Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ «САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

**Лабораторная работа №7 «Вывод цифры на экран»**

Выполнил: студент 2 курса 241 группы  
направления [010500.62](http://www.sgu.ru/education/courses/010500-62-matematicheskoe-obespechenie-i)Математическое обеспечение и администрирование информационных систем (профиль Параллельное программирование)

факультета компьютерных наук и информационных технологий

Акимов Артемий Андреевич

Саратов 2014

**Задание 1.1**: Первая цифра задана в AX, вторая цифра задана в BX. Написать программу, которая выводит в одну строку первую цифру, пробел, вторую цифру.

**Задание 1.2**: Первая цифра задана в AX, вторая цифра задана в BX. Написать программу, которая выводит в одну строку первую цифру (AX), пробел, вторую цифру (BX). Далее совершает обмен значений регистров AX и BX и снова в новой строке на экране выводит в одну строку первую цифру (AX), пробел, вторую цифру (BX). Обмен совершить без использования дополнительной памяти, регистров.

**Тексты программ**

**Программа 1:**

.model small ;Модель памяти SMALL использует сегменты размером не более 64Кб

.stack 100h ;Сегмент стека размером 100h (256 байт)

.data ;Начало сегмента данных

;Определение переменных

.code ;Начало сегмента кода

start:

mov AX, 7

mov BX, 8

mov AH, 02h

mov DX, AX

call return

int 21h

mov AX, 20h

mov AH, 02h

mov DX, AX

int 21h

mov AH, 02h

mov DX, BX

call return

int 21h

;Точка входа в программу start

;Предопределенная метка @data обозначает

;адрес сегмента данных в момент запуска программы,

mov AX, @data ;который сначала загрузим в AX,

mov DS, AX ;а затем перенесем из AX в DS

mov AX,4C00h

int 21h

return proc

add DX, 30h

ret

endp

end start

**Программа 2:**

.model small ;Модель памяти SMALL использует сегменты размером не более 64Кб

.stack 100h ;Сегмент стека размером 100h (256 байт)

.data ;Начало сегмента данных

;Определение переменных

.code ;Начало сегмента кода

start:

mov AX, 7

mov BX, 8

push AX

mov AH, 02h

mov DX, AX

call return

int 21h

mov AH, 02h

mov DX, 20h

int 21h

mov AH, 02h

mov DX, BX

call return

int 21h

mov AH, 02h

mov DX, 10

int 21h

mov AH, 02h

mov DX, 13

int 21h

pop AX

XCHG AX, BX

mov DX, AX

mov AH, 02h

call return

int 21h

mov AX, 20h

mov AH, 02h

mov DX, AX

int 21h

mov AH, 02h

mov DX, BX

call return

int 21h

;Точка входа в программу start

;Предопределенная метка @data обозначает

;адрес сегмента данных в момент запуска программы,

mov AX, @data ;который сначала загрузим в AX,

mov DS, AX ;а затем перенесем из AX в DS

mov AX,4C00h

int 21h

return proc

add DX, 30h

ret

endp

end start

**Текст командного файла make.bat**

cls

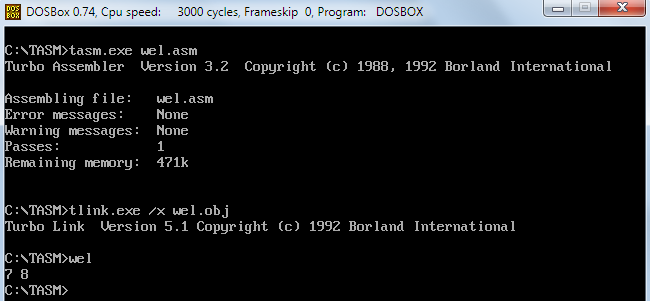
tasm.exe %1.asm

tlink.exe /x %1.obj

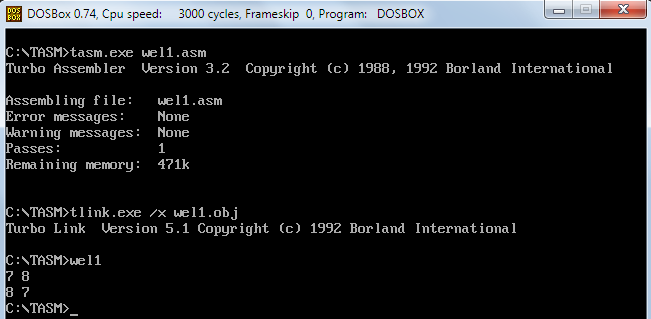
%1

**Результат выполнения команды** *make*

Программа 1:



Программа 2:



**Задание 2:** Заполните таблицы трассировки для программ 1.1 и 1.2.

Трассировка программы 1:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Шаг | Машинный код | Команда | Регистры | | | | | | | | | Флаги |
| AX | BX | CX | DX | SP | DS | SS | CS | IP | CZSOPAID |
| 1 | B80700 | mov ax, 0007 | 0000 | 0000 | 0000 | 0000 | 0100 | 489D | 48B0 | 48AD | 0000 | 00000010 |
| 2 | BB0800 | mov bx, 0008 | 0007 | 0000 | 0000 | 0000 | 0100 | 489D | 48B0 | 48AD | 0003 | 00000010 |
| 3 | B402 | mov ah, 02 | 0007 | 0008 | 0000 | 0000 | 0100 | 48BD | 48B0 | 48AD | 0006 | 00000010 |
| 4 | 8BD0 | mov dx ,ax | 0207 | 0008 | 0000 | 0000 | 0100 | 48BD | 48B0 | 48AD | 0008 | 00000010 |
| 5 | E81E00 | call 002B | 0207 | 0008 | 0000 | 0207 | 0100 | 48BD | 48B0 | 48AD | 000A | 00000010 |
| 6 | CD21 | int 21 | 0207 | 0008 | 0000 | 0237 | 0100 | 48BD | 48B0 | 48AD | 000D | 00000010 |
| 7 | B82000 | mov ax, 0020 | 0237 | 0008 | 0000 | 0237 | 0100 | 48BD | 48B0 | 48AD | 000F | 00000010 |
| 8 | B402 | mov ah, 02 | 0020 | 0008 | 0000 | 0237 | 0100 | 48BD | 48B0 | 48AD | 0012 | 00000010 |
| 9 | 8BD0 | mov dx, ax | 0220 | 0008 | 0000 | 0237 | 0100 | 48BD | 48B0 | 48AD | 0014 | 00000010 |
| 10 | CD21 | int 21 | 0220 | 0008 | 0000 | 0220 | 0100 | 48BD | 48B0 | 48AD | 0016 | 00000010 |
| 11 | B402 | mov ah, 02 | 0220 | 0008 | 0000 | 0220 | 0100 | 48BD | 48B0 | 48AD | 0018 | 00000010 |
| 12 | 8BD3 | mov dx,bx | 0220 | 0008 | 0000 | 0220 | 0100 | 48BD | 48B0 | 48AD | 001A | 00000010 |
| 13 | E80C00 | call 002B | 0220 | 0008 | 0000 | 0008 | 0100 | 48BD | 48B0 | 48AD | 001C | 00000010 |
| 14 | CD21 | int 21 | 0220 | 0008 | 0000 | 0038 | 0100 | 48BD | 48B0 | 48AD | 001F | 00000010 |
| 15 | B8B048 | mov ax, 48B0 | 0238 | 0008 | 0000 | 0038 | 0100 | 48BD | 48B0 | 48AD | 0021 | 00000010 |
| 16 | 8ED8 | mov ds, ax | 48B0 | 0008 | 0000 | 0038 | 0100 | 48BD | 48B0 | 48AD | 0024 | 00000010 |
| 17 | B8004C | mov ax, 4C00 | 48B0 | 0008 | 0000 | 0038 | 0100 | 48B0 | 48B0 | 48AD | 0026 | 00000010 |
| 18 | CD21 | int 21 | 4C00 | 0008 | 0000 | 0038 | 0100 | 48B0 | 48B0 | 48AD | 0029 | 00000010 |
|  |  |  | 0192 | 000c | f70d | 0999 | 0106 | 20f1 | 0192 | 0000 | 0000 | 10100011 |

Трассировка программы 2:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Шаг | Машинный код | Команда | Регистры | | | | | | | | | Флаги |
| AX | BX | CX | DX | SP | DS | SS | CS | IP | CZSOPAID |
| 1 | B80700 | mov ax, 0007 | 0000 | 0000 | 0000 | 0000 | 0100 | 489D | 48B3 | 48AD | 0000 | 00000010 |
| 2 | BB0800 | mov bx, 0008 | 0007 | 0008 | 0000 | 0000 | 0100 | 489D | 48B3 | 48AD | 0003 | 00000010 |
| 3 | 50 | push ax | 0007 | 0008 | 0000 | 0000 | 0100 | 489D | 48B3 | 48AD | 0006 | 00000010 |
| 4 | B402 | mov ah, 02 | 0007 | 0008 | 0000 | 0000 | 00FE | 489D | 48B3 | 48AD | 0007 | 00000010 |
| 5 | 8BD0 | mov dx, ax | 0207 | 0008 | 0000 | 0000 | 00FE | 489D | 48B3 | 48AD | 0009 | 00000010 |
| 6 | E84700 | call 0055 | 0207 | 0008 | 0000 | 0207 | 00FE | 489D | 48B3 | 48AD | 000B | 00000010 |
| 7 | CD21 | int 21 | 0207 | 0008 | 0000 | 0237 | 00FE | 489D | 48B3 | 48AD | 000E | 00000010 |
| 8 | B402 | mov ah, 02 | 0237 | 0008 | 0000 | 0237 | 00FE | 489D | 48B3 | 48AD | 0010 | 00000010 |
| 9 | BA2000 | mov dx, 0020 | 0237 | 0008 | 0000 | 0237 | 00FE | 489D | 48B3 | 48AD | 0012 | 00000010 |
| 10 | CD21 | int 21 | 0237 | 0008 | 0000 | 0020 | 00FE | 489D | 48B3 | 48AD | 0015 | 00000010 |
| 11 | B402 | mov ah, 02 | 0220 | 0008 | 0000 | 0020 | 00FE | 489D | 48B3 | 48AD | 0017 | 00000010 |
| 12 | 8BD3 | mov dx, bx | 0220 | 0008 | 0000 | 0020 | 00FE | 489D | 48B3 | 48AD | 0019 | 00000010 |
| 13 | E83700 | call 055 | 0220 | 0008 | 0000 | 0008 | 00FE | 489D | 48B3 | 48AD | 001B | 00000010 |
| 14 | CD21 | int 21 | 0220 | 0008 | 0000 | 0038 | 00FE | 489D | 48B3 | 48AD | 001E | 00000010 |
| 15 | B402 | mov ah, 02 | 0238 | 0008 | 0000 | 0038 | 00FE | 489D | 48B3 | 48AD | 0020 | 00000010 |
| 16 | BA0A00 | mov dx, 000A | 0238 | 0008 | 0000 | 0038 | 00FE | 489D | 48B3 | 48AD | 0022 | 00000010 |
| 17 | CD21 | int 21 | 0238 | 0008 | 0000 | 000A | 00FE | 489D | 48B3 | 48AD | 0025 | 00000010 |
| 18 | B402 | mov ah, 02 | 020A | 0008 | 0000 | 000A | 00FE | 489D | 48B3 | 48AD | 0027 | 00000010 |
| 19 | BA0D00 | mov dx, 000D | 020A | 0008 | 0000 | 000A | 00FE | 489D | 48B3 | 48AD | 0029 | 00000010 |
| 20 | CD21 | int 21 | 020A | 0008 | 0000 | 000D | 00FE | 489D | 48B3 | 48AD | 002C | 00000010 |
| 21 | 58 | pop ax | 020D | 0008 | 0000 | 000D | 00FE | 489D | 48B3 | 48AD | 002E | 00000010 |
| 22 | 93 | xchg bx, ax | 0007 | 0008 | 0000 | 000D | 0100 | 489D | 48B3 | 48AD | 002F | 00000010 |
| 23 | 8BD0 | mov dx, ax | 0008 | 0007 | 0000 | 000D | 0100 | 489D | 48B3 | 48AD | 0030 | 00000010 |
| 24 | B402 | mov ah, 02 | 0008 | 0007 | 0000 | 0008 | 0100 | 489D | 48B3 | 48AD | 0032 | 00000010 |
| 25 | E81E00 | call 0055 | 0208 | 0007 | 0000 | 0008 | 0100 | 489D | 48B3 | 48AD | 0034 | 00000010 |
| 26 | CD21 | int 21 | 0208 | 0007 | 0000 | 0038 | 0100 | 489D | 48B3 | 48AD | 0037 | 00000010 |
| 27 | B82000 | mov ax, 0020 | 0238 | 0007 | 0000 | 0038 | 0100 | 489D | 48B3 | 48AD | 0039 | 00000010 |
| 28 | B402 | mov ah, 02 | 0020 | 0007 | 0000 | 0038 | 0100 | 489D | 48B3 | 48AD | 003C | 00000010 |
| 29 | 8BD0 | mov dx, ax | 0220 | 0007 | 0000 | 0038 | 0100 | 489D | 48B3 | 48AD | 003E | 00000010 |
| 30 | CD21 | int 21 | 0220 | 0007 | 0000 | 0220 | 0100 | 489D | 48B3 | 48AD | 0040 | 00000010 |
| 31 | B402 | mov ah, 02 | 0220 | 0007 | 0000 | 0220 | 0100 | 489D | 48B3 | 48AD | 0042 | 00000010 |
| 32 | 8BD3 | mov dx, bx | 0220 | 0007 | 0000 | 0220 | 0100 | 489D | 48B3 | 48AD | 0044 | 00000010 |
| 33 | E80C00 | call 0055 | 0220 | 0007 | 0000 | 0007 | 0100 | 489D | 48B3 | 48AD | 0046 | 00000010 |
| 34 | CD21 | int 21 | 0220 | 0007 | 0000 | 0037 | 0100 | 489D | 48B3 | 48AD | 0049 | 00000010 |
| 35 | B8B348 | mov ax, 48B3 | 0237 | 0007 | 0000 | 0037 | 0100 | 489D | 48B3 | 48AD | 004B | 00000010 |
| 36 | 8BD8 | mov ds, ax | 48B3 | 0007 | 0000 | 0037 | 0100 | 489D | 48B3 | 48AD | 004E | 00000010 |
| 37 | B8004C | mov ax, 4C00 | 48B3 | 0007 | 0000 | 0037 | 0100 | 48B3 | 48B3 | 48AD | 0050 | 00000010 |
| 38 | CD21 | int 21 | 4C00 | 0007 | 0000 | 0037 | 0100 | 48B3 | 48B3 | 48AD | 0053 | 00000010 |
|  |  |  | 0192 | 000C | F709 | 0999 | 0106 | 20F1 | 0192 | 0000 | 0000 | 10100011 |

**Ответы на контрольные вопросы**

1. *Объясните назначение процедуры. Когда процедура определяется как near? Когда процедура определяется как far? Как определяются начало и конец процедуры?*

Все современные программы разрабатываются по модульному принципу – программа обычно состоит из одной или нескольких небольших частей, называемых процедурами, и одной главной программы, которая вызывает эти процедуры на выполнение, передавая им управление процессором. После завершения работы процедуры возвращают управление главной программе и выполнение продолжается с команды, следующей за командой вызова подпрограммы.

Язык программирования ассемблера поддерживает применение процедур двух типов – ближнего (near) и дальнего (far).

Процедуры ближнего типа должны находиться в том же сегменте, что и вызывающая программа. Дальний тип процедуры означает, что к ней можно обращаться из любого другого кодового сегмента.

Описание процедуры имеет следующий синтаксис:

<имя\_процедуры> PROC <параметр>  
<тело\_процедуры>   
RET ;Возврат из подпрограммы в точку вызова  
<имя\_процедуры> ENDP

Начало процедуры определяется словом PROC, конец процедуры словом ENDP.

1. *Ваша программа состоит из главной процедуры и процедур-подпрограмм. Каким может быть взаимное расположение главной процедуры и подпрограмм?*

В общем случае, размещать подпрограмму в теле программы можно где угодно, но при этом следует помнить, что сама по себе подпрограмма выполняться не должна, а должна выполняться лишь при обращении к ней. Поэтому подпрограммы принято размещать либо в конце сегмента кода, после команд завершения программы, либо в самом начале сегмента кода, перед точкой входа в программу. В больших программах подпрограммы нередко размещают в отдельном кодовом сегменте.

1. *Дайте определения понятиям точка возврата и адрес возврата.*

 После завершения работы процедуры возвращают управление главной программе и выполнение продолжается с команды, следующей за командой вызова подпрограммы (точка возврата).

При вызове процедуры в стеке сохраняется адрес возврата в вызывающую программу:

- при вызове ближней процедуры – слово, содержащее смещение точки вызова относительно текущего кодового сегмента;

- при вызове дальней процедуры – слово, содержащее адрес сегмента, в котором расположена точка возврата, и слово, содержащее смещение точки возврата в этом сегменте.

1. *Используя данные из таблиц трассировки, объясните из чего состоит код команды call. Что делает процессор при выполнении команды call?*
2. *Какой машинный код имеет команда ret в зависимости от того является она командой retn или retf? Что делает процессор при выполнении команды ret (retn или retf)?*

Близкий возврат (машинный код СЗ) — возврат управления вызывающей процедуре в пределах текущего сегмента кода (внутрисегментный возврат). При этом из вершины стека выталкивается значение в регистр EIP/IP. Регистр CS не изменяется. Процессор продолжает выполнение кода в том же сегменте, но по другому смещению.

Дальний возврат (машинный код СВ) — возврат управления вызывающей процедуре, расположенной в отличном от текущего сегменте кода (межсегментный возврат). При этом из вершины стека последовательно выталкиваются значения в регистры EIP/IP и CS. Процессор продолжает выполнение кода в другом сегменте.

1. *Какими способами можно завершить работу программы?*

Директива END указывает на конец программы, наличие в директиве END параметра MAIN означает, что процедура MAIN будет использоваться в качестве точки входа при выполнении программы. Вместо MAIN можно использовать любое другое имя, но оно должно быть одно и то же для PROC, ENDP и END.

С помощью прерывания INT 20h или функции 0 прерывания INT 21h обычно завершают свою работу com-программы. Перед завершением работы программы регистр CS должен указывать на PSP .

Более удобна функция 4Ch прерывания INT 21h которую можно использовать с любым содержимым регистров.

Существуют еще способы завершения работы программы, при которых программа (или ее часть) остается резидентной в памяти. Это вызов прерывания INT 27h или функции 31h прерывания INT 21h .