

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. 1383

Валиев Р.Р.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Изучить структуру программы, записанной на языке ассемблера.
Получить навыки в работе с транслятором и отладчиком для ассемблера.

Задание.

1. Просмотреть программу hello1.asm.
2. Разобраться в структуре и реализации каждого сегмента программы.
Строку-приветствие преобразовать в соответствии со своими личными данными.
3. Загрузить файл hello1.asm из каталога Задания в каталог Masm.
4. Протранслировать программу 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 созданием карты памяти и исполняемого файла hello1.exe.
6. Выполнить программу в автоматическом режиме, убедиться в корректности ее работы и зафиксировать результат выполнения в протоколе.
7. Запустить выполнение программы под управлением отладчика.
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.

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

1. 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. Строка-приветствие преобразована в соответствии с 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.exe
```

Начальные значения сегментных регистров hello1.exe: CS — 1A05, DS - 19F5, ES —19F5, SS - 1A0A.

Начальные значения сегментных регистров hello2.exe: CS — 1A0A, DS - 19F5, ES —19F5, SS — 1A05.

Таблица 1 – Результаты отладки программы 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 (DS) = 19F5 (IP) = 0010 | (AX) = 1A07 (DS) = 19F5 (IP) = 0013 |
| 0013 | Mov DS, AX | 8ED8 | (AX) = 1A07 (DS) = 19F5 (IP) = 0013 | (AX) = 1A07 (DS) = 1A07 (IP) = 0015 |
| 0015 | Mov DX, 0000 | BA0000 | (AX) = 1A07 (DS) = 1A07 (IP) = 0015 | (AX) = 1A07 (DS) = 1A07 (IP) = 0018 |
| 0018 | Mov AH, 09 | B409 | (AX) = 1A07 (DS) = 1A07 (IP) = 0018 | (AX) = 0907 (DS) = 1A07 (IP) = 001A |
| 001A | Int 21 | CD21 | (AX) = 0907 (DS) = 1A07 (IP) = 001A | (AX) = 0907 (DS) = 1A07 (IP) = 001C |
| 001C | Mov AH, 4C | B44C | (AX) = 0907 (DS) = 1A07 (IP) = 001C | (AX) = 4C07 (DS) = 1A07 (IP) = 001E |
| 001E | Int 21 | CD21 | (AX) = 4C07 (DS) = 1A07 (IP) = 001E | (AX) = 0000 (DS) = 19F5 (IP) = 0010 |

Таблица 2 – Результаты отладки программы 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 (DS) = 19F5 (IP) = 0005 (SP) = 0018 Stack +0 0000 | (AX) = 0000 (DS) = 19F5 (IP) = 0006 (SP) = 0016 Stack +0 19F5 |
| 0006 | Sub AX, AX | 2BC0 | (AX) = 0000 (DS) = 19F5 (IP) = 0006 | (AX) = 0000 (DS) = 19F5 (IP) = 0008 |
| 0008 | Push AX | 50 | (AX) = 0000 (DS) = 19F5 (IP) = 0008 (SP) = 0016 Stack +0 19F5 +2 0000 | (AX) = 0000 (DS) = 19F5 (IP) = 0009 (SP) = 0014 Stack +0 0000 +2 19F5 |
| 0009 | Mov AX, 1A07 | B8071A | (AX) = 0000 (DS) = 19F5 (IP) = 0009 | (AX) = 1A07 (DS) = 19F5 (IP) = 000C |
| 000C | Mov DS, AX | 8ED8 | (AX) = 1A07 (DS) = 19F5 (IP) = 000C | (AX) = 1A07 (DS) = 1A07 (IP) = 000E |
| 000E | Mov DX, 0000 | BA0000 | (AX) = 1A07 (DS) = 1A07 (IP) = 000E | (AX) = 1A07 (DS) = 1A07 (IP) = 0011 |
| 0011 | Call 0000 | E8ECFF | (AX) = 1A07 (DS) = 1A07 (IP) = 0011 (SP) = 0014 Stack +0 0000 +2 19F5 +4 0000 | (AX) = 1A07 (DS) = 1A07 (IP) = 0000 (SP) = 0012 Stack +0 0014 +2 0000 +4 19F5 |

| | | | | |
|------|--------------|--------|---|---|
| 0000 | Mov AH, 09 | B409 | (AX) = 1A07 (DS) = 1A07 (IP) = 0000 | (AX) = 0907 (DS) = 1A07 (IP) = 0002 |
| 0002 | Int 21 | CD21 | (AX) = 0907 (DS) = 1A07 (IP) = 0002 | (AX) = 0907 (DS) = 1A07 (IP) = 0004 |
| 0004 | Ret | C3 | (AX) = 0907 (DS) = 1A07 (IP) = 0004 (SP) = 0012 Stack +0 0014 +2 0000 +4 19F5 | (AX) = 0907 (DS) = 1A07 (IP) = 0014 (SP) = 0014 Stack +0 0000 +2 19F5 +4 0000 |
| 0014 | Mov DX, 0010 | BA1000 | (AX) = 0907 (DS) = 1A07 (IP) = 0014 (DX) = 0000 | (AX) = 0907 (DS) = 1A07 (IP) = 0017 (DX) = 0010 |
| 0017 | Call 0000 | E8E6FF | (AX) = 0907 (DS) = 1A07 (IP) = 0017 (SP) = 0014 Stack +0 0000 +2 19F5 +4 0000 | (AX) = 0907 (DS) = 1A07 (IP) = 0000 (SP) = 0012 Stack +0 001A +2 0000 +4 19F5 |
| 0000 | Mov AH, 09 | B409 | (AX) = 0907 (DS) = 1A07 (IP) = 0000 | (AX) = 0907 (DS) = 1A07 (IP) = 0002 |
| 0002 | Int 21 | CD21 | (AX) = 0907 (DS) = 1A07 (IP) = 0002 | (AX) = 0907 (DS) = 1A07 (IP) = 0004 |
| 0004 | Ret | C3 | (AX) = 0907 (DS) = 1A07 (IP) = 0004 | (AX) = 0907 (DS) = 1A07 (IP) = 001A |

| | | | | |
|------|---------|------|--|--|
| | | | (SP) = 0012 Stack +0 001A +2 0000 +4 19F5 | (SP) = 0014 Stack +0 0000 +2 19F5 +4 0000 |
| 001A | Ret Far | CB | (AX) = 0907 (DS) = 1A07 (IP) = 001A (CS) = 1A0B (SP) = 0014 Stack +0 0000 +2 19F5 +4 0000 | (AX) = 0907 (DS) = 1A07 (IP) = 0000 (CS) = 19F5 (SP) = 0018 Stack +0 0000 +2 0000 +4 0000 |
| 0000 | Int 20 | CD20 | (AX) = 0907 (DS) = 1A07 (IP) = 0000 | (AX) = 0000 (DS) = 19F5 (IP) = 0005 |

Выводы.

Изучена структура программы, записанной на языке ассемблера.
Получены навыки в работе с транслятором и отладчиком для ассемблера.

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.1383 - Валиев Р.Р.',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: 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 1383 - Valiev R.R.$'
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
```

ПРИЛОЖЕНИЕ В

ТЕКСТЫ ФАЙЛОВ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: hello1.lst

#Microsoft (R) Macro Assembler Version 5.10
10/11/22 19:26:15

Page

1-1

```
прогр          ; HELLO1.ASM - упрощенная версия учебной
                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.1383 -
Валиев Р.Р.',13,10,'$'
      A8 A2 A5 E2 E1 E2
      A2 E3 A5 E2 20 E1
      E2 2E A3 E0 2E 37
      33 30 33 20 2D 20
      88 A2 A0 AD AE A2
      20 88 2E 88 2E 0D
      0A 24

; .CODE
Начал
0000 B8 ---- R о сегмента кода
Загр mov ax, @data
; сегм
0003 8E D8 узка в DS адреса начала
mov ds, ax
0005 BA 0000 R ента данных
Загр mov dx, OFFSET Greeting
узка в dx смещения

#Microsoft (R) Macro Assembler Version 5.10
10/11/22 19:26:15
1-2
Page
;
адрес
0008 а текста приветствия
0008 B4 09 DisplayGreeting:
; # фу mov ah, 9
000A нкции ДОС печати строки
CD 21 int 21h
; вывод
000C на экран приветствия
B4 4C mov ah, 4ch
; # фу
000E нкции ДОС завершения программы
CD 21 int 21h
; завер
шение программы и выход в ДОС
END

#Microsoft (R) Macro Assembler Version 5.10
10/11/22 19:26:15
Symb
ols-1

```

Segments and Groups:

| Class | N a m e | Length | Align | Combine |
|--------|-----------|--------|-------------|---------|
| DGROUP | | GROUP | | |
| _DATA | | 002C | 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

Название файла: hello2.lst

#Microsoft (R) Macro Assembler Version 5.10

10/11/22 19:29:43

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
СИМВОЛЬ        ной константы
;          "Конец строки"

; Стек программы

ASSUME CS:CODE, SS:AStack

0000          AStack      SEGMENT  STACK
0000 000C[          DW 12 DUP('!')      ; Отводится 12
СЛОВ          памяти
          0021          ]

0018          AStack      ENDS

; Данные программы

0000          DATA      SEGMENT

; Директивы описания данных
```

```

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 1383 - Valiev
R.R.$'
      74 20 66 72 6F 6D
      20 31 33 38 33 20
      2D 20 4D 61 6C 79
      6B 68 20 41 2E 41
      2E 24
0030 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

; Головная процедура
0005 Main PROC FAR
0005 1E push DS ;\ Сохранение адреса

#Microsoft (R) Macro Assembler Version 5.10
10/11/22 19:29:43

Page
1-2

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
экран первой
0011  E8 0000 R        call  WriteMsg           ; строки пр
иветствия.
0014  BA 0010 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

10/11/22 19:29:43

Symbols-1

Segments and Groups:

| N a m e | Length | Align | Combine Class |
|------------------|--------|-------|---------------|
| ASTACK | 0018 | PARA | STACK |
| CODE | 001B | PARA | NONE |
| DATA | 0030 | 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

48002 + 461305 Bytes symbol space free

0 Warning Errors

0 Severe Errors