

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. 9382

Пя С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Изучить строение программ на языке Ассемблера, научиться следить за изменениями в ходе их выполнения, познакомиться с функционалом языка.

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.

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. Открыв в текстовом редакторе программу hello1.asm, изучила ее строение, обратив особое внимание на команды вывода.
2. Пошагово разобралась в структуре кода и преобразовала строку-приветствие в соответствии со своими личными данными.

3. Загрузила файл hello1.asm в каталог Masm.
4. Протранслировала программу с созданием объектного файла и файла диагностических сообщений. Ошибок, как и предупреждений, не было обнаружено.
5. Скомпилировала загрузочный модуль с созданием карты памяти и исполняемого файла hello1.exe.
6. Выполнила программу, предварительно введя строку для правильного отображения русских слов, программа сработала корректно, вывод программы:

> Вас приветствует ст.гр. 9382 – Пя С.

7. Запустила выполнение программы под управлением отладчика.

Содержимое сегментных регистров до старта программы: CS:1A05, DS:19F5, ES:19F5, SS:1A0A, HS:19F5, FS:19F5

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	(DS) = 19F5 (IP) = 0013	(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	(AX) = 4C07 (DS) = 1A07 (IP) = 001E	(AX) = 0000 (DS) = 19F5 (IP) = 0010

Часть 2.

1. Открыла программу hello2.asm с помощью текстового редактора, акцентируя внимание на процедуру WriteMsg.
2. Изучила строение программы hello2.asm, параллельно преобразовав строку-приветствие в соответствие с личными данными.
3. Загрузила hello2.asm в каталог Masm.

4. Протранслировала программу с созданием объектного файла и файла с диагностическими сообщениями. Синтаксических ошибок не обнаружено.
5. Скомпилировала загрузочный модуль с созданием карты памяти и исполняемого файла hello2.exe.
6. Выполнила программу hello2.exe и проверила корректность ее работы:
 > Hello Worlds!
 > Student from 9382 – Pya Son Khwa
7. Запустила программу с помощью отладчика afddpro.exe и пошагово исполнила с записью используемых регистров и ячеек команд.

Содержимое сегментных регистров до старта программы: CS:1A0B, DS:19F5, ES:19F5, SS:1A05, HS:19F5, FS:19F5

Табл.2.

Адрес Команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти	
			До выполнения	После выполнения
0005	PUSH DS	1E	(DS) = 19F5 (SP) = 0018 (IP) = 0005 Stack +0 0000 +2 0000 +4 0000	(DS) = 19F5 (SP)=0016 (IP) = 0006 Stack +0 19F5 +2 0000 +4 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	(AX) = 0000 (SP) = 0014 (IP) = 0009 Stack +0 0000 +2 19F5 +4 0000
0009	MOV AX, 1A07	B8071A	(AX) = 0000 (IP) = 0009	(AX) = 1A07 (IP) = 000C
000C	MOV 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	(SP) = 0012 (IP) = 0000 Stack +0 0014 +2 0000 +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 Stack +0 0014 +2 0000 +4 19F5	(SP) = 0014 (IP) = 0014 Stack +0 0000 +2 119C +4 0000
0014	MOV DX,0010	BA1000	(DX) = 0000 (IP) = 0014	(DX) = 0010 (IP) = 0017
0017	CALL 0000	E8E6FF	(SP)=0014 (IP) = 0017 Stack +0 0000 +2 19F5 +4 0000	(SP) = 0012 (IP) = 0000 Stack +0 001A +2 0000 +4 19F5
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	(SP) = 0014 (IP) = 001A Stack +0 0000 +2 19F5 +4 0000
001A	RET Far	CB	(SP) = 0014 (IP) = 001A Stack +0 0000 +2 19F5 +4 0000	(SP) = 0018 (IP) = 0000 Stack +0 0000 +2 0000 +4 0000
0000	INT 20	CD20	(AX) = 0907 (DS) = 1A07 (IP) = 0000	(AX) = 0000 (DS) = 19F5 (IP) = 0005

Выводы.

В ходе выполнения лабораторной работы было совершенно ознакомление с языком Ассемблера, был освоен этап трансляции, также была изучена структура данных программ и работа с программой DOSBox 0.74-3.

Приложение.

Текст файла HELLO1.ASM

```
; HELLO1.ASM - rĭa®ЙГ п ŷГабЕп гзГŷ® ĩa®Ja ¬¬л « ŷ.а ŷ. N1
;               ĩ® ¢ЕбжЕĭ«ЕГ "ѲаеЕвГЄвга €®¬ĭмовГа "
;
*****
; К § зГЕГ: Ѳa®Ja ¬¬ д®а¬ЕагГв Е ŷлŷ®Ев нСа ĩаЕŷГвбвŷЕГ
;               ĩ®«м$®ŷ вГ«п б ĩ®¬®Ймо дгЄжЕЕ „Ѳ` ",лŷ® ¢ва®СЕ"
;               (®¬Га 09 ĩаГалŷ ЕГ 21h), €в®а п:
;               - ®ŷГбĭГзЕŷ Гв ŷлŷ® нСа ¢ва®СЕ бЕ¬ŷ®«®ŷ,
;               § € зЕŷ ойГ®бп § €®¬ "$";
;               - ваГŷгГв § ¢ Еп ŷ аГЈЕбваГ аh ®¬Га дгЄжЕЕ=09h,
;               ŷ аГЈЕбваГ dx - б¬ГЙГЕп ¢аГб ŷлŷ®Е¬®
;               ¢ва®СЕ;
;               - Ебĭ®«м$гГв аГЈЕбва ах Е Г б®еа пГв ГЈ®
;               ¢®аГа|Е¬®Г.
;
*****

                DOSSEG                                ; ‡ ¢ ЕГ бГЈ¬Гв®ŷ
ĭ® ¢ „Ѳ`
                .MODEL    SMALL                        ; Ь®аГ«м ĭ ¬пвЕ-
SMALL (Ъ « п)
                .STACK    100h                          ; ѲвŷГбвЕ ĩ® ¢ `вГЄ
256 ŷ €в
                .DATA                                ; К з «® бГЈ¬Гв ¢ ле
Greeting LABEL BYTE                                ; 'ГЄбв ĩаЕŷГвбвŷЕп
                DB 'Вас приветствует ст.гр.9382 - Пя С.',13,10,'$'
                .CODE                                ; К з «® бГЈ¬Гв €®
                mov ax, @data                          ; ‡ Јаг$Є ŷ DS ¢аГб з «
                mov ds, ax                             ; бГЈ¬Гв ¢ ле
                mov dx, OFFSET Greeting                ; ‡ Јаг$Є ŷ dx б¬ГЙГЕп

; ¢аГб вГЄбв ĩаЕŷГвбвŷЕп
                DisplayGreeting:
                mov ah, 9                                ; # дгЄжЕЕ „Ѳ` ĭГз вЕ
¢ва®СЕ
                int 21h                                ; ŷлŷ® нСа ĩаЕŷГвбвŷЕп
                mov ah, 4ch                            ; # дгЄжЕЕ „Ѳ` § ŷГаиГЕп
ĭa®Ja ¬¬л
                int 21h                                ; § ŷГаиГЕГ ĭa®Ja ¬¬л Е
ŷле® ŷ „Ѳ`
                END
```

Текст файла HELLO2.ASM

```
; HELLO2 - "зГЎ п ĩa®Ja ¬¬ N2 « Ÿ.a Ÿ.#1 ĩ® ¤ĖĖжĖĭ«ĖГ
"БаеĖвГĖвга Ė®¬ĭмовГа "

;          Цa®Ja ¬¬ ĖĖĭ®«м$гГв ĩa®жГ¤гaг ¤«п ĭГз вĖ бвa®ĖĖ
;
;      '...Љ''  ЦђГѓђђђђђђ>

EOFLine EQU '$'          ; ГĭaГ¤Г«ГĖГ бĖ¬Ÿ®«м®© Ė®бв вл
;          "Љ®Гж бвa®ĖĖ"

; 'вГĖ ĩa®Ja ¬¬л

AStack    SEGMENT    STACK
            DW 12 DUP(?)    ; ГвŸ®¤Ėвбп 12 б«®Ÿ ĭ ¬пвĖ
AStack    ENDS

; „ лГ ĩa®Ja ¬¬л

DATA      SEGMENT

; „ĖaГĖвĖŸл ®ĭĖĖ Ėп ¤ лe

HELLO     DB 'Hello Worlds!', 0AH, 0DH, EOFLine
GREETING  DB 'Student from 9382 - Pya Son Khwa$'
DATA      ENDS

; Љ®¤ ĩa®Ja ¬¬л

CODE      SEGMENT

            ASSUME CS:Code DS:DATA SS:AStack

; Цa®жГ¤гa ĭГз вĖ бвa®ĖĖ
```



```

WriteMsg PROC NEAR

    mov     AH,9

    int     21h    ; ,л$®Ÿ дг€жЁЁ DOS ĩ® ĩаГалŸ Ёо

    ret

WriteMsg ENDP

; ĩ®«®Ÿ п ĩа®жГга

Main PROC FAR

    push    DS      ;\  '®ea ГЁГ маГб з « PSP Ÿ бвГГГ
    sub     AX,AX    ; > «п ĩ®б«ГгагойГЈ® Ÿ®ббв ®Ÿ«ГЁп ĩ®
    push    AX      ;/ €®¬ «Г ret, $ ŸГгаи ойГ© ĩа®жГгаг.
    mov     AX,DATA      ; ‡ Јар$€ бГЈ¬Гв®Ј®
    mov     DS,AX      ; аГЈЁбва « ле.
    mov     DX, OFFSET HELLO ; ,лŸ®« нСа ĩГай®©
    call    WriteMsg    ; бва®€Ё ĩаЁŸГвбвŸЁп.
    mov     DX, OFFSET GREETING ; ,лŸ®« нСа Ÿв®а®©
    call    WriteMsg    ; бва®€Ё ĩаЁŸГвбвŸЁп.
    ret      ; ,ле®« Ÿ DOS ĩ® €®¬ «Г,
    ;      е®«пйГ©бп Ÿ 1-®¬

б«®ŸГ PSP.

Main ENDP

CODE ENDS

END Main

```