

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. 0383

Парфенов В.М.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Изучать как устроены ТРАНСЛЯЦИЯ, ОТЛАДКА И ВЫПОЛНЕНИЕ 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

> 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.

Таблица 1 - Пример

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) = 2D75 (IP) = 0003

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

1. Часть 1

- 1.1. Каталог с MASM смонтирован с помощью команды mount с C:\Путь. Использовалась встроенная возможность Windows перетаскивания каталога на иконку приложения DOSBox)
- 1.2. Изменена символьная строка Greeting в сегменте данный на «Вас приветствует ст.гр.0383 - Парфенов В.М.»
- 1.3. Протранслирована программа с помощью MASM.EXE с созданием объектного файла hello1.obj и файлом листинга hello1.lst без ошибок.
- 1.4. Скомпонован загрузочный модуль с созданием исполняемого файла hello1.exe без ошибок.
- 1.5. Программа hello1.exe запущена, все отработало без ошибок, вывелась строка «Вас приветствует ст.гр.0383 - Парфенов В.М.».
- 1.6. Запущено выполнение программы под управлением отладчика с пошаговым выполнением и занесением данных в таблицу 2.

Начальное значение сегментных регистров:

CS = 1A05 ; DS = 19F5;

ES = 19F5 ; SS = 1A08 ;

Таблица 2 - Отладка hello1.exe

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 = 10A7 DS = 19F5 IP = 0013	AX = 10A7 DS = 1A07 IP = 0015
0015	MOV DX, 000	BA0000	DX = 0000 AX = 10A7 DS = 10A7 IP = 0015	DX = 0000 AX = 10A7 DS = 10A7 IP = 0018
0018	MOV AH, 09	B409	AX = 1A07 DS = 10A7 IP = 0018	AX = 0907 DS = 10A7 IP = 001A
001A	INT 21	CD21	AX = 0907 DS = 10A7 IP = 001A	AX = 0907 DS = 10A7 IP = 001C
001C	MOV AH, 4C	B44C	AX = 0907 DS = 10A7 IP = 001C	AX = 4C07 DS = 10A7 IP = 001E
001E	INT 21	CD21	AX = 4C07 DS = 10A7 IP = 001E	AX = 0000 DS = 19F5 IP = 14A1

2. Часть 2

2.1. Прделаны аналогичные шаги 1.1-1.6 для программы hello2.asm.

Результат пошагового выполнения находится в таблице 3.

Начальное значение сегментных регистров:

CS = 1A0A; DS = 19F5;

ES = 19F5; SS = 1A05;

Таблица 3 - Отладка hello2.exe

Адрес Команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			до выполнения	После выполнения

0005	PUSH DS	1E	AX = 0000 IP = 0005 SP = 0018 Stack +0 0000 +2 0000 +4 0000 +6 0000	A[= 0000 IP = 0006 SP = 0016 Stack +0 19F5 +2 0000 +4 0000 +6 0000
0006	SUB AX, AX	2BC0	AX = 0000 IP = 0006 SP = 0016 Stack: + 0 19F5	AX = 0000 IP = 0008 SP = 0016 Stack: + 0 19F5
0008	PUSH AX	50	AX = 0000 IP = 0008 SP = 0016 Stack +0 19F5 +2 0000 +4 0000 +6 0000	AX = 0000 IP = 0009 SP = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0009	MOV AX, 1A07	B8071A	IP = 0009 AX = 0000 SP = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000	IP = 000C AX = 1A07 SP = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000
000C	MOV DS, AX	8ED8	AX = 1A07 IP = 000C SP = 0014 DS = 19F5 Stack +0 0000 +2 19F5 +4 0000 +6 0000	AX = 1A07 IP = 000E DS = 1A07 SP = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000

000E	MOV DX, 0000	BA0000	AX = 1A07 SP = 0014 DX = 0000 IP = 000E Stack +0 0000 +2 19F5 +4 0000 +6 0000	AX = 1A07 SP = 0014 DX = 0000 IP = 0011 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0011	CALL 0000	E8ECFF	AX = 1A07 IP = 0011 SP = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000	AX = 1A07 IP = 0000 SP = 0012 Stack +0 0014 +2 0000 +4 19F5 +6 0000
0000	MOV AH, 09	B409	AX = 1A07 IP = 0000 AX = 1A07	AX = 0907 IP = 0002 AX = 0907
0002	INT 21	CD21	AX = 0907 SP = 0012 IP = 0002 Stack +0 0014 +2 0000 +4 19F5 +6 0000	AX = 0907 SP = 0012 IP = 0004 Stack +0 0014 +2 0000 +4 19F5 +6 0000
0004	RET	C3	AX = 0907 IP = 0004 SP = 0012 Stack +0 0014 +2 0000 +4 19F5 +6 0000	AX = 0907 IP = 0014 SP = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000

0014	MOV DX,0010	BA0F00	AX = 0907 SP = 0014 IP = 0014 DX = 0000 Stack +0 0000 +2 19F5 +4 0000 +6 0000	AX = 0907 SP = 0014 IP = 0017 DX = 0010 Stack +0 0000 +2 19F5 +4 0000 +6 0000
0017	CALL 0000	E8E6FF	AX = 0907 IP = 0017 SP = 0014 Stack +0 0000 +2 19F5 +4 0000 +6 0000	AX = 0907 IP = 0000 SP = 0012 Stack +0 001A +2 0000 +4 19F5 +6 0000
0000	MOV AH, 09	B409	AX = 0907 SP = 0012 IP = 0000 Stack +0 001A +2 0000 +4 19F5 +6 0000	AX = 0907 SP = 0012 IP = 0002 Stack +0 001A +2 0000 +4 19F5 +6 0000
0002	INT 21	CD21	AX = 0907 SP = 0012 IP = 0002 Stack +0 001A +2 0000 +4 19F5 +6 0000	AX = 0907 SP = 0012 IP = 0004 Stack +0 001A +2 0000 +4 19F5 +6 0000
0004	RET	C3	AX = 0907 IP = 0004 SP = 0012 Stack +0 001A	AX = 0907 IP = 001A SP = 0014 Stack +0 0000

			+2 0000 +4 19F5 +6 0000	+2 19F5 +4 0000 +6 0000
001A	Ret Far	CB	AX = 0907 IP = 001A SP = 0014 CS = 1A0A Stack +0 0000 +2 19F5 +4 0000 +6 0000	AX = 0907 IP = 0000 SP = 0018 CS = 19F5 Stack +0 0000 +2 0000 +4 0000 +6 0000
0000	INT 20	CD20	AX = 0907 SP = 0018 IP = 0000 CX = 006B CS = 19F5	AX = 0000 SP = 0018 IP = 0005 CX = 0000 CS = 1A0A

Вывод.

В ходе данной лабораторной работы были изучены основы языка Ассемблера, такие вещи как трансляция, отладка и выполнение программ. Также было отмечено, что программа Hello1.asm более автоматизирована, в отличии от hello2.asm, так как там используются упрощенные директивы.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название 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.0383 - Парфенов В.М',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
```

Название файла: hello1.lst

#Microsoft (R) Macro Assembler Version 5.10

9/12/21 13:29:40

; 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
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 регистры 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 A8 A2 A5 E2 E1 E2 DB 'Вас приветствует ст.гр.0383 – Парфенов
В.М',13,10,'\$'

A2 E3 A5 E2 20 E1

E2 2E A3 E0 2E 30

33 38 33 20 2D 20

8F A0 E0 E4 A5 AD

AE A2 20 82 2E 8C

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

функции ДОС печати строки

```
000A CD 21          int 21h          ; ВЫВОД
#Microsoft (R) Macro Assembler Version 5.10    9/12/21 13:29:40
Page 1-2
```

на экран приветствия

```
000C B4 4C          mov ah, 4ch      ; # фу
```

функции ДОС завершения программы

```
000E CD 21          int 21h          ; завер
```

шение программы и выход в ДОС

END

Название 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 'Вас приветствует ст.гр.0383 – Парфенов В.М.', 0AH,
0DH,EOLine

GREETING DB 'Student from 0383 - \$'

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

```

Название файла: hello2.lst

Microsoft (R) Macro Assembler Version 5.10 9/15/21 11:21:52

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

```

```

ASSUME CS:CODE, SS:AStack

```

```

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 82 A0 E1 20 AF E0      HELLO    DB 'Вас приветствует ст.гр.0383 -

```

Пар

```

                                фенов В.М.', 0AH, 0DH,EOFLine

```

```

                                A8 A2 A5 E2 E1 E2

```

```

                                A2 E3 A5 E2 20 E1

```

```

                                E2 2E A3 E0 2E 30

```

```

                                33 38 33 20 2D 20

```

```

                                8F A0 E0 E4 A5 AD

```

```

                                AE A2 20 82 2E 8C

```

```

                                2E 0A 0D 24

```

```

002E 53 74 75 64 65 6E GREETING DB 'Student from 0383 - $'

```

```

                                74 20 66 72 6F 6D

```

```

                                20 30 33 38 33 20

```

```

                                2D 20 24

```

```

0043                                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 пре
                      рыванию

0004 C3            ret

0005          WriteMsg ENDP

```

Microsoft (R) Macro Assembler Version 5.10 9/15/21 11:21:52

Page 1-2

```

; 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
                      остановления 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
                      экран первой

0011 E8 0000 R     call WriteMsg    ; строки пр
                      иветствия.

```

```

0014 BA 002E 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 9/15/21 11:21:52

Symbols-1

Segments and Groups:

N a m e	Length	Align	Combine	Class
ASTACK	0018	PARA		STACK
CODE	001B	PARA		NONE
DATA	0043	PARA		NONE

Symbols:

N a m e	Type	Value	Attr
EOFLINE	NUMBER	0024	
GREETING	L BYTE	002E	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

48000 + 461307 Bytes symbol space free

0 Warning Errors

0 Severe Errors