h                ; вывод
           на экран приветствия
000C  B4 4C      mov  ah, 4ch            ; # фy
           нкции ДОС завершения программы
000E  CD 21      int  21h                ; завер
           шение программы и выход в ДОС
           END

```

Segments and Groups:

N a m e	Length	Align	Combine Class
DGROUP	GROUP		
_DATA	002D	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

- имя файла: hello2.lst

```

        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 (0)
        0000 ]

0018 AStack ENDS; Данные программы
0000 DATA SEGMENT ; Директивы описания данных
0000 8F E0 A8 A2 A5 E2 HELLO DB 'Привет, мир!', 0AH, 0DH, EOFLine
        2C 20 AC A8 E0 21
        0A 0D 24
000F 91 E2 E3 A4 A5 AD GREETING DB 'Студент группы 9382 - Роман Павло
        в$'
        E2 20 A3 E0 E3 AF
        AF EB 20 39 33 38
        32 20 2D 20 90 AE
        AC A0 AD 20 8F A0
        A2 AB AE A2 24
0032 DATA ENDS

        ; Код программы

0000 CODE SEGMENT
        ASSUME CS:Code DS:DATA SS:AStack ; П
        роцедура печати строки
hello2.asm(21): warning A4001: Extra characters on line
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; 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

#Microsoft (R) Macro Assembler Version 5.10 10/13/20 21:22:0
Page 1-2

        экран первой
0011 E8 0000 R call WriteMsg ; строки пр
        иветствия.
0014 BA 000F 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,
                                ; находящейся
                                в 1-ом слове PSP.
001B Main              ENDP
001B CODE              ENDS
                                END Main

```

#Microsoft (R) Macro Assembler Version 5.10

10/13/20 21:22:0
Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine Class
ASTACK	0018	PARA	STACK
CODE	001B	PARA	NONE
DATA	0032	PARA	NONE

Symbols:

N a m e	Type	Value	Attr
EOFLINE	NUMBER	0024	
GREETING	L BYTE	000F	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	

41 Source Lines
41 Total Lines
13 Symbols

48002 + 461305 Bytes symbol space free

1 Warning Errors
0 Severe Errors