

МИНОБРНАУКИ РОССИИ

Орский гуманитарно-технологический институт (филиал)  
федерального государственного бюджетного образовательного учреждения  
высшего образования «Оренбургский государственный университет»  
(Орский гуманитарно-технологический институт (филиал) ОГУ)

Кафедра программного обеспечения

Методические указания  
для обучающихся по освоению дисциплины  
«Б.1.В.ОД.9 Вычислительные системы, сети и телекоммуникации»

Уровень высшего образования

БАКАЛАВРИАТ

Направление подготовки

09.03.03 Прикладная информатика

(код и наименование направления подготовки)

Прикладная информатика в экономике

(наименование направленности (профиля) образовательной программы)

Тип образовательной программы

Программа академического бакалавриата

Квалификация

Бакалавр

Форма обучения

Очная

Год начала реализации программы (набора)

2014, 2015, 2016

г. Орск 2017

Методические указания для обучающихся по освоению дисциплины «Б.1.В.ОД.9 Вычислительные системы, сети и телекоммуникации» предназначены для обучающихся очной формы обучения направления подготовки 09.03.03 Прикладная информатика, профиля «Прикладная информатика в экономике».

Составитель \_\_\_\_\_ В.Н. Муллабаев

Методические указания рассмотрены и одобрены на заседании кафедры программного обеспечения, протокол № 10 от «07» июня 2017 г.

Заведующий кафедрой программного обеспечения

\_\_\_\_\_ Е.Е.Сурина

© Муллабаев В.Н., 2017  
© Орский гуманитарно-технологический институт (филиал) ОГУ, 2017

## **I Методические указания по проведению лекционных занятий**

Лекционные занятия в высшем учебном заведении являются основной формой организации учебного процесса и должны быть нацелены на выполнение ряда задач:

- ознакомить студентов со структурой дисциплины;
- изложить основной материал программы курса дисциплины;
- ознакомить с новейшими подходами и проблематикой в данной области;
- сформировать у студентов потребность к самостоятельной работе с учебной, нормативной и научной литературой.

Лекционное занятие представляет собой систематическое, последовательное, монологическое изложение преподавателем-лектором учебного материала, как правило, теоретического характера.

Цель лекции – организация целенаправленной познавательной деятельности студентов по овладению программным материалом учебной дисциплины.

Чтение курса лекций позволяет дать связанное, последовательное изложение материала в соответствии с новейшими данными науки, сообщить слушателям основное содержание предмета в целостном, систематизированном виде.

В ряде случаев лекция выполняет функцию основного источника информации, когда новые научные данные по той или иной теме не нашли отражения в учебниках.

Организационно-методической базой проведения лекционных занятий является рабочий учебный план направления подготовки. При подготовке лекционного материала преподаватель обязан руководствоваться учебными программами по дисциплинам кафедры, тематика и содержание лекционных занятий которых представлена в рабочих программах, учебно-методических комплексах.

При чтении лекций преподаватель имеет право самостоятельно выбирать формы и методы изложения материала, использовать различные технические средства обучения.

Рекомендации по работе студентов с конспектом лекций.

Изучение дисциплины студенту следует начинать с проработки рабочей программы, особое внимание, уделяя целям и задачам, структуре и содержанию курса.

При конспектировании лекций студентам необходимо излагать услышанный материал кратко, своими словами, обращая внимание, на логику изложения материала, аргументацию и приводимые примеры. Необходимо выделять важные места в своих записях. Если непонятны какие-либо моменты, необходимо записывать свои вопросы, постараться найти ответ на них самостоятельно. Если самостоятельно не удалось разобраться в материале, впоследствии необходимо либо на следующей лекции, либо на лабораторном занятии или консультации обратиться к ведущему преподавателю за разъяснениями.

Успешное освоение курса предполагает активное, творческое участие студента путем планомерной, повседневной работы. Лекционный материал следует просматривать в тот же день. Рекомендуемую дополнительную литературу следует прорабатывать после изучения данной темы по учебнику и материалам лекции.

Каждая тема имеет свои специфические термины и определения. Усвоение материала необходимо начинать с усвоения этих понятий. Если какое-либо понятие вызывает затруднения, необходимо посмотреть его суть и содержание в словаре (Интернете), выписать его значение в тетрадь для подготовки к занятиям.

При подготовке материала необходимо обращать внимание на точность определений, последовательность изучения материала, аргументацию, собственные примеры, анализ конкретных ситуаций. Каждую неделю рекомендуется отводить время для повторения пройденного материала, проверяя свои знания, умения и навыки по контрольным вопросам и тестам.



## 2 Методические указания по лабораторным и практическим работам

Изучение дисциплины «Б.1.В.ОД.9 Вычислительные системы, сети и телекоммуникации» предполагает посещение обучающимися не только лекций, но и лабораторных работ. Лабораторные работы со студентами предназначены для проверки усвоения ими теоретического материала дисциплины.

Основные цели лабораторных работ:

- закрепить основные положения дисциплины;
- проверить уровень усвоения и понимания студентами вопросов, рассмотренных на лекциях и самостоятельно изученных по учебной литературе;
- научить пользоваться нормативной и справочной литературой для получения необходимой информации о конкретных технологиях;
- оказать помощь в приобретении навыков расчета точностных характеристик;
- восполнить пробелы в пройденной теоретической части курса и оказать помощь в его усвоении.

Для контроля знаний, полученных в процессе освоения дисциплины на лабораторных занятиях обучающиеся выполняют задания реконструктивного уровня и комплексное практическое задание.

Целью выполнения задания реконструктивного уровня и комплексного задания студентами является систематизация, закрепление и расширение теоретических знаний, полученных в ходе изучения дисциплины.

Ниже приводятся общие методические указания, которые относятся к занятиям по всем темам:

- в начале каждого лабораторного занятия необходимо сформулировать цель, поставить задачи;
- далее необходимо проверить знания студентами лекционного материала по теме занятий;
- в процессе занятия необходимо добиваться индивидуальной самостоятельной работы студентов;
- знания студентов периодически контролируются путем проведения текущей аттестации (рубежного контроля), сведения о результатах которой доводятся до студентов и подаются в деканат;
- время, выделенное на отдельные этапы занятий, указанное в рабочей программе, является ориентировочным; преподаватель может перераспределить его, но должна быть обеспечена проработка в полном объеме приведенного в рабочей программе материала;
- на первом занятии преподаватель должен ознакомить студентов с правилами поведения в лаборатории и провести инструктаж по охране труда и по пожарной безопасности на рабочем месте;
- преподаватель должен ознакомить студентов со всем объемом лабораторных работ и требованиями, изложенными выше;
- преподаватель уделяет внимание оценке активности работы студентов на занятиях, определению уровня их знаний на каждом занятии.

На лабораторных работах решаются задачи из всех разделов изучаемой дисциплины.

## Лабораторная работа №1

### 2.1. Знакомство с инструментами для программирования на языке Ассемблер.

Для программирования на языке Ассемблер потребуются следующие программные модули:

- транслятор TASM.EXE – переводит исходный программный код в объектный код;
- компоновщик TLINK.EXE – преобразует объектный файл в исполнимый для последующего запуска и отладки;
- отладчик TD.EXE – позволяет отлаживать исполнимый двоичный файл в пошаговом режиме;
- эмулятор процессора Intel 8086 (эмулятор DOS-консоли) B32TOOLS.PIF– создает среду, где выполняются вышеперечисленные модули и пользовательские программы и получают конечный результат.

Все вышеназванные модули входят в пакет TASM.

Редактировать исходные тексты ассемблерных программ можно в любом текстовом редакторе. Желательно, чтобы редактор поддерживал кодировку CP866, для правильного отображения русских символов. Например Notepad++ или встроенный редактор файлового менеджера FAR. Если их нет, то можно на встроенном текстовом редакторе Windows Notepad. Тогда русские слова придется писать на латинице.

### 2.2. Написание и отладка первой ознакомительной программы.

Цель работы – ознакомление со структурой написания программы на языке Assembler x86.

#### Исходный текст программы.

```
text segment 'code'
assume cs:text, ds:data

stdout=1
cr=13
lf=10

myproc proc
    mov ax, data
    mov ds, ax
    mov ah, 40h
    mov bx, stdout
    mov cx, meslen
    mov dx, offset mes
    int 21h
    mov ax, 4c00h
    int 21h
myproc endp
text ends

data segment
mes db 'Программа .EXE стартовала',cr,lf
meslenequ $-mes
data ends
```



```

stack1 segment      para stack 'stack'
                db    128 dup (?)
stack1 ends
end    myproc

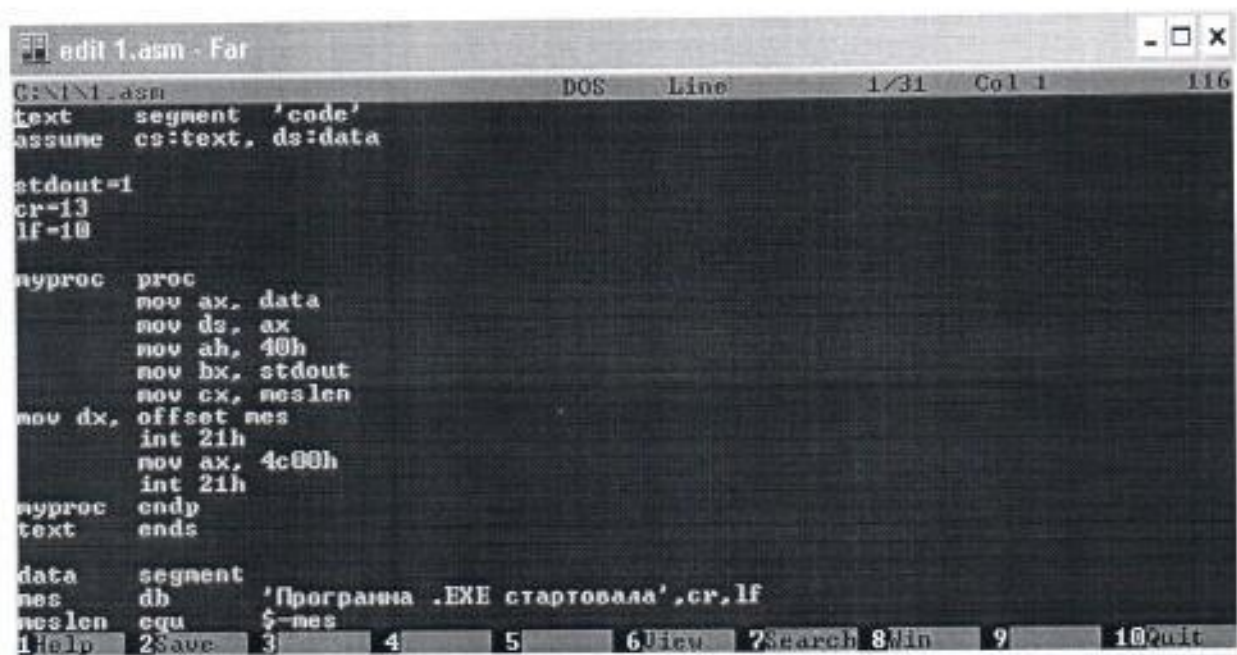
```

### Краткое описание ознакомительной программы

- Оператор TITLE задает текстовый заголовок, который будет выводиться на все страницы листинга
- Программа состоит из трех сегментов памяти: сегмента команд с именем text, сегмента данных с именем data и сегмента стека с именем stack1. Имена могут выбираться произвольно. Каждый сегмент открывается оператором SEGMENT и закрывается оператором ENDS.
- Слово 'code' в апострофах указывает класс сегмента – программный. Если в программе несколько сегментов одного класса то они загружаются в память последовательно один за другим.
- Оператор ASSUME указывает транслятору соответствие между сегментным регистром и сегментом. CS:TEXT – указывает на сегмент кода, DS:DATA – указывает, что регистр DS будет использоваться по умолчанию как регистр сегмента данных.
- Данная программа состоит из одной процедуры myproc, который открывается оператором proc и закрывается оператором endp.
- В начале программы инициализируется регистр DS. В него заносится адрес сегмента данных data. Инициализация сегментных регистров производится только через регистр ax.
- Далее в регистр ah заносится номер функции 40h – функция записи в файл.
- В регистр bx заносится номер дескриптора открытого файла. Это значение константы stdout=1, т.е. стандартное устройство вывода. В итоге, запись сообщения произведется не в реальный файл, а на экран компьютера.
- В регистр cx заносится длина выводимой строки meslen.
- В регистр dx заносится адрес выводимой строки (сообщения).
- Вызывается прерывание int 21h - которое запустит функцию 40h. В итоге строка «Программа .EXE стартовала» появится на экране.
- Далее снова вызывается программное прерывание int 21h, но с функцией 4Ch – корректное завершение программы
- На этом описание процедуры myproc завершается
- В сегменте данных находятся переменные, выводимая строка mes и его длина meslen.
- Строка описания сегмента стека содержит класс сегмента 'stack' и тип объединения stack.
- Для одномодульных программ тип объединения значения не имеет, но для сегмента стека обязательно указание типа stack.
- Текст программы заканчивается директивой END, завершающий трансляцию. В качестве операнда этой директивы указывается точка входа в главную процедуру, т.е. в процедуру myproc.

### Шаги работы по выполнению лабораторной работы №1

1. Набрать в текстовом редакторе исходный текст программы. Текст, набранный в редакторе FAR выглядит следующим образом:



```
edit 1.asm - Far
G:\NIN1.asm      DOS      Line      1/31      Col 1      116
text segment 'code'
assume cs:text, ds:data

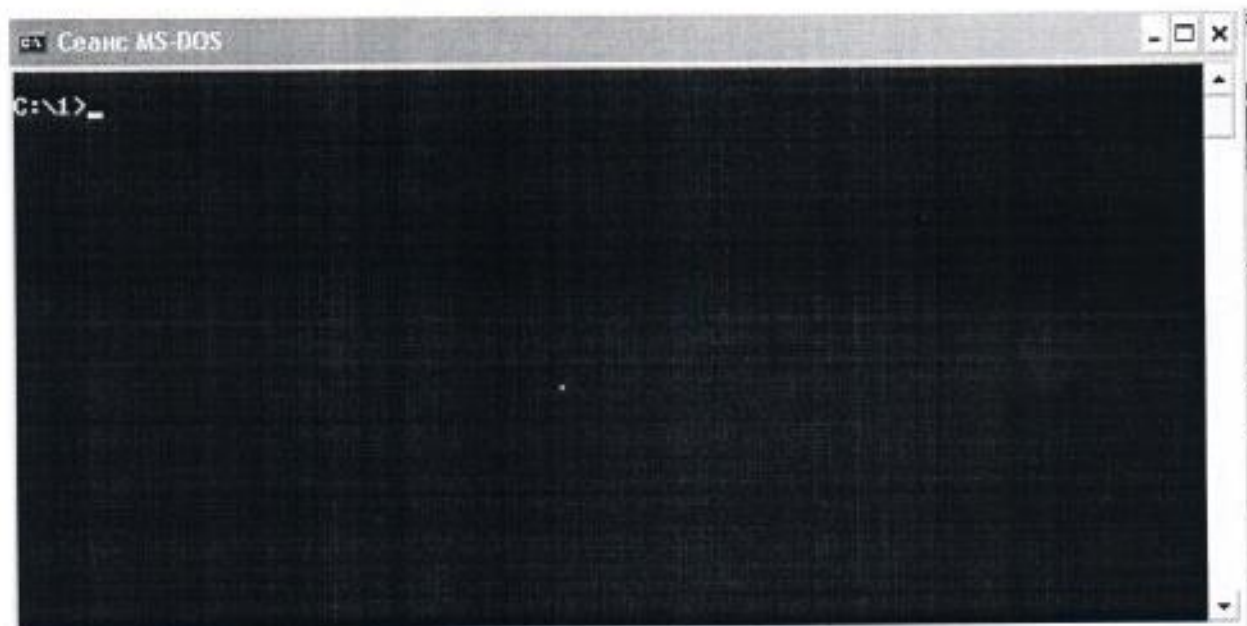
stdout=1
cr=13
lf=10

myproc proc
    mov ax, data
    mov ds, ax
    mov ah, 40h
    mov bx, stdout
    mov cx, meslen
    mov dx, offset mes
        int 21h
    mov ax, 4c00h
    int 21h
myproc endp
text ends

data segment
mes db 'Программа .EXE стартовала',cr,lf
meslen equ $-mes
1|help 2|save 3| 4| 5| 6|view 7|search 8|min 9| 10|quit
```

По клавише F2 сохраняем текст программы в файл под именем 1.asm.

2. Далее запускаем среду отладки и выполнения B32TOOLS.PIF. На рабочем столе Windows он может находиться под именем «MS DOS Prompt».



3. В появившемся черном окне последовательно выполняем следующие действия. Сначала, командой dir убеждаемся, что файл 1.asm нам доступен, далее выполняем трансляцию исходного кода в объектный код с помощью команды tasm. Иллюстрация приведена ниже:



```

Секс MS-DOS
C:\>dir
Том в устройстве C не имеет метки.
Серийный номер тома: 9CE7-E745

Содержимое папки C:\1
18.01.2020 15:44 <DIR> .
18.01.2020 15:44 <DIR> ..
18.01.2020 15:36          413 1.asm
                1 файл          413 байт
                2 папок      2 163 400 704 байт свободно

C:\>tasm 1.asm
Turbo Assembler Version 4.1 Copyright (c) 1988, 1996 Borland International

Assembling file: 1.asm
Error messages:  None
Warning messages: None
Passes:         1
Remaining memory: 448k

C:\>

```

4. Если при трансляции появятся ошибки, то необходимо вернуться в редактор, поправить и пересохранить исходный код и повторить пункт 3. Если ошибок нет, то появляется файл с объектным кодом 1.obj и выполняем работу далее.
5. Выполняем компоновку с помощью команды tlink. Выходом команды tlink является исполнимый файл 1.exe.

```

Секс MS-DOS
Turbo Assembler Version 4.1 Copyright (c) 1988, 1996 Borland International

Assembling file: 1.asm
Error messages:  None
Warning messages: None
Passes:         1
Remaining memory: 448k

C:\>dir
Том в устройстве C не имеет метки.
Серийный номер тома: 9CE7-E745

Содержимое папки C:\1
18.01.2020 15:44 <DIR> .
18.01.2020 15:44 <DIR> ..
18.01.2020 15:36          413 1.asm
18.01.2020 15:44          242 1.OBJ
                2 файл          655 байт
                2 папок      2 163 384 320 байт свободно

C:\>tlink 1.obj
Turbo Link Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International

```

6. Исполнимый файл 1.exe можно запустить и получить результат работы программы. Ниже приводится иллюстрация вышесказанного:



```

С:\>Сессия MS-DOS
2 папок 2 163 384 320 байт свободно
C:\>tlink 1.obj
Turbo Link Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International
C:\>dir
Том в устройстве C не имеет метки.
Серийный номер тома: 9CE7-E745

Содержимое папки C:\1
18.01.2020 15:46 <DIR>
18.01.2020 15:46 <DIR>
18.01.2020 15:36 413 1.asm
18.01.2020 15:46 571 1.EXE
18.01.2020 15:46 228 1.MAP
18.01.2020 15:44 242 1.OBJ
4 файлов 1 454 байт
2 папок 2 163 384 320 байт свободно

C:\>
C:\>1.exe
Программа .EXE стартовала
C:\>

```

### Вопросы для самостоятельной работы

- Из каких сегментов должна состоять любая программа написанная на языке Ассемблер и как они обозначаются.
- Как объявляются процедуры и какой директивой должен завершаться текст программы на языке Ассемблер.
- Перечислите перечень регистров и поясните их назначение в функции 40h программного прерывания 21h.
- Что находится в константе meslen и как она получилась.
- Каким образом происходит корректное завершение программы.

### Лабораторная работа №2

#### 2.3. Программирование на языке Ассемблер. Арифметические команды.

Используя лекционный материал повторить особенности работы арифметических команд процессора Intel 8086 с операндами типа BYTE и WORD. Это следующие команды:

- ADD – сложение;
- SUB – вычитание;
- ADC – сложение с добавлением значения бита переноса;
- SBB – вычитание с последующим вычитанием значения бита переноса;
- MUL – умножения без учета знака операндов;
- DIV – деление без учета знака операндов;
- IMUL – умножения с учетом знака операндов;
- IDIV – деление с учетом знака операндов;

Используя эти команды необходимо написать программу решающее следующее математическое выражение:

$$x = \frac{2a + bc}{d - 3}$$

Фрагмент программного кода, выполняющее поставленную задачу выглядит следующим образом:

```

MOV  AX, 2      ; Загрузка константы
IMUL A         ; DX:AX = A * 2
MOV  BX, DX
MOV  CX, AX     ; BX:CX = A * 2
MOV  AX, B
IMUL C         ; DX:AX = B * C
ADD  AX, CX
ADC  DX, BX     ; DX:AX = A * 2 + B * C
MOV  CX, D
SUB  CX, 3     ; CX = D - 3
IDIV CX        ; AX = (A*2 + B*C) / (D-3)
MOV  X, AX     ; Сохранение результата

```

Используя принципы построения ассемблерных программ из лабораторной работы №1 необходимо написать программу расчета вышеприведенного математического выражения и отладить её. Отладку провести с помощью отладчика TD.EXE.

#### 2.4. Варианты индивидуальных заданий:

1.  $x = \frac{a^2 + b^2}{c - d} + ab$

2.  $x = \frac{155d - c}{a + b} + c^2$

3.  $x = \frac{607c - ab}{b} + k$

4.  $y = \frac{138x + kz}{k - z} - x^2$

5.  $x = 705a + \frac{bc}{a - b}$

6.  $x = 811c - \frac{d^2}{c - d}$

7.  $z = cd + bc - \frac{c^2}{b - c}$

8.  $z = ab - 207c + \frac{bc}{c - d}$



$$9. \quad z = xy - 13d + \frac{138d - a}{b}$$

$$10. \quad k = \frac{207d - b^2}{205 - d} + ab$$

$$11. \quad k = \frac{c^2 - 155d}{a - b} + d^2$$

$$12. \quad z = \frac{a^2 + 255d}{ad - b} + d^2$$

### Лабораторная работа №3

#### 2.5. Программирование на языке Ассемблер. Стандартное устройство ввода.

Используя лекционный материал необходимо детально разобраться в принципе работы клавиатуры и взаимосвязь операционной системы (драйвера клавиатуры) с пользовательской программой. При нажатии на клавишу пользовательская программа получает ASCII-коды нажатых клавиш. Для получения числовых данных в переменные программы, и.п. значения переменных а, b, с и т.д. необходимо преобразовать строку ASCII-кодов получаемых при нажатии на клавиши в число. Это выполняет процедура ввода числа `inp`. Исходный код процедуры `inp` приведена ниже:

```

inp proc
push cx
push dx
push bx
    xor bx, bx
    xor ch, ch
    mov cl, 02h
go:
    mov ah, 01h
    int 21h
    inc ch
    cmp al, 30h
    jb all
    cmp al, 39h
    ja all
    sub al, 30h
    shl bx, 1
    mov dx, bx
    shl bx, cl
    add bx, dx
    add bl, al
    adc bh, 0
    jmp go
all:
    cmp ch, 1
    jne all1

```

```

        cmp al, '-'
        jne all1
        mov sign, 1
        jmp go
all1:
        mov ah, 02h
        mov dl, 10
        int 21h
        mov dl, 13
        int 21h
        cmp sign, 1
        jne all2
        neg bx
all2:
        mov sign, 0
        mov ax, bx
pop bx
pop dx
pop cx
ret
inp endp

```

Пример использования данной процедуры будет приведена ниже.

При выводе результата расчета на экран необходимо провести обратное преобразование числа в строку ASCII-символов. Эту задачу выполняет процедура `outp`. Исходный код процедуры `inp` приведена ниже:

```

outp proc
push ax
push bx
push cx
push dx
push bp
push si

test ax, 8000h
jz ou1
mov dl, '-'
push ax
mov ah, 02h
int 21h
pop ax
neg ax
ou1:
xor cx, cx
xor si, si
mov bx, 0ah
next:
xor dx, dx
div bx
push dx

```



```
inc cx
or ax,ax
jnz next
mov ah, 02h
nex:
pop dx
add dl, 30h
int 21h
loop nex
```

```
pop si
pop bp
pop dx
pop cx
pop bx
pop ax
ret
outp endp
```

Пример использования данной процедуры будет приведена ниже.

#### Лабораторная работа №4

##### **2.5. Программирование на языке Ассемблер. Итоговая программа по материалам лабораторных работ №1-3.**

**Цель работы:** Написать программу вычисления результата математического выражения средствами языка программирования Assembler.

**Задание:** Вычислить результат выражения  $(207*d - b^2)/(205-d) + a*b = x$

##### **Блок схемы алгоритма**

Процедура перевода двоичного числа в десятичное и запись его в строку представлена на рисунке 1.



Рисунок 1 – Блок-схема процедуры outstr

На рисунке 2 отображена процедура вычисления заданного арифметического выражения.



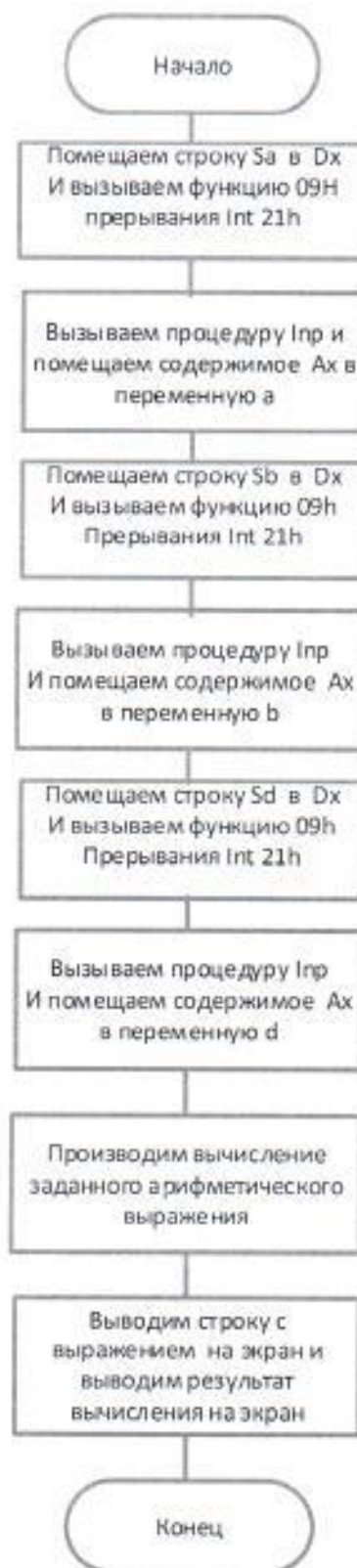


Рисунок 2 – Блок-схема процедуры расчета  $\text{trg}$

Процедура считывания данных с клавиатуры представлена на рисунке 3.





```

;Процедура вычисления арифметического выражения
prg proc
    mov ax,data
    mov ds,ax
;Считывание в переменную a данных с клавиатуры
    lea dx,sa
    mov ah,09h
    int 21h
    call inp
    mov a,ax
;Считывание в переменную b данных с клавиатуры
    lea dx,sb
    mov ah,09h
    int 21h
    call inp
    mov b,ax
;Считывание в переменную d данных с клавиатуры
    lea dx,sd
    mov ah,09h
    int 21h
    call inp
    mov d,ax
;Вычисление части выражения:b*b
    mov ax,b
    mov bx,b
    imul bx
;Вычисление части выражения:207*d
    push ax
    mov ax,207
    mov bx,d
    imul bx
;Вычисление части выражения:207*d - b*b
    pop bx
    mov dx,0
    mov cx,0
    sub ax,bx
    sbb dx,cx
    push ax
;Вычисление части выражения:205-d
    mov ax,205
    mov bx,d
    mov dx,0
    mov cx,0
    sub ax,bx
    sbb dx,cx
;Вычисление части выражения: (207*d - b^2)/205-d
    mov bx,ax
    pop ax
    cwd
    idiv bx
    push ax
;Вычисление части выражения:a*b
    mov ax,a
    mov bx,b
    imul bx
;Вычисление выражения: (207*d - b^2)/(205-d) +a*b=
    mov bx,ax
    pop ax
    mov dx,0
    mov cx,0
    add ax,bx
    adc dx,cx

```

```

    mov x,ax
;Вывод результата на экран
    call outstr
;Вывод оформления результата на экран
    lea dx,str1
    mov ah,09h
    int 21h
    mov dx, offset text1
    mov ah,09h
    int 21h
; Выход в ОС
    mov ah,4ch
    int 21h
mpc endp

;Процедура считывания числовых данных с клавиатуры
inp proc
    push cx
    push dx
    push bx
    xor bx, bx
;Вызываем процедуру считывания символа клавиши
    mov cl,02h
    mov ah,01h
go:
;Переводим полученный символ в двоичную систему счисления и записываем в регистр bx
    int 21h
    cmp al,30h
    jb all
    cmp al,39h
    ja all
    sub al,30h
    shl bx,1
    mov dx,bx
    shl bx,cl
    add bx,dx
    add bl,al
    adc bh,00h
    jmp go
;После того как введён не числовой символ записываем результат в регистр ax и сме-
щаем каретку на начало следующей строки
all:
    mov ah,02h
    mov dl,0Ah
    int 21h
    mov dl,0Dh
    int 21h
    mov ax,bx
    pop bx
    pop dx
    pop cx
    ret
inp endp

;Процедура преобразования двоичного числа в знаковое десятичное
outstr proc
    push ax
    push cx
    push dx
    push bx
    push si
    xor cx,cx

```

```

xor si,si
;Проверяем число <0
test ax,ax
jns o1
;Если отрицательное то в первый символ результирующего числа заносим знак минус
mov text1[si], '-'
inc si
neg ax
; Переводим число в десятичную систему счисления и полученные цифры заталкиваем в
стек
o1:
mov bx,000Ah
count1:
xor dx,dx
div bx
push dx
inc cx
or ax,ax
jnz count1
; После перевода всего числа посимвольно из стека заносим числа в результирующую
строку и в конце добавляем символ конца строки
next1:
pop dx
add dl,30h
mov text1[si],dl
inc si
loop next1
mov textlen, si
inc si
mov text1[si], '$'
pop si
pop bx
pop dx
pop cx
pop ax
ret
outstr endp
text ends
data segment
a dw ?
b dw ?
d dw ?
x dw ?
str1 db '(207*d -b^2)/(205-d) +a*b= ', '$'
sa db 'a:', '$'
sb db 'b:', '$'
sd db 'd:', '$'
textlen dw ?
text1 db 10 dup(?)
data ends
stack1 segment para stack 'stack1'
db 128 dup(?)
stack1 ends
end mpr

```

### Тестирование программы

После запуска программы необходимо ввести данные в переменные, после ввода последней переменной выводится результат вычисления.

Пример работы программы изображён на рисунке 4.



```
GUI Turbo Assembler x64
a:21
b:13
d:54
(207*d -b^2)/(205-d) +a*b= 964
Program successfully executed !
Press any key to continue.
```

Рисунок 4 – Пример работы программы расчёта арифметического выражения

### 3 Методические указания по самостоятельной работе

Для успешного освоения курса «Б.1.В.ОД.9 Вычислительные системы, сети и телекоммуникации» необходима самостоятельная работа. В настоящее время актуальными становятся требования к личным качествам современного студента – умению самостоятельно пополнять и обновлять знания, вести самостоятельный поиск необходимого материала, быть творческой личностью.

Самостоятельную работу по освоению дисциплины обучающимся следует начинать с изучения содержания рабочей учебной программы дисциплины, цели и задач, структуры и содержания курса, основной и дополнительной литературы, рекомендованной для самостоятельной работы.

Самостоятельная учебная деятельность является необходимым условием успешного обучения. Многие профессиональные навыки, способность мыслить и обобщать, делать выводы и строить суждения, выступать и слушать других, – все это развивается в процессе самостоятельной работы студентов.

Самостоятельная работа по освоению дисциплины включает:

- самостоятельное изучение разделов;
- самоподготовку (проработку и повторение лекционного материала и материала учебников и учебных пособий);
- подготовку к лабораторным работам;
- подготовку к рубежному и итоговому контролю.

Самостоятельная учебная работа – условие успешного окончания высшего учебного заведения. Она является равноправной формой учебных занятий, наряду с лекциями, семинарами, экзаменами и зачетами, но реализуемая во внеаудиторное время.

Эффективность аудиторных занятий во многом зависит от того, насколько умело студенты организуют в ходе них свою самостоятельную учебную познавательную деятельность. Такая работа также способствует самообразованию и самовоспитанию, осуществляемому в интересах повышения профессиональных компетенций, общей эрудиции и формировании личностных качеств.

Самостоятельная работа реализуется:

1. непосредственно в процессе аудиторных занятий – на лекциях, лабораторных занятиях, при проведении рубежного контроля;
2. в контакте с преподавателем вне рамок расписания – на консультациях по учебным вопросам, при ликвидации задолженностей, при выполнении индивидуальных заданий;
3. в библиотеке, дома, в общежитии, на кафедре при выполнении студентом учебных



задач.

В процессе проведения самостоятельной работы необходимо производить подбор литературных источников, научной периодической печати и т.д

#### **4 Методические указания по итоговому контролю**

Итоговый контроль знаний по дисциплине «Б.1.В.ОД.9 Вычислительные системы, сети и телекоммуникации» проводится в форме экзамена. Для подготовки к итоговому контролю знаний по дисциплине «Б.1.В.ОД.9 Вычислительные системы, сети и телекоммуникации» обучающиеся используют перечень вопросов, приведенный в фонде оценочных средств. Экзамен проводится в устной форме. В экзаменационный билет включен один теоретический вопрос. На подготовку студенту отводится 20-25 минут. На дифференцированном зачете ответы обучающегося оцениваются с учетом их полноты, правильности и аргументированности с учетом шкалы оценивания.

Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, использует в ответе профессиональные термины, правильно обосновывает принятое решение.

Оценка «хорошо» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов, владеет необходимыми навыками и приемами их выполнения.

Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала.

Оценка «неудовлетворительно» выставляется студенту за отсутствие знаний по дисциплине, представления по вопросу, непонимание материала по дисциплине, наличие коммуникативных «барьеров» в общении, отсутствие ответа на предложенный вопрос.

#### **5 Учебно-методическое обеспечение дисциплины**

##### **5.1 Основная литература**

1. **Архитектура ЭВМ и систем** [Электронный ресурс] / Ю.Ю. Громов [и др.]. - Тамбов: Издательство ФГБОУ ВПО «ТГТУ», 2012. - 200 с. - Режим доступа: <http://biblioclub.ru/index.php?page=book&id=277352>

2. **Таненбаум, Э. Архитектура компьютера** [Текст] / Э. Таненбаум, Т. Остин. - 6-е изд. - Москва : Питер, 2013. - 816 с. : ил. - (Классика computer science). - Алф. указ. : с. 791. - ISBN 978-0132916523 англ.

##### **5.2 Дополнительная литература**

1. **Кирнос В. Н. Введение в вычислительную технику: основы организации ЭВМ и программирование на Ассемблере. Учебное пособие** [Электронный ресурс] / Кирнос В. Н. - Эль Контекст, 2011. - Режим доступа: <http://biblioclub.ru/index.php?page=book&id=208652>

##### **5.3 Периодические издания**

1. Журнал «ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ».
2. Журнал «ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ».
3. Журнал «ВЕСТНИК КОМПЬЮТЕРНЫХ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ».
4. Журнал «ОТКРЫТЫЕ СИСТЕМЫ. СУБД».

#### **5.4 Интернет-ресурсы**

1. Степанов А.Н. Архитектура вычислительных систем и компьютерных сетей. – СПб.: Питер, 2007 [Электронный ресурс]. – Режим доступа: [http://tverhbk.ru/library/predmets/vichislitelnaa\\_tehnika/Arhitektura\\_vychislitelnyh\\_sistem\\_i\\_komputernyh\\_setej\\_2007.pdf](http://tverhbk.ru/library/predmets/vichislitelnaa_tehnika/Arhitektura_vychislitelnyh_sistem_i_komputernyh_setej_2007.pdf)
2. Толстобров А.П. Архитектура ЭВМ: Учебное пособие. - Воронеж: Изд-во ВГУ, 2004 [Электронный ресурс]. – Режим доступа: <http://window.edu.ru/resource/871/26871>