

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»
Трансляция, отладка и выполнение программ на языке Ассемблера

Студент гр. 0383

Девятериков И.С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Познакомиться с работой программ, написанных на Ассемблере. Изучить особенности работы низкоуровневого языка программирования. Проследить перевод инструкций Ассемблера в машинный код.

Задание.

Лабораторная работа 1 использует 2 готовых программы на ассемблере: hello1 – составлена с использованием сокращенного описания сегментов.

hello2 – составлена с полным описанием сегментов и выводом строки, оформленным как процедура. Выполнение работы состоит из двух частей, по каждой из которых необходимо представить протокол с фиксацией всех выполняемых действий и полученных результатов.

Уточнение задания следует посмотреть в файле lr1_comp.txt каталога Задания.

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.

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.

В ходе работы были скомпилированы два файла hello1.asm и hello2.asm. Работа обоих из них представлена в Табл. 1 и Табл. 2 соответственно.

| | | | | |
|-------------|-------------|-------------|-------------|---------------|
| (AX) = 0000 | (SI) = 0000 | (CS) = 1A05 | (IP) = 0010 | Stack +0 8325 |
| (BX) = 0000 | (DI) = 0000 | (DS) = 19F5 | | +2 04C4 |
| (CX) = 0051 | (BP) = 0000 | (ES) = 19F5 | (HS) = 19F5 | +4 068E |
| (DX) = 0000 | (SP) = 0100 | (SS) = 1A0B | (FS) = 19F5 | +6 4FAA |

Табл. 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 | (IP) = 001E | |

(AX) = 0000 (SI) = 0000 (CS) = 1A0B (IP) = 0005 Stack +0 0000
 (BX) = 0000 (DI) = 0000 (DS) = 19F5 +2 0000
 (CX) = 007B (BP) = 0000 (ES) = 19F5 (HS) = 19F5 +4 0000
 (DX) = 0000 (SP) = 0018 (SS) = 1A05 (FS) = 19F5 +6 0000

Табл. 2

| Адрес Команды | Символический код команды | 16-ричный код команды | Содержимое регистров и ячеек памяти | |
|------------------|------------------------------|--------------------------|---|---|
| | | | до выполнения | после выполнения |
| 0005 | PUSH DS | 1E | (SP) = 0018 (IP) = 0005 Stack +0 0000 +2 0000 +4 0000 +6 0000 | (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 | MOVE 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 19F5 +4 0000 +6 0000 |
| 0014 | MOV DX, 0010 | BA0000 | (DX) = 0000 (IP) = 0014 | (DX) = 0010 (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 19F5 +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 (CS) = 1A0B (IP) = 001A Stack +0 0000 +2 19F5 | (SP) = 0018 (CS) = 19F5 (IP) = 0000 Stack +0 0000 +2 0000 |

| | | | | |
|------|--------|------|-------------|---------|
| | | | +4 0000 | +4 0000 |
| | | | +6 0000 | +6 0000 |
| 0000 | INT 20 | CD20 | (IP) = 0000 | |

Выводы.

В ходе выполнения лабораторной работы произошло ознакомление с некоторыми особенностями работы Ассемблера. Было выявлено, что в идентичные по задаче программы могут нести в себе, значительное различие как в непосредственной реализации, так и в хранении данных после этапа компиляции.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

; 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 - Девятериков И.С.',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 - 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 0383 - Devyaterikov Ilya\$'

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

```

прогр ; HELLO1.ASM - упрощенная версия учебной

аммы лаб.раб. N1

; по дисциплине "Архитектура комп
ьютера"

;

на ; Назначение: Программа формирует и выводит

экран приветствие

; пользователя с помощью функции ДО
С "Вывод строки"

; (номер 09 прерывание 21h), которая
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.0383 -

```

```

Девятерико

```

```

в И.С.',13,10,'$'

```

```

A8 A2 A5 E2 E1 E2

```

```

A2 E3 A5 E2 20 E1

```

```

E2 2E A3 E0 2E 30

```

```

33 38 33 20 2D 20

```

```

84 A5 A2 EF E2 A5

```

```

E0 A8 AA AE A2 20

```

```

88 2E 91 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 |

```
        ; указка 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
        ; 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
```

Symbols-1

Segments and Groups:

| N a m e | Length | Align | Combine | Class |
|--------------|--------|-------|---------|---------|
| DGROUP | GROUP | | | |
| _DATA | 0031 | 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

```

; 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
п
амяти
????
]

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 0383 -
Devyaterikov

Ilya\$'

74 20 66 72 6F 6D

20 30 33 38 33 20

2D 20 44 65 76 79

61 74 65 72 69 6B

6F 76 20 49 6C 79

61 24

0036 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 |
| | | ; Главная 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 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

Symbols-1

Segments and Groups:

| N a m e | Length | Align | Combine | Class |
|--------------|--------|-------|---------|-------|
| ASTACK | 0018 | PARA | | STACK |
| CODE | 001B | PARA | | NONE |
| DATA | 0036 | 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