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 числа 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**: В регистре AX задано число от 0 до 65535. Выведите это число на экран.

**Задание 1.2**: Используя 32-битные регистры процессора, напишите программу, выводящую на экран число 65536.

**Тексты программ**

**Программа 1:**

.model small

.stack 100h

.386

.data

.code

start:

mov AX, @data ; значение физического адреса сегмента данных

mov DS, AX

mov AX, 65533 ; Заносим число в регистр AX

mov BX, 10 ; Заносим основание системы счисления в регистр BX

mov CX, 0 ; Номер младшего разряда числа

M1:

mov DX, 0 ; Подготавливаем регистр DX для записи очередной цифры числа

div BX ; Выполняем деление AX на BX

inc CX

push DX ; Сохраняем содержимое DX в стеке

cmp AX, 0 ; если неполное частное равно 0, то

jnz M1

; иначе

M2:

pop DX

call print

loop M2

mov AX,4C00h ; Завершение программы

int 21h

print proc ; Процедура print вывода на экран одной цифры

push AX

mov AH,02h

add DL, 30h

int 21h

pop AX

ret

print endp

end start

**Программа 2:**

.model small

.stack 100h

.386

.data

.code

start:

mov AX, @data ; значение физического адреса сегмента данных

mov DS, AX

mov EAX, 65536 ; Заносим число в регистр EAX (вместо АХ)

mov BX, 10 ; Заносим основание системы счисления в регистр BX

mov CX, 0 ; Номер младшего разряда числа

M1:

mov DX, 0 ; Подготавливаем регистр DX для записи очередной цифры числа

div EBX ; Выполняем деление ЕAX на ЕBX

inc CX

push DX ; Сохраняем содержимое DX в стеке

cmp EAX, 0 ; если неполное частное равно 0, то

jnz M1

; иначе

M2:

pop DX

call print

loop M2

mov AX,4C00h ; Завершение программы

int 21h

print proc ; Процедура print вывода на экран одной цифры

push AX

mov AH,02h

add DL, 30h

int 21h

pop AX

ret

print endp

end start

**Текст командного файла make.bat**

cls

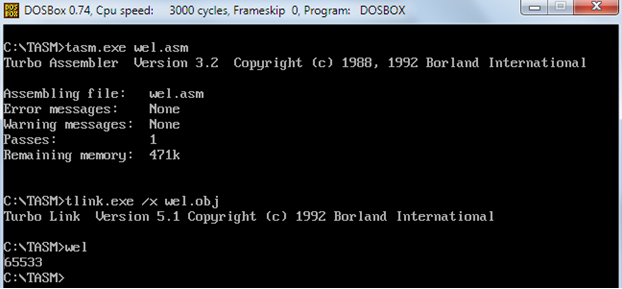
tasm.exe %1.asm

tlink.exe /x %1.obj

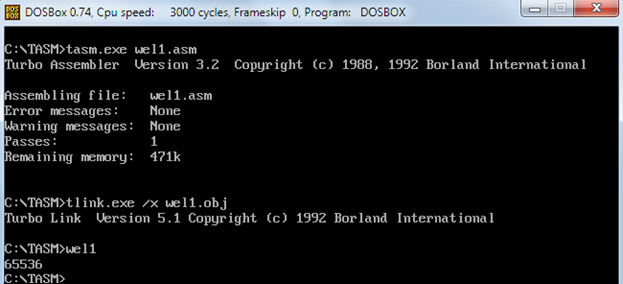
%1

**Результат выполнения команды** *make*

Программа 1:



Программа 2:



**Ответы на контрольные вопросы**

1. *По какой схеме выполняется операция деления?*

Существуют две формы команды деления – одна для двоичных чисел без знака DIV , а вторая для чисел со знаком – IDIV. Любая форма деления может работать с байтами и словами. Один из операндов (делимое) всегда в два раза длиннее операнда делителя. Ниже приведены схемы, иллюстрирующие команды деления.

Байтовая команда делит 16-битовое делимое на 8-битовый делитель. Делимое находится в регистре AX . В результате деления получается два числа: частное помещается в регистр AL , а остаток – в AH .

Команда, работающая со словами, делит 32-битовое делимое на 16-битовый делитель. Делимое находится в паре регистров DX : AX, причем регистр DX содержит старшую значимую часть, а регистр AX – младшую. Команда деления помещает частное в регистр AX, а остаток в DX .

1. *Как формируется машинный код для команд безусловного перехода типа SHORT, NEAR, FAR?*

Команда сравнения CMP сравнивает два числа, вычитая второе из первого. Основное назначение команды CMP – это организация ветвлений (условных переходов) в ассемблерных программах.

Безусловный переход – это переход, который передает управление без сохранения информации возврата всякий раз, когда выполняется. Ему соответствует команда JMP.

Если заранее известно, что переход вперед делается на место, лежащее в диапазоне 128 байт от текущего места, можно использовать команду JMP SHORT LABEL. Атрибут SHORT заставляет Ассемблер сформировать короткую форму команды перехода, даже если он еще не встретил метку LABEL. Представление машинного кода: *EB+смещение*, где смещение занимает 1 байт.

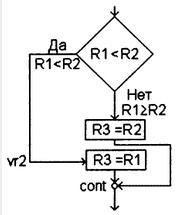
Переход типа *NEAR* (ближний) – если адрес перехода находится в том же сегменте памяти, что и команда JMP. Представление машинного кода: *E9 + смещение*, где смещение 2 байта (для 16-бит) и 4 байта (для 32-бит).

Переход типа *FAR*(дальний) – если адрес перехода находится в другом сегменте, может выполняться переход в тот же самый сегмент, если в сегментной части операнда указано число, совпадающее с текущем значением CS. Представление машинного кода: *EA + полный указатель* (сегмент + адрес в этом сегменте), Полный указатель занимает 4 байта (для 16-бит), 6 байт (для 32-бит).

1. *Каковы условия перехода для чисел со знаком и для чисел без знака?*

Арифметическое сравнение со знаком. Проверяются условия: меньше (JL), меньше или равно (JLE), больше (JG), больше или равно (JGE). Эти команды проверяют одновременно три флага: знака, переполнения и нуля.

Арифметическое сравнение без знака. Здесь также существует 4 возможных соотношения между операндами: меньше (JB), меньше или равно (JBE), больше (JA), больше или равно (JAE). Учитываются только два флага. Флаг переноса показывает какое из двух чисел больше. Флаг нуля определяет равенство.

1. *С помощью команд условного и безусловного перехода выполнить программную реализацию алгоритма ветвления для определения наименьшего числа из двух заданных. Алгоритм изображен в виде блок-схемы, приведенной на рисунке.*

R1 - первое число хранится в регистре AX; R2 - второе число хранится в регистре BX;R3 - результат заносится в регистр DX; vr2, cont - метки команд.

.model small

.stack 100h

.386

.data

.code

start:

mov AX, @data

mov DS, AX

mov AX, 10

mov BX, 25

cmp AX, BX

jb vr2

mov DX, BX

jmp cont

vr2:

mov DX, AX

jmp cont

cont:

mov AX,4C00h

int 21h

end start

1. *Какие действия выполняют команды организации цикла?*

Все команды цикла используют регистр CX в качестве счетчика цикла. Простейшая из них – команда LOOP. Она в конце каждой итерации уменьшает содержимое CX на 1 и передает управление на метку (указанную в команде), если содержимое CX не равно 0. Если вычитание 1 из CX привело к нулевому результату, выполняется следующая команда.

Команда LOOPNE (цикл пока не равно) выходит из цикла, если установлен флаг нуля или если в регистре CX получился 0. Команда LOOPE (цикл пока равно) выполняет обратную к описанной проверку флага нуля: цикл здесь завершается, если регистр CX достиг 0 или если не установлен флаг 0.

1. *Какой диапазон образуют знаковые числа, которые можно умножать на 2 путем сдвига в одном байте?*

Путем сдвига в одном байте можно реализовать умножение на 2 знаковых чисел в диапазоне от -64 до 63 и беззнаковых чисел, меньших 128.

1. *Как с помощью команды сдвига проверить содержимое регистра AH на четность?*

Необходимо произвести сдвиг вправо(SHR или SAR) на 1 над регистром AH, после проверить состояние регистра CF. Если CF=1, то содержимое регистра AH не четное, в противном случае четное.