

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 системы»
Тема: Трансляция, отладка и выполнение программ на языке
Ассемблер

Студента гр. 1383

Панов М.Ю.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Научиться работать с системой dos и с программами на ассемблере.

Задание.

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`
- 4 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.

Выполнение работы.

Программа была отредактирована под личные данные. Программа была протранслирована с созданием объектного файла hello1.obj и файла листинга hello1.lst. После компоновки hello1.obj создается исполняемый файл hello1.exe.

При выполнении программы в автоматическом режиме программа выводит “Вас приветствует ст.гр.1383 - Панов М.Ю.”, после чего запускаем программу в отладчике.

Начальное содержимое сегментных регистров:

CS=1A05; DS=19F5; ES=19F5; SS=1A0B;3

Результат прогона программы hello1.exe в отладчике:

Адрес команды	Символический код команды	16-ричный код команды	Содержимое регистров и ячеек памяти.	
			до выполнения	после выполнения
0010	mov AX, 1A07	B8 07 1A	(AX)=0000 (IP)=0010	(AX)=1A07 (IP)=0013
0013	mov DS, AX	8E D8	(DS)=19F5 (IP)=0013	(DS)=1A07 (IP)=0015
0015	mov DX, 0000	BA 00 00	(IP)=0015	(IP)=0018
0018	mov AX, 09	B4 09	(AX)=1A07 (IP)=0018	(AX)=0907 (IP)=001A

001A	int 21	CD 21	(IP)=001A	(IP)=001C
001C	mov AH, 4C	B4 4C	(AX)=0907 (IP)=001C	(AX)=4C07 (IP)=001E
001E	int 21	CD 21	(AX)=4C07 (CX)=0055 (DS)=1A07 (IP)=001E	(AX)=0000 (CX)=0055 (DS)=19F5 (IP)=0010

При выполнении программы HELLO2.exe в автоматическом режиме выводится сообщение “Hello Worlds!

Student from 1383 - Panov Mikhail”

Начальное содержимое сегментных регистров:

CS=1A0B; DS=19F5; ES=19F5; SS=1A05;

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	(SP)=0016 (IP)=0006 Stack: +0 19F5
0006	SUB AX, AX	2B C0	(IP)=0006	(IP)=0008
0008	PUSH AX	50	(SP)=0016 (IP)=0008 Stack: +0 19F5 +0 0000	(SP)=0014 (IP)=0009 Stack: +0 0000 +2 19F5

0009	mov AX, 1A07	B8 07 1A	(AX)=0000 (IP)=0009	(AX)=1A07 (IP)=000C
000C	mov DS, AX	8ED8	(DS)=19F5 (IP)=000C	(DS)=1A07 (IP)=000E
000E	mov DX, 0000	BA 00 00	(IP)=000E	(IP)=0011
0011	CALL 0000	E8 EC FF	(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	B4 09	(IP) = 0000 (AX) = 1A07	(IP) = 0002 (AX) = 0907
0002	INT 21	CD 21	(IP) = 0002	(IP) = 0004
0004	RET	C3	(IP) = 0004 (SP) = 0012 Stack: +0 0014 +2 0000 +4 19F5	(IP) = 0014 (SP) = 0014 Stack: +0 0000 +2 19F5 +4 0000
0014	MOV DX,0010	BA 10 00	(IP) = 0014 (DX) = 0000	(IP) =0017 (DX) = 0010
0017	CALL 0000	E8 E6 FF	(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	B4 09	(IP) = 0000	(IP)= 0002
0002	INT 21	CD 21	(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	(IP) = 001A (CS) = 1A0B (SP) = 0014 Stack: +2 19F5	(IP) = 0000 (CS) = 19F5 (SP) = 0018 Stack: +2 0000
0000	INT 20	CD 20	(AX) = 0907 (CX) = 007B (DX) = 0010 (CS) = 19F5 (DS) = 1A07 (IP) = 0000	(AX) = 0000 (CX) = 0000 (DX) = 0000 (CS) = 0000 (DS) = 19F5 (IP) = 0005

Выводы.

В ходе работы были изучены способы трансляции, отладки и запуска программ на ассемблере.

Исходный код

HELLO1:

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 - Панов М.Ю.',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:

```

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 - Panov Mikhail - \$'

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