

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. 1381

Таргонский М. А.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2022

Цель работы.

Изучить принципы работы простейших программ на языке Ассемблера.

А также выполнить отладку, компоновку и трансляцию данных задач.

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 `> linkhello1.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.

Выполнение работы.

Часть 1.

1. Просмотрена программа hello1.asm
2. Разобрался в структуре и реализации каждого сегмента программы.
3. Загрузил файл hello1.asm из каталога Задания в каталог Masm.
4. Протранслировал программу с помощью строки > masm hello1.asm с созданием объектного файла и файла диагностических сообщений.
5. Скомпоновал загрузочный модуль с помощью строки > link hello1.obj с созданием карты памяти и исполняемого файла hello1.exe.
6. Выполнил программу в автоматическом режиме путем набора строки > hello1.exe
7. Запустил выполнение программы под управлением отладчика с помощью команды > afd hello1.exe

Часть 2

Выполнил 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.

hello1.asm

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До	После
0010	MOV AX, 1A07	B8071A	(AX) = 0000 (IP) = 0010	(AX) = 1A07 (IP) = 0013
0013	MOV DS, AX	8ED8	(DS) = 19F5 (IP) = 0013 (AX) = 1A07	(DS) = 1A07 (IP) = 0015 (AX) = 1A07
0015	MOV DX, 0000	BA0000	(IP) = 0015 (DX) = 0000	(IP) = 0018 (DX) = 0000
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

hello2.asm

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До	После
0005	PUSH DS	1E	(SP) = 0018 (IP) = 0005 (DS) = 19F5 STACK(+0) = 0000	(SP) = 0016 (IP) = 0006 (DS) = 19F5 STACK(+0) = 19F5
0006	SUB AX,AX	2BC0	(IP) = 0006	(IP) = 0008
0008	PUSH AX	50	(SP) = 0016 (IP) = 0008 (AX) = 0000 STACK(+0) = 19F5 STACK(+2) = 0000	(SP) = 0014 (IP) = 0009 (AX) = 0000 STACK(+0) = 0000 STACK(+2) = 19F5
0009	MOV AX,1A07	B8071A	(IP) = 0009 (AX) = 0000	(IP) = 000C (AX) = 1A07
000C	MOV DS,AX	8ED8	(DS) = 19F5 (IP) = 000C (AX) = 1A07	(DS) = 1A07 (IP) = 000E (AX) = 1A07
000E	MOV DX,0000	BA0000	(IP) = 000E (DX) = 0000	(IP) = 0011 (DX) = 0000
0011	CALL 0000	E9ECFF	(SP) = 0014 (IP) = 0011 STACK(+0) = 0000 STACK(+2) = 19F5 STACK(+4) = 0000	(SP) = 0012 (IP) = 0000 STACK(+0) = 0014 STACK(+2) = 0000 STACK(+4) = 19F5
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	(SP) = 0014 (IP) = 0014

			STACK(+0) = 0014 STACK(+2) = 0000 STACK(+4) = 19F5	STACK(+0) = 0000 STACK(+2) = 19F5 STACK(+4) = 0000
0014	MOV DX, 0010	BA1000	(DX) = 0000 (IP) = 0014	DX = 0010 IP = 0017
0017	CALL 0000	3836FF	(IP) = 0017 (SP) = 0014 STACK(+0) = 0000 STACK(+2) = 19F5 STACK (+4) = 0000	IP = 0000 SP = 0012 STACK(+0) = 001A STACK(+2) = 0000 STACK (+4) = 19F5
0000	MOV AH, 09	B409	(IP) = 0000	IP = 0002
0002	INT 21	CD21	(IP) = 0002	IP = 0004
0004	RET	C3	(SP) = 0012 (IP) = 0004 STACK(+0) = 001A STACK(+2) = 0000 STACK (+4) = 19F5	SP = 0014 IP = 001A STACK(+0) = 0000 STACK(+2) = 19F5 STACK (+4) = 0000
001A	RET FAR	CB	(CS) = 1A0A (SP) = 0014 (IP) = 001A STACK(+0) = 0000 STACK(+2) = 19F5 STACK(+4) = 0000	CS = 19F5 SP = 0018 IP = 0000 STACK(+0) = 0000 STACK(+2) = 0000 STACK(+4) = 0000
0000	INT 20	CD20	(AX) = 0907 (DX) = 0010 (CS) = 19F5 (DS) = 1A07 (IP) = 0000	AX = 0000 DX = 0000 CS = 1A0A DS = 19F5 IP = 0005

Выводы.

Изучил принципы работы простейших программ на языке Ассемблера.

А также выполнил отладку, компоновку и трансляцию данных задач.

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

Листинг успешной трансляции программами.

LST1.LST

#Microsoft (R) Macro Assembler Version 5.10

9/14/22 20:23:49

Page 1-1

; HELLO1.ASM - упрощенная версия учебной программы лаб.р
б. 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 -
мещения адреса выводимо й
;          строки;
;          - использует реги
тр ах и не сохраняет его ;
содержимое.
; *****
*****

```

DOSSEG

```

; Задание сегментов под ДО
С

```

```

.MODEL SMALL

```

```

; Модель памяти-SMALL(Малая)

```

```

.STACK 100h

```

```

; Отвести под Стек 256 байт

```

```

.DATA

```

```

; Начало сегмента данных

```

```

0000      Greeting LABEL BYTE

```

```

; Текст приветствия

```

```

0000  D0 92 D0 B0 D1 81      DB 'Вас приветствует ст.г
                               .7303 - Иванов И.И.',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                                ; На
                                ало сегмента кода
0000 B8 ---- R                mov ax, @data                        ; За
                                грузка в DS адреса начала
0003 8E D8                    mov ds, ax                            ; се
                                гмента данных
0005 BA 0000 R                mov dx, OFFSET Greeting              ; За
                                грузка в dx смещения
                                ; ад
                                еса текста приветствия
0008                            DisplayGreeting:
0008 B4 09                    mov ah, 9                            ; # ф
                                ункции ДОС печати строки
000A CD 21                    int 21h                                ; вы
                                од на экран приветствия
000C B4 4C                    mov ah, 4ch                          ; # ф
                                ункции ДОС завершения про
                                граммы
000E CD 21                    int 21h                                ; за
                                ершение программы и выход
                                в ДОС
                                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

47998 + 459262 Bytes symbol space free

0 Warning Errors
0 Severe Errors

LST2.LST

#Microsoft (R) Macro Assembler Version 5.10

9/13/22 14:57:18

Page 1-1

```

; HELLO2 - Учебная программа N2
лаб.раб.#1 по дисциплине "
рхитектура компьютера"
; Программа используе
т процедуру для печати ст
роки
;
; ТЕКСТ ПРОГРАММЫ
= 0024 EOFLine EQU '$' ; Определен
е символьной константы
; "Конец с
роки"

; Стек программы
```

ASSUME CS:CODE, SS:AStack

0000 AStack SEGMENT STACK

0000 000C[DW 12 DUP('!') ; Отводитс

12 слов памяти

0021

]

0018 AStack ENDS

; Данные программы

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 6EGREETING DB 'Student from 4350 - \$'

74 20 66 72 6F 6D

20 34 33 35 30 20

2D 20 24

0025 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

; Головная процедура

#Microsoft (R) Macro Assembler Version 5.10

9/13/22 14:57:18

Page 1-2

0005 Main PROC FAR

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

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

0008 50 push AX ;/ команде ret
, завершающей процедуру.

0009 B8 ---- R mov AX,DATA ; Загрузка
сегментного

000C 8E D8 mov DS,AX ; регистрация
данных.

000E BA 0000 R mov DX, OFFSET HELLO ; Вывод на
экран первой

0011 E8 0000 R call WriteMsg ; строка
и приветствия.

0014 BA 0010 R mov DX, OFFSET GREETING ; Вывод на
экран второй

0017 E8 0000 R call WriteMsg ; строка

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
```

#Microsoft (R) Macro Assembler Version 5.10

9/13/22 14:57:18

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK	0018	PARA	STA	
...			CK	
CODE	001B	PARA	NON	
.			E	
DATA	0025	NONE		

PARA Symbols:

N a m e	Type
Value Attr	
EOFLINE	NUMBE 0024
...	R
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

47990 + 459267 Bytes symbol space free

0 Warning Errors

0 Severe Errors