

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. 1303

Хулап О.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Разобраться в структуре и реализации каждого сегмента программы, протранслировать и выполнить программу.

Текст задания

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.

Результаты прогона программы под управлением отладчика должны быть представлены в

виде, показанном на примере одной команды в табл.1, и подписаны преподавателем.

Выполнение задания

1. Была просмотрена программа hello1.asm, которая формирует и выводит на экран приветствие пользователя с помощью функции ОС MSDOS, вызываемой через прерывание с номером 21H.

2. Разобрана структура и реализация каждого сегмента программы.
Строку-приветствие преобразована в соответствии со своими личными данными.
3. Загружен файл hello1.asm из каталога Задания в каталог Masm.
4. Протранслирована программа с помощью строки > masm hello1.asm с созданием объектного файла и файла диагностических сообщений (файла листинга).
5. Скомпонован загрузочный модуль с помощью строки > link hello1.obj с созданием карты памяти и исполняемого файла hello1.exe.
6. Выполнена программа в автоматическом режиме путем набора строки > hello1.exe
7. Запущено выполнение программы под управлением отладчика с помощью команды
> afdpro hello1.ex
hello1.asm

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

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 (CX) = 0000	(DX) = 0000 (IP) = 0018 (CX) = 004B
0018	MOV AH, 09	B409	(IP) = 0018 (AX) = 1A07	(IP) = 001A (AX) = 0907
001A	INT 21	CD21	(IP) = 001A	(IP) = 001C
001C	MOV AH, 4C	B44C	(IP) = 001C (AX) = 0907	(IP) = 001E (AX) = 4C07
001E	INT 21	CD21	(IP) = 001E (AX) = 4C07 (DS) = 1A07 (CX) = 004B	(IP) = 0010 (AX) = 0000 (DS) = 19F5 (CX) = 0000

hello2.asm

CS: 1A0B

DS: 19F5

ES: 19F5

SS: 1A05

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0005	PUSH DS	1E	(IP) = 0005 Stack +0 = 0000 (SP) = 0018	(IP) = 0006 STACK +0 = 19F5 (SP) = 0016
0006	SUB AX, AX	2BC0	(IP) = 0006	(IP) = 0008
0008	PUSH AX	50	(IP) = 0008 Stack +0 = A9F5 Stack +2 = 0000 (SP) = 0016	(IP) = 0009 Stack +0 = 0000 Stack +2 = 19F5 (SP) = 0014
0009	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 (DX) = 0000	(IP) = 0011 (DX) = 0000
0011	CALL 0000	E8ECFF	(IP) = 0011 Stack +0 = 0000 Stack +2 = 19F5 Stack +4 = 0000 (SP) = 0014	(IP) = 0000 Stack +0 = 0014 Stack +2 = 0000 Stack +4 = 19F5 (SP) = 0012
0000	MOV AH, 09	B409	(IP) = 0000 (AX) = 1A07	(IP) = 0002 (AX) = 0907
0002	INT 21	CD21	(IP) = 0002	(IP) = 0004
0004	RET	C3	(IP) = 0004 Stack +0 = 0014 Stack +2 = 0000 Stack +4 = 19F5 (SP) = 0012	(IP) = 0014 Stack +0 = 0000 Stack +2 = 19F5 Stack +4 = 0000 (SP) = 0014
0014	MOV DX, 0010	BA1000	(IP) = 0014 (DX) = 0000	(IP) = 0017 (DX) = 0010
0017	CALL 0000	E8E6FF	(IP) = 0017 Stack +0 = 0000 Stack +2 = 19F5 Stack +4 = 0000 (SP) = 0014	(IP) = 0000 Stack +0 = 001A Stack +2 = 0000 Stack +4 = 19F5 (SP) = 0012
0000	MOV AH, 09	B409	(IP) = 0000	(IP) = 0002
0002	INT 21	CD21	(IP) = 0002	(IP) = 0004
0004	RET	C3	(IP) = 0004 Stack +0 = 001A Stack +2 = 0000 Stack +4 = 19F5 (SP) = 0012	(IP) = 001A Stack +0 = 0000 Stack +2 = 19F5 Stack +4 = 0000 (SP) = 0014

001A	RET FAR	CB	(IP) = 001A Stack +2 = 19F5 (SP) = 0014	(IP) = 0000 Stack +2 = 0000 (SP) = 0018
0000	INT 20	CD20	(IP) = 0000 (AX) = 0907 (CX) = 007B (DX) = 0010 (DS) = 1A07	(IP) = 0005 (AX) = 0000 (CX) = 0000 (DX) = 0000 (DS) = 19F5

Выводы.

В ходе выполнения лабораторной работы были получены основные навыки программирования на ассемблере.

ПРИЛОЖЕНИЕ А

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.1381 - Мелькумянц Д.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
```

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

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

```

```

;              Главная          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

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 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
0000 Greeting LABEL BYTE
; Текст приветствия
0000 82 A0 E1 20 AF E0      DB 'Вас приветствует ст.гр.1303 - Хулап О.А.
                               ',13,10','$'

A8 A2 A5 E2 E1 E2
A2 E3 A5 E2 20 E1
E2 2E A3 E0 2E 31
33 30 33 20 2D 20
95 E3 AB A0 AF 20
8E 2E 80 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          ; # 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
```

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine	Class
DGROUP	GROUP			
_DATA	002B	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 Total Lines

19 Symbols

48000 + 461307 Bytes symbol space free

0 Warning Errors

0 Severe Errors

hello2.lst


```

0000                                DATA    SEGMENT

                                ; Директивы описания данн
                                ых

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 1303 - Hulap Olesya$
                                ,
                                74 20 66 72 6F 6D
                                20 31 33 30 33 20
                                2D 20 48 75 6C 61
                                70 20 4F 6C 65 73
                                79 61 24
0031                                DATA    ENDS

                                ; Код программы

0000                                CODE    SEGMENT

                                ; 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

; Главная процедура

0005 Main    PROC FAR

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

0006 2B C0          sub  AX,AX  ; > для после◆
                     ◆ующего восстановления по
0008 50          push AX      ;/ команде ret
                     , завершающей процедуру.

0009 B8 ---- R     mov  AX,DATA      ; Загр◆
                     ◆зка сегментного
000C 8E D8          mov  DS,AX      ; реги◆
                     ◆тра данных.

000E BA 0000 R     mov  DX, OFFSET HELLO  ; Выво◆
                     ◆на экран первой
0011 E8 0000 R     call WriteMsg      ; стро◆
                     ◆и приветствия.

0014 BA 0010 R     mov  DX, OFFSET GREETING ; Выво◆
                     ◆на экран второй
0017 E8 0000 R     call WriteMsg      ; стро◆
                     ◆и приветствия.

001A CB          ret              ; Выхо◆
                     ◆в DOS по команде,
                                     ; нахо◆
                     ◆ящейся в 1-ом слове 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	0031	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

47992 + 459265 Bytes symbol space free

0 Warning Errors

0 Severe Errors

