

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»
(БГТУ им. В.Г. Шухова)**



ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЯЮЩИХ СИСТЕМ

Лабораторная работа №1

по дисциплине: Архитектура вычислительных систем

тема: «Разработка программ на ассемблере.

Работа с отладчиком x32dbg, пакетом masm32»

Выполнил: ст. группы ПВ-223
Пахомов Владислав Андреевич

Проверили:
ст. пр. Осипов Олег Васильевич

Белгород 2024 г.

Лабораторная работа №1

Разработка программ на ассемблере. Работа с отладчиком x32dbg, пакетом masm32

Вариант 8

Цель работы: получить навыки создания простейших ассемблерных программ с использованием пакета masm32 и научиться пользоваться отладчиком x32dbg.

Задания для выполнения к работе:

1. Ознакомиться со средой x32dbg и компилятором masm32.
2. Создать и скомпилировать программу в соответствии с вариантом задания. В программу включить комментарии с описанием, что делает каждая инструкция. Подробное описание каждой команды можно найти в приложении учебника В.И. Юрова «Assembler», начиная со стр. 511. Комментарии следует выравнивать по левому краю (как в примере).
3. С помощью отладчика определить местонахождение переменных, строк и массивов в сегменте данных, а также их размер. Составить таблицу и подробное описание ячеек сегмента данных (как в примере).
4. Выполнить пошаговую трассировку программы. Определить какие регистры, флаги и ячейки памяти изменяют свои значения в процессе выполнения команд. Обеспечить корректное завершение программы вызовом системной функции ExitProcess с кодом завершения 0. Если в сегменте данных есть строки, то вывести её в консоль. Трассировку требуется выполнить до команды «call ExitProcess» включительно. Составить для каждой инструкции таблицу трассировки (как в примере).
5. Сделать выводы о проделанной работе.

Задание:

```
.DATA
    a DD 30201, 30201h
    b DB 43h, 0F3h, 0F3h, 0E5h
    DF 1500
    DD 1.5, 1.6, 1.9, -1.9
    t DQ 0E7D32A1h
    stra DB 16 dup(1)

.CODE
START:
    MOV ESI, 65737341h
    AND ESI, dword ptr b
    MOV dword ptr stra, ESI
    MOV ECX, dword ptr t
    IMUL ECX, 7
    ADD ECX, 6
    MOV dword ptr stra[4], ECX
    ADD stra[8], 'q'
    DEC stra[9]

END START
```

1. Создать файл lab1.asm со следующим содержанием:

```
.686
.model flat, stdcall
option casemap: none

include windows.inc
include kernel32.inc
include msvcrt.inc
includelib kernel32.lib
includelib msvcrt.lib

.data
a DD 30201, 30201h
b DB 43h, 0F3h, 0F3h, 0E5h
DF 1500
DD 1.5, 1.6, 1.9, -1.9
t DQ 0E7D32A1h
stra DB 16 dup(1)

.code
start:
MOV ESI, 65737341h
AND ESI, dword ptr b
MOV dword ptr stra, ESI
MOV ECX, dword ptr t
IMUL ECX, 7
ADD ECX, 6
MOV dword ptr stra[4], ECX
ADD stra[8], 'q'
DEC stra[9]

push offset stra
call crt_puts
ADD ESP, 4

call crt__getch
push 0
call ExitProcess
end start
```

2. Скомпилировать программу и получить исполняемый файл lab1.exe.
3. Открыть lab1.exe в отладчике
4. Сегмент данных содержит массивы a, b, stra; переменную t

Адрес	Шестнадцатеричное																ASCII
00403000	F9	75	00	00	01	02	03	00	43	F3	F3	E5	DC	05	00	00	uu.....C66aU...
00403010	00	00	00	00	C0	3F	CD	CC	CC	3F	33	33	F3	3F	33	33	...A?iii?336?33
00403020	F3	BF	A1	32	7D	0E	00	00	00	00	01	01	01	01	01	01	6zi2}.....
00403030	01	01	01	01	01	01	01	01	01	01	00	00	00	00	00	00

Адрес	Шестнадцатеричное																ASCII
00403000	F9	75	00	00	01	02	03	00	43	F3	F3	E5	DC	05	00	00	uu.....C66aU...
00403010	00	00	00	C0	C0	3F	CD	CC	CC	3F	33	33	F3	3F	33	33	...A?iii?336?33
00403020	F3	BF	A1	32	7D	0E	00	00	00	00	01	01	01	01	01	01	6zi2}.....
00403030	01	01	01	01	01	01	01	01	01	01	00	00	00	00	00	00

Название переменной	Начальный адрес	Конечный адрес	Размер данных, байт	Описание
a	00403000	00403007	8	Массив a из двух 4-байтовых чисел
b	00403008	0040300B	4	Массив b из четырёх 1-байтовых чисел
-	0040300C	00403011	6	Неименованная область, содержащая число размером 6 байт
-	00403012	00403021	16	Неименованная область, содержащая четыре вещественных числа длиной 4 байт
t	00403022	00403029	8	8-байтовое число t
stra	0040302A	00403039	16	Массив stra из 16 1-байтовых чисел
Общий размер сегмента данных:			58	

Ячейки с адресами 00403000-00403003 содержит число $30201_{10} = 75F9_{16}$

Ячейки с адресами 00403004-00403007 содержит число 030201_{16}

Ячейки с адресами 00403008-0040300B содержит части строки "Asse" в непреобразованном виде, для получения строки "Asse" в программе будет применена маска.

Ячейки с адресами 0040300C-00403011 содержат число $1500_{10} = 05DC_{16}$

Ячейки с адресами 00403012-00403021 содержат 4 вещественных числа

Ячейки с адресами 00403022-00403029 содержит 8-байтовое число $243085985_{10} = 0E7D32A1_{16}$

Ячейки с адресами 0040302A-00403039 содержит 16 ASCII символов проинициализированных 01.

5. Пошаговая трассировка программы

00401000	BE 41737365	mov esi,65737341	esi:EntryPoint
00401005	2335 08304000	and esi,dword ptr ds:[403008]	esi:EntryPoint
0040100B	8935 2A304000	mov dword ptr ds:[40302A],esi	esi:EntryPoint
00401011	8B0D 22304000	mov ecx,dword ptr ds:[403022]	ecx:EntryPoint
00401017	6BC9 07	imul ecx,ecx,7	ecx:EntryPoint
0040101A	83C1 06	add ecx,6	ecx:EntryPoint
0040101D	890D 2E304000	mov dword ptr ds:[40302E],ecx	ecx:EntryPoint
00401023	8005 32304000 71	add byte ptr ds:[403032],71	ecx:EntryPoint
0040102A	FE0D 33304000	dec byte ptr ds:[403033]	
00401030	68 2A304000	push lab1.40302A	
00401035	FF15 08204000	call dword ptr ds:[&puts]	
0040103B	83C4 04	add esp,4	
0040103E	FF15 0C204000	call dword ptr ds:[&_getch]	
00401044	6A 00	push 0	
00401046	E8 01000000	call <JMP.&ExitProcess>	
0040104B	CC	int3	
0040104C	FF25 00204000	jmp dword ptr ds:[&ExitProcess]	JMP.&ExitProcess

Исходное состояние регистров:

EAX=	0019FFCC	EBX=	0031A000	ECX=	00401000	EDX=	00401000
ESP=	0019FF78	EBP=	0019FF84	ESI=	00401000	EDI=	00401000
EIP=	00401000						
ZF=	1	PF=	1	AF=	0		
OF=	0	SF=	0	DF=	0		
CF=	0	TF=	1	IF=	1		

```
mov esi,65737341
```

КОП:	BE 41737365
------	-------------

EAX=	0019FFCC	EBX=	0031A000	ECX=	00401000	EDX=	00401000
ESP=	0019FF78	EBP=	0019FF84	ESI=	65737341	EDI=	00401000
EIP=	00401005						
ZF=	1	PF=	1	AF=	0		
OF=	0	SF=	0	DF=	0		
CF=	0	TF=	0	IF=	1		

Записывает в регистр ESI число 65737341. Увеличивает EIP на 5

```
and esi,dword ptr  
ds:[403008]
```

КОП:	2335 08304000
------	---------------

EAX=	0019FFCC	EBX=	0031A000	ECX=	00401000	EDX=	00401000
ESP=	0019FF78	EBP=	0019FF84	ESI=	65737341	EDI=	00401000
EIP=	0040100B						
ZF=	0	PF=	1	AF=	0		
OF=	0	SF=	0	DF=	0		
CF=	0	TF=	0	IF=	1		

Выполняет побитовое AND над регистром ESI со значением из ячейки памяти с адресом 403008 и записывает результат в ESI. Увеличивает EIP на 5. Обнуляет флаг ZF.

```
mov dword ptr
ds:[40302A],esi
```

КОП:	8935 2A304000
------	---------------

EAX=	0019FFCC	EBX=	0031A000	ECX=	00401000	EDX=	00401000
ESP=	0019FF78	EBP=	0019FF84	ESI=	65737341	EDI=	00401000
EIP=	00401011						
ZF=	0	PF=	1	AF=	0		
OF=	0	SF=	0	DF=	0		
CF=	0	TF=	0	IF=	1		

Записываем в ячейку с адресом 40302A значение из ячейки ESI.
Увеличивает EIP на 6.

```
mov ecx,dword ptr
ds:[403022]
```

КОП:	8B0D 22304000
------	---------------

EAX=	0019FFCC	EBX=	0031A000	ECX=	0E7D32A1	EDX=	00401000
ESP=	0019FF78	EBP=	0019FF84	ESI=	65737341	EDI=	00401000
EIP=	00401017						
ZF=	0	PF=	1	AF=	0		
OF=	0	SF=	0	DF=	0		
CF=	0	TF=	0	IF=	1		

Записываем в EAX значение из ячейки памяти с адресом 403022.
Увеличивает EIP на 6

Выполняет целочисленное произведение ЕСХ на 7 и записывает результат в ЕСХ. Увеличивает ЕІР на 3. Устанавливает РF (флаг чётности) на 0.

Увеличивает ЕСХ на 6 и сохраняет результат сложения в ЕСХ. Увеличивает ЕІР на 3.

Записываем в ячейку с адресом 40302E значение из ячейки ESX. Увеличивает EIP на 6.

Увеличивает ячейку с адресом 403032 на 71. Увеличивает EIP на 7.
Устанавливает PF на 1.

```
dec byte ptr ds:[403033]
```

КОП:	FE0D 33304000
------	---------------

EAX=	0019FFCC	EBX=	0031A000	ECX=	656C626D	EDX=	00401000
ESP=	0019FF78	EBP=	0019FF84	ESI=	65737341	EDI=	00401000
EIP=	00401030						
ZF=	1	PF=	1	AF=	0		
OF=	0	SF=	0	DF=	0		
CF=	0	TF=	0	IF=	1		

Уменьшает ячейку с адресом 403032 на 1. Увеличивает EIP на 6.
Устанавливает ZF (флаг нуля) на 1.

```
push lab1.40302A
```

КОП:	68 2A304000
------	-------------

EAX=	0019FFCC	EBX=	0031A000	ECX=	656C626D	EDX=	00401000
ESP=	0019FF74	EBP=	0019FF84	ESI=	65737341	EDI=	00401000
EIP=	00401035						
ZF=	1	PF=	1	AF=	0		
OF=	0	SF=	0	DF=	0		
CF=	0	TF=	0	IF=	1		

Кладёт в стек значение из ячейки с адресом 40302A. Увеличивает EIP на 5.

```
call dword ptr ds:[<&puts>]
```

КОП:	FF15 08204000
------	---------------

EAX=	00000000	EBX=	0031A000	ECX=	01D45822	EDX=	00000000
ESP=	0019FF74	EBP=	0019FF84	ESI=	65737341	EDI=	00401000
EIP=	0040103B						
ZF=	1	PF=	1	AF=	0		
OF=	0	SF=	0	DF=	0		
CF=	0	TF=	0	IF=	1		

Вызов puts. Увеличивает EIP на 6.

```
add esp, 4
```

КОП:	83С4 04
------	---------

EAX=	00000000	EBX=	0031A000	ECX=	01D45822	EDX=	00000000
ESP=	0019FF78	EBP=	0019FF84	ESI=	65737341	EDI=	00401000
EIP=	0040103E						
ZF=	1	PF=	1	AF=	0		
OF=	0	SF=	0	DF=	0		
CF=	0	TF=	0	IF=	1		

Вызов puts. Увеличивает EIP на 3.

```
call dword ptr
ds:[<& getch>]
```

КОП:	FF15 0C204000
------	---------------

EAX=	0000000D	EBX=	0031A000	ECX=	C1B64159	EDX=	0019FDD8
ESP=	0019FF78	EBP=	0019FF84	ESI=	65737341	EDI=	00401000
EIP=	00401044						
ZF=	1	PF=	1	AF=	0		
OF=	0	SF=	0	DF=	0		
CF=	0	TF=	0	IF=	1		

Вызов `getch`. Увеличивает EIP на 6.

push 0						КОП:	6A 00				
EAX=	0000000D		EBX=	0031A000		ECX=	C1B64159		EDX=	0019FDD8	
ESP=	0019FF74		EBP=	0019FF84		ESI=	65737341		EDI=	00401000	
EIP=	00401046										
ZF=	1	PF=	1	AF=	0						
OF=	0	SF=	0	DF=	0						
CF=	0	TF=	0	IF=	1						
Кладёт в стек 0. Увеличивает EIP на 2.											

call <JMP.&ExitProcess>						КОП:	E8 01000000		
EAX=	0000000D		EBX=	0031A000		ECX=	C1B64159	EDX=	0019FDD8
ESP=	0019FF74		EBP=	0019FF84		ESI=	65737341	EDI=	00401000
EIP=	0040104B								
ZF=	1	PF=	1	AF=	0				
OF=	0	SF=	0	DF=	0				
CF=	0	TF=	0	IF=	1				
Вызывает выход из программы. Увеличивает EIP на 6. Конец программы.									

Вывод: в ходе лабораторной работы получены навыки создания простейших ассемблерных программ с использованием пакета masm32, получены навыки пользования отладчиком x32dbg.