

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.ДВ.9.2 Машинно-зависимые языки программирования»

Уровень высшего образования

БАКАЛАВРИАТ

Направление подготовки

09.03.01 информатика и вычислительная техника

(код и наименование направления подготовки)

Программное обеспечение средств вычислительной техники и
автоматизированных систем

(наименование направленности (профиля) образовательной программы)

Тип образовательной программы

Программа академического бакалавриата

Квалификация

Бакалавр

Форма обучения

Очная

Год начала реализации программы (набора)

2014, 2015, 2016, 2017

г. Орск 2017

Методические указания для обучающихся по освоению дисциплины «Б.1.В.ДВ.9.2 Машинно-зависимые языки программирования» предназначены для обучающихся очной формы обучения направления подготовки 09.03.01 информатика и вычислительная техника, профиля «Программное обеспечение средств вычислительной техники и автоматизированных систем»

Составитель  В.Н. Муллабаев

Методические указания рассмотрены и одобрены на заседании кафедры программного обеспечения, протокол № 2 от «07» июня 2017 г.

Заведующий кафедрой программного обеспечения

 Е.Е. Сурина

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

1 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 к выполнению курсового проекта.

2.1. Введение.

Курсовой проект по дисциплине «Б.1.В.ДВ.9.2 Машинно-зависимые языки программирования» является самостоятельной научной работой студента в письменной форме по заданной теме. Материальная часть исследования предполагает создание компьютерной программы, выполняющей поставленную задачу или группу задач. Дисциплина «Б.1.В.ДВ.9.2 Машинно-зависимые языки программирования» предполагает написание компьютерной программы на машинно-зависимом языке, а именно язык программирования Ассемблер для процессора Intel 8086.

Для выполнения курсового проекта студент должен детально изучить архитектуру процессора Intel 8086, его систему команд и методы адресации. Также, для обеспечения интерфейса между программой и пользователем, необходимо изучить стандартные устройства ввода и вывода. Курсовой проект должен состоять из двух частей.

Первая часть – расчет математического выражения согласно полученному варианту.

Вторая часть – творческая. Создание программы анимации. Полученные результаты расчета математического выражения из первой части должны передвигаться по экрану по траектории согласно варианту задания разными цветами и по разным видеостраницам.

Для выполнения первой части задания студент должен знать ответы на следующие вопросы:

1. Какие регистры образуют группу "Регистры данных",
2. Какие регистры образуют группу "Регистры указатели".
3. Какие регистры образуют группу "Сегментные регистры".
4. Какие регистры образуют группу "Дополнительные регистры".
5. Перечислите и поясните специальные функции регистра AX.
6. Перечислите и поясните специальные функции регистра BX.
7. Перечислите и поясните специальные функции регистра CX.
8. Перечислите и поясните специальные функции регистра DX.
9. Какие функции выполняют регистры указатели.
10. Для чего служат сегментные регистры.
11. Что находится внутри сегментных регистров CS, DS, ES и SS и с какими областями памяти они ассоциируются.
12. Как происходит преобразование содержимого сегментного регистра и смещения в физический адрес ячейки памяти.
13. Перечислите флаги в регистре FLAGS.
14. Для чего служит флаг CF в регистре FLAGS.
15. Для чего служит флаг AF в регистре FLAGS.
16. Для чего служит флаг ZF в регистре FLAGS.
17. Для чего служит флаг SF в регистре FLAGS.
18. Для чего служит флаг TF в регистре FLAGS.
19. Для чего служит флаг IF в регистре FLAGS.
20. Для чего служит флаг DF в регистре FLAGS.
21. Для чего служит флаг OF в регистре FLAGS.
22. Перечислите способы адресации в процессоре Intel 8086.
23. Объясните регистровый способ адресации.
24. Объясните непосредственный способ адресации.
25. Объясните прямой способ адресации.

26. Объясните регистровый косвенный способ адресации.
27. Объясните способ адресации - регистровый косвенный со смещением.
28. Объясните способ адресации - базовый индексный.
29. Объясните способ адресации - базовый индексный со смещением.
30. Какие директивы ассемблера существуют для определения данных в процессоре Intel 8086.
31. Какая директива ассемблера предназначена для резервирования массива памяти.
32. Приведите примеры объявления и определения данных на языке Ассемблер.
33. Приведите примеры объявления и определения констант на языке Ассемблер.
34. Что такое атрибутный оператор PTR.
35. Для чего используется счетчик текущего адреса \$ и как его применить на конкретном примере.
36. Арифметические команды процессора Intel 8086. Объясните и приведите примеры использования команды ADD.
37. Арифметические команды процессора Intel 8086. Объясните и приведите примеры использования команды SUB.
38. Арифметические команды процессора Intel 8086. Объясните и приведите примеры использования команды ADC.
39. Арифметические команды процессора Intel 8086. Объясните и приведите примеры использования команды SBB.
40. Арифметические команды процессора Intel 8086. Объясните и приведите примеры использования команды MUL.
41. Арифметические команды процессора Intel 8086. Объясните и приведите примеры использования команды DIV.
42. Арифметические команды процессора Intel 8086. Объясните и приведите примеры использования команды IMUL.
43. Арифметические команды процессора Intel 8086. Объясните и приведите примеры использования команды IDIV.

Для выполнения второй части задания студент должен знать ответы на следующие вопросы:

1. ЭЛТ-монитор. Что такое строчная (горизонтальная) развертка и как она формируется.
2. ЭЛТ-монитор. Что такое кадровая (вертикальная) развертка и как она формируется.
3. ЭЛТ-монитор. Сколько строчных разверток помещается в одной кадровой развертке в стандартном VGA-режиме.
4. ЭЛТ-монитор. Что такое пиксель.
5. ЭЛТ-монитор. Сколько пикселей помещается в одной строчной развертке в стандартном VGA-режиме.
6. ЭЛТ-монитор. Что представляет собой композитный видеосигнал в стандартном VGA-режиме.
7. Что такое текстовый видеорежим.
8. Что такое знакогенератор.
9. Изобразите графическую фигуру буквы 'q' в виде точечной матрицы размером 8x8 и получите коды формы.
10. Что такое кодовая страница и что в ней хранится.
11. Схема простейшего текстового видеоадаптера. Что находится в видеопамяти.
12. Схема простейшего текстового видеоадаптера. Что находится в таблице знакогенератора.

13. Схема простейшего текстового видеоадаптера. Что выполняет сдвиговый регистр.
14. Схема простейшего текстового видеоадаптера. Что выполняет счетчик С3.
15. Схема простейшего текстового видеоадаптера. Что выполняет счетчик С2.
16. Схема простейшего текстового видеоадаптера. Что выполняет счетчик С1.
17. Схема простейшего текстового видеоадаптера. Какие счетчики формируют строчный и кадровый синхросигналы.
18. Видеоадаптер VGA. Сколько видеостраниц доступны в текстовом режиме.
19. Видеоадаптер VGA. Сколько байт занимает один символ в видеопамяти.
20. Видеоадаптер VGA. Что находится в младшем байте и что находится в старшем байте отображаемого символа.
21. Что определяет атрибут символа в видеопамяти.
22. Какие биты атрибута определяют цвет символа, а какие биты – цвет фона.

2.2. Варианты для первой части курсового проекта. Разработка программы расчета математического выражения.

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

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

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

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

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

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

2.3. Варианты для второй части курсового проекта. Разработка программы анимации.

Задания по работе с текстовой видеостраницей

Ответ решения математического выражения поместить в синем квадрате в центре экрана 0 видеостраницы, далее перемещаем квадрат с решением согласно полученному варианту:

1. По клавише «1» - влево до конца, затем появляется справа, доходит до центра и останавливается

По клавише «2» - вправо до конца, затем появляется слева, доходит до центра и останавливается

2. По клавише «1» - движение по диагонали экрана вверх до упора и останавливается

По клавише «2» - обратное движение

3. По клавише «1» - поместить квадрат с ответом по углам разными цветами

По клавише «2» - смена цветов

По клавише «3» - вернуться в исходное состояние

4. По клавише «1» - движение по диагонали экрана вверх на верхний правый угол. При достижении угла продолжить движение с нижнего левого угла по диагонали на верхний левый угол следующей видеостраницы и т.д. по всем видеостраницам. При достижении 7 видеостраницы перейти снова на 0 видеостраницу и т.д. (бесконечный цикл)

По клавише «2» - переключение страниц

5. По клавише «1» - поместить квадрат с ответом на следующей видеостранице по углам разными цветами

По клавише «2» - движение всех квадратов по диагонали в центр с последующим переключением на следующую видеостраницу, при достижении последней видеостраницы возврат на нулевую и повторный цикл.

6. Ответ поместить в двух квадратах, в первом частное, во втором остаток.

По клавише «1» - поместить на следующей видеостранице, частное в правых углах разным цветом, остаток в левых углах разным цветом

По клавише «2» - вернуться в исходное состояние, но на следующей видеостранице. При достижении последней видеостраницы все повторяется начиная с нулевой видеостраницы

7. Ответ поместить в двух квадратах, в первом частное, во втором остаток.

По клавише «1» - частное поместить в центре следующей видеостраницы другим цветом, остаток поместить в центре ещё следующей видеостраницы другим цветом

По клавише «2» - вернуться в исходное состояние, но на следующей видеостранице.

По клавише «3» - переключение видеостраниц для независимого просмотра частного и остатка.

8. Ответ поместить, частное в правых углах разным цветом, остаток в левых углах разным цветом.

По клавише «1» - круговое движение, частное по часовой стрелке, остаток против часовой стрелки. При встрече квадратов происходит смена цветов.

По клавише «2» - обратное круговое движение, частное против часовой стрелки, остаток по часовой стрелке. При встрече квадратов происходит смена цветов

9. Ответ поместить, частное в левых нижних углах четных видеостраниц разным цветом, остаток в левых нижних углах нечетных видеостраниц разным цветом.

По клавише «1» - движение по диагонали в сторону правого верхнего угла во всех видеостраницах, при достижении верхнего правого угла движение продолжить снова с нижнего левого угла

По клавише «2» - обратное движение по диагонали в сторону нижнего левого угла во всех видеостраницах, при достижении нижнего левого угла движение продолжить снова с правого верхнего угла

По клавише «3» - переключение видеостраниц для независимого просмотра частного и остатка.

10. Ответ поместить, частное в правых нижних углах четных видеостраниц разным цветом, остаток в правых нижних углах нечетных видеостраниц разным цветом.

По клавише «1» - движение по диагонали в сторону левого верхнего угла во всех видеостраницах, при достижении верхнего левого угла движение продолжить снова с нижнего правого угла

По клавише «2» - происходит смена диагонали, т.е. движение с левого нижнего угла в сторону верхнего правого во всех видеостраницах.

По клавише «3» - переключение видеостраниц для независимого просмотра частного и остатка.

11. Ответ поместить, частное в правых нижних углах четных видеостраниц разным цветом, остаток в правых нижних углах нечетных видеостраниц разным цветом.

По клавише «1» - круговое движение по часовой стрелке во всех видеостраницах.

По клавише «2» - круговое движение против часовой стрелки во всех видеостраницах.

По клавише «3» - переключение видеостраниц для независимого просмотра частного и остатка.

12. Ответ поместить в центре в двух квадратах, в первом частное, во втором остаток разным цветом.

По клавише «1» - частное движется вверх, при достижении верхнего края экрана появляется снизу, движется до исходной позиции и останавливается. Остаток движется вправо, при достижении правого края появляется слева, движется до исходной позиции и останавливается.

По клавише «2» - повторить то же самое, но в обратном направлении.

2.4. Пример выполнения.

Первая часть. Разработка программы расчета математического выражения.

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.

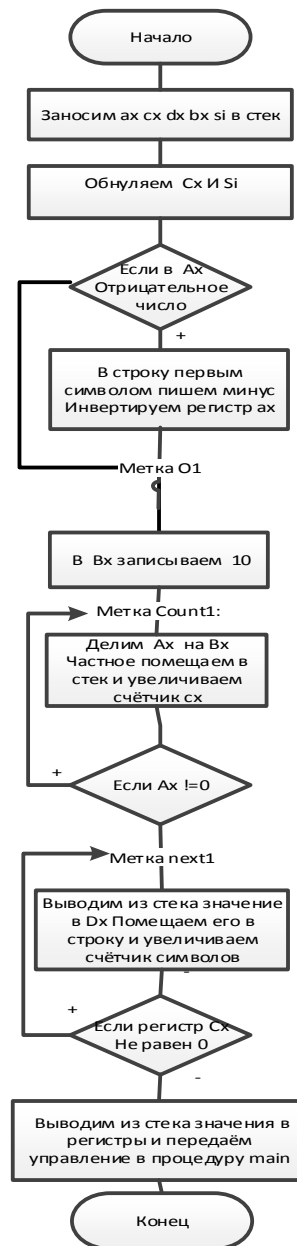


Рисунок 1 – Блок-схема процедуры outstr

На рисунке 2 отображена процедура вычисления заданного арифметического выражения.

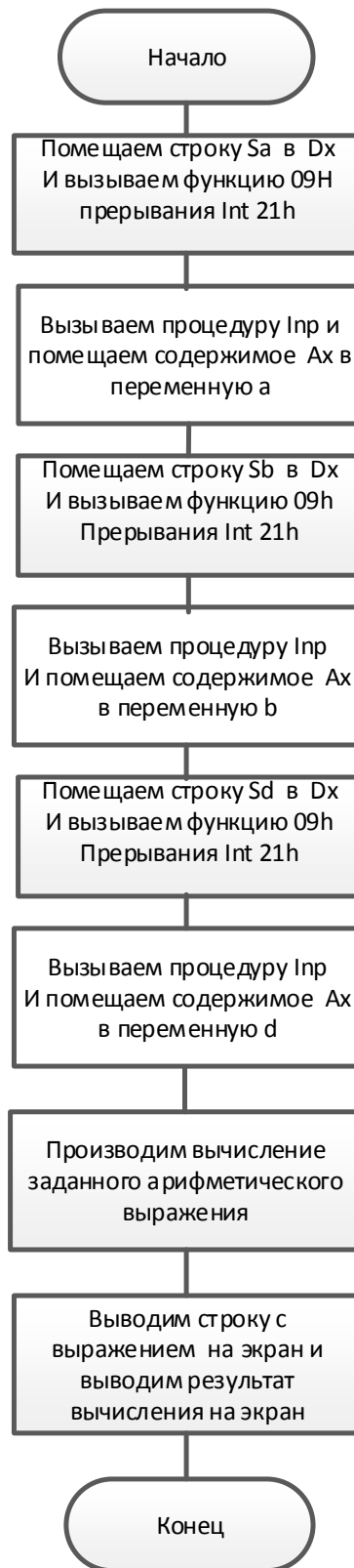


Рисунок 2 – Блок-схема процедуры расчета mrg

Процедура считывания данных с клавиатуры представлена на рисунке 3.

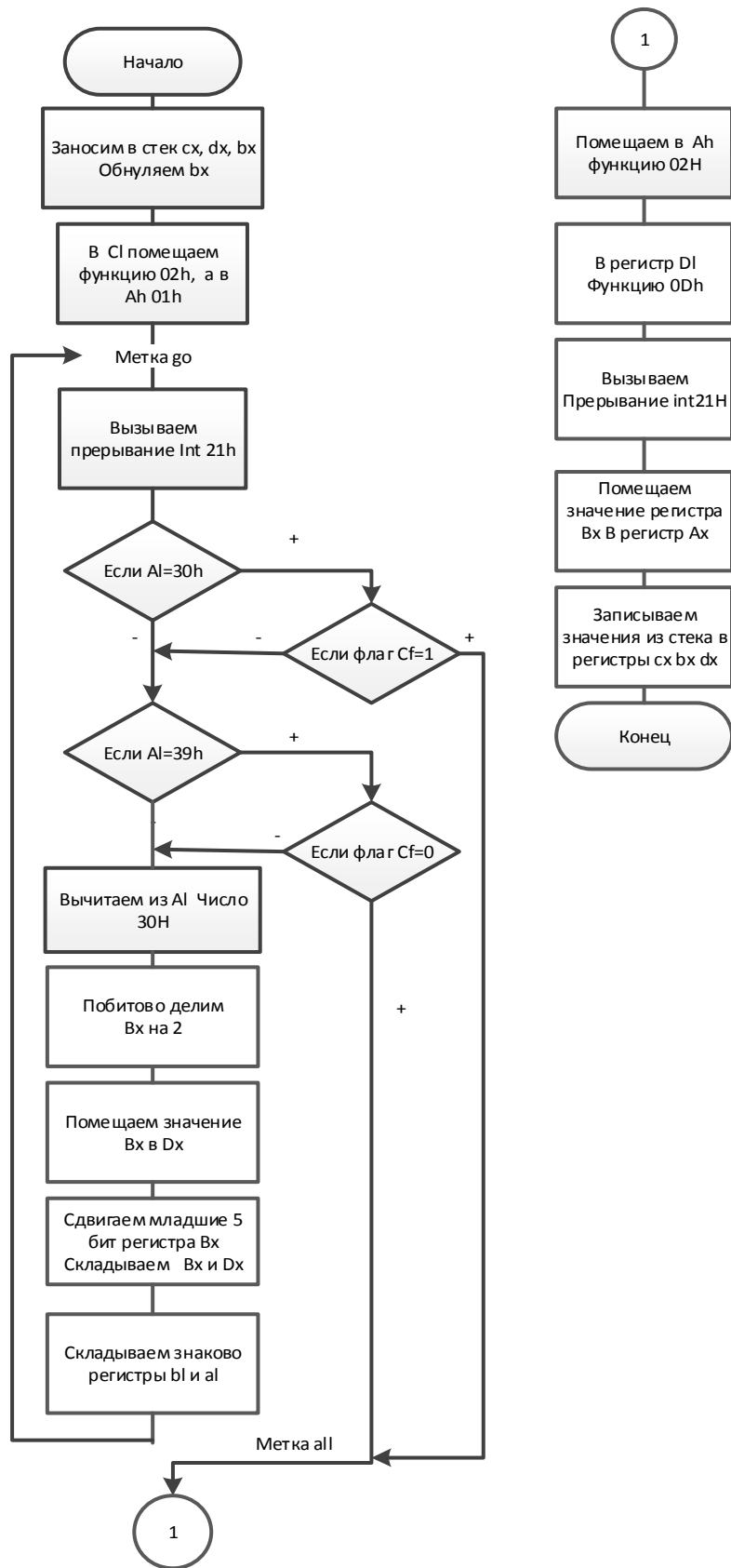


Рисунок 3 – Блок-схема процедуры inr

Программный код

```

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

```

```

;Процедура вычисления арифметического выражения
mpc 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
mpr 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

```

```

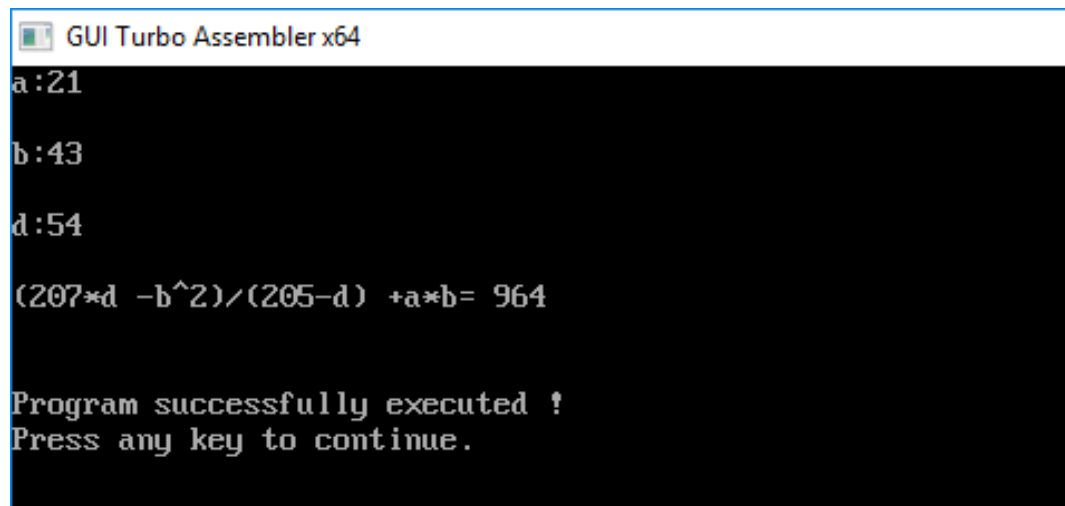
    xor si,si
;Проверяем число <0
    test ax,ax
    jns o1
;Если отрицательное то в первый символ результирующего числа заносим знак минус
    mov text1[si], '-'
    inc si
    neg ax
; Переводим число в десятичную систему счисления и полученные цифры заталкиваем в
стек
o1:
    mov bx,000Ah
count1:
    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:43
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

Вторая часть. Разработка программы анимации.

Цель работы: Написать программу на языке программирования Assembler для работы с видеостраницами.

Задание: Результат вычисления выражения из лабораторной работы №1 поместить в нижний угол видеостраницы.

При нажатии на клавишу 1 – движение по диагонали экрана на верхний правый угол. При достижении угла продолжить движение с нижнего левого угла по диагонали на верхний левый угол следующей видеостраницы и т. д. по всем видеостраницам. При достижении 7 видеостраницы перейти на 1 видеостраницу и т. д. По клавише 2 – переключение видеостраниц.

Блок схемы алгоритма

Блок схема данного алгоритма использует процедуры из прошлых лабораторных работ. В процедуру video были внесены некоторые изменения. Так же были добавлены три процедуры move, changeway и rkey. Процедура перемещения прямоугольника по видеостранице представлена на рисунке 8.

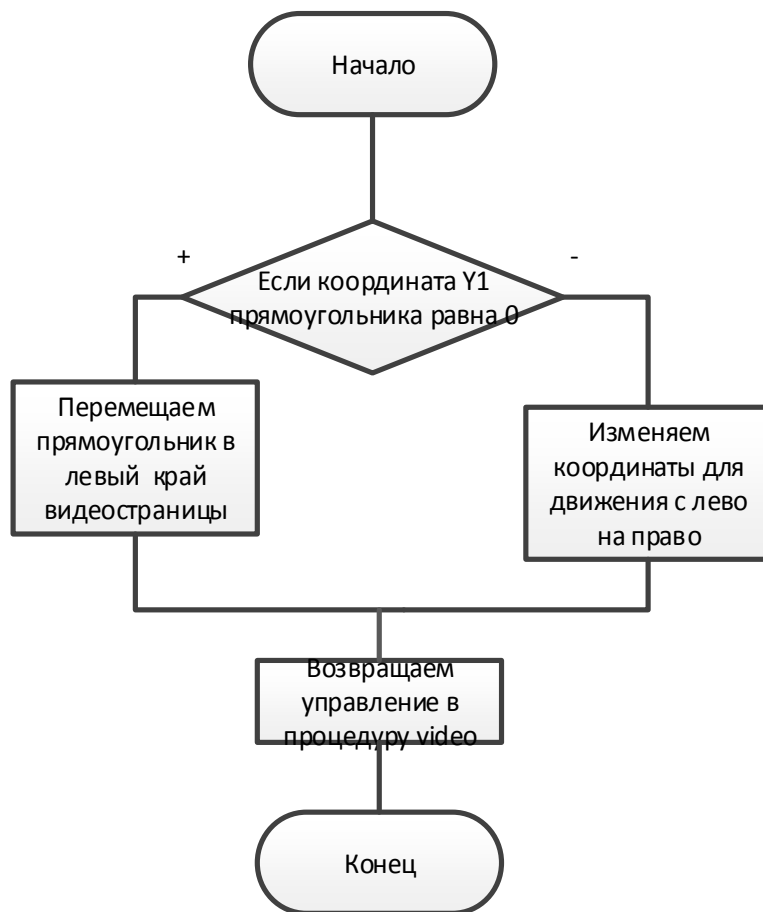


Рисунок 8 – Блок схема процедуры move

На рисунке 9 показана блок схема изменённой процедуры video.

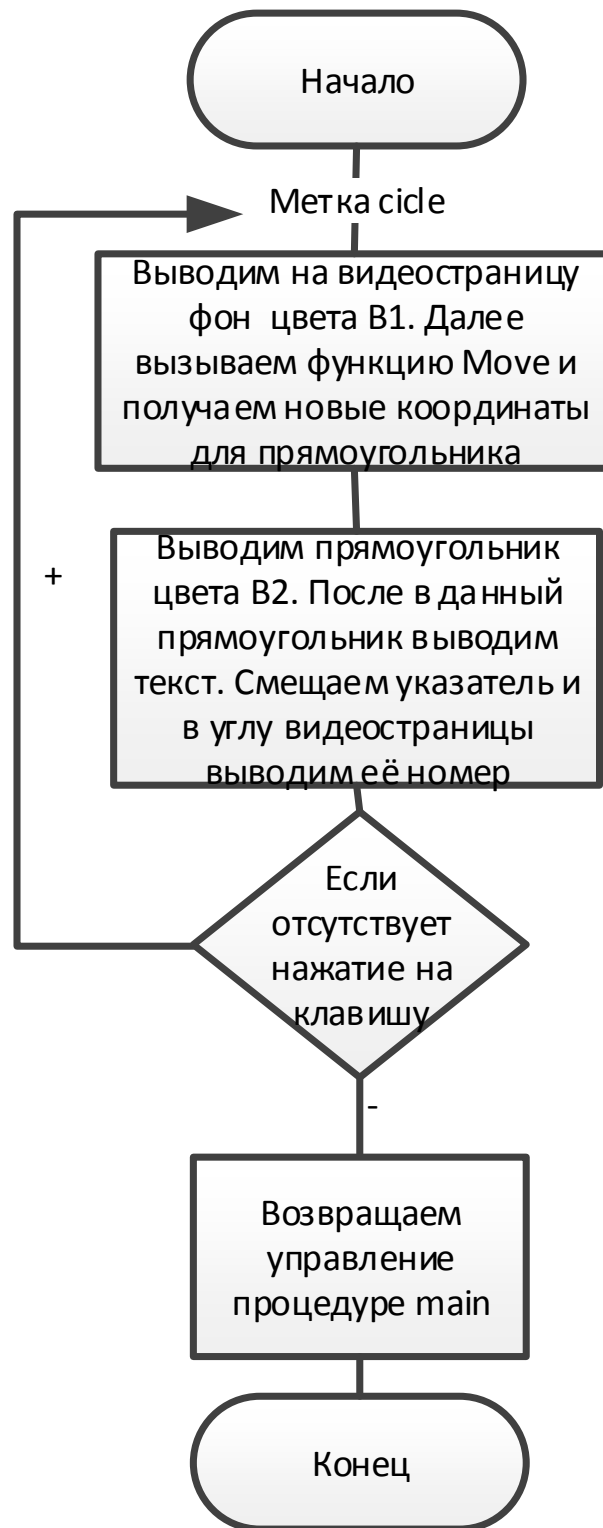


Рисунок 9 – Блок схема изменённой процедуры video

Процедура обработки нажатия клавиш изображена на рисунке 10.

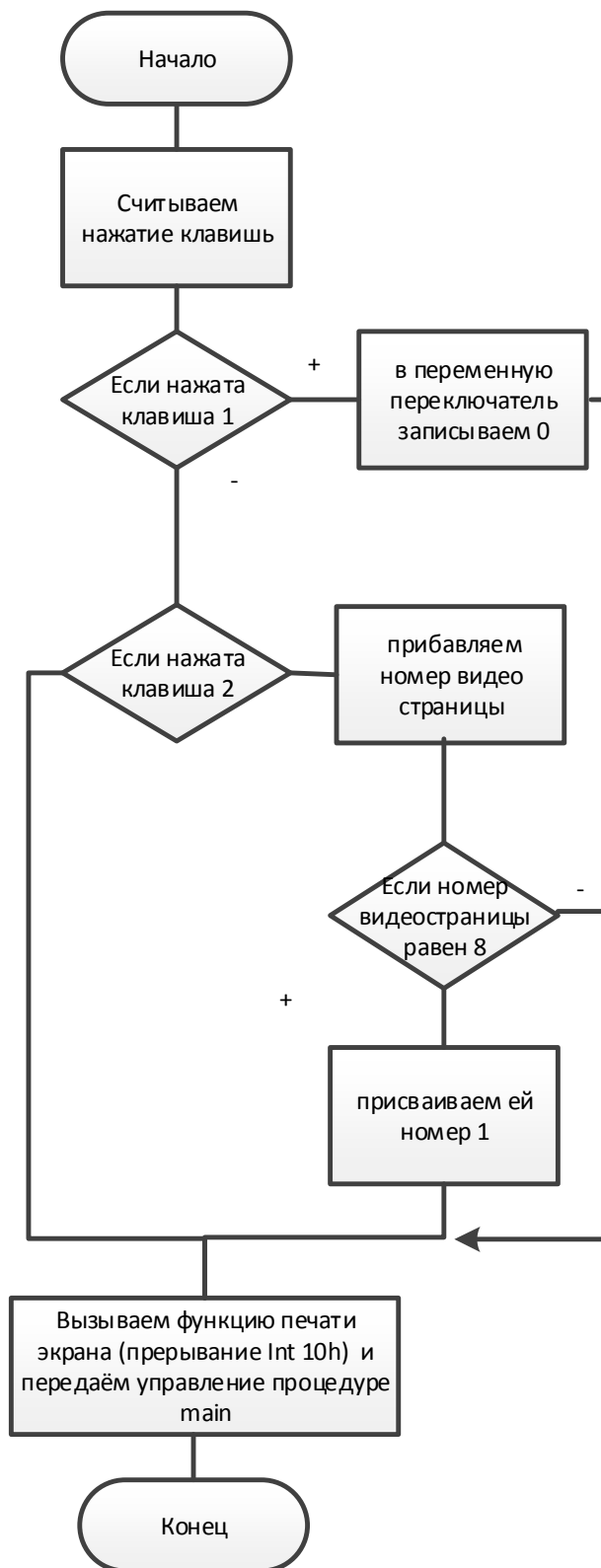


Рисунок 10 - Блок схема процедуры pkey

Процедура переключения видеостраниц изображена на рисунке 11.

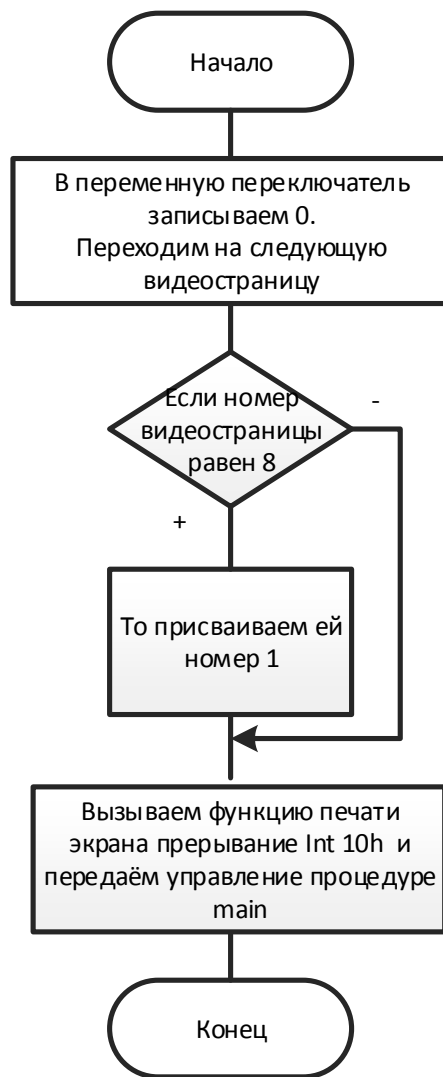


Рисунок 11 – Блок схема процедуры changeway

Программный код

```

stdin =0
stdout=1
stderr=2
data segment
  text1 db 10 dup(?)
  textlen dw ?
  time dw 10
  vs db 0
  b1 db 30h
  b2 db 14h
  tab1 db '0123456789ABCDEF'
  a dw ?
  b dw ?
  d dw ?
  x dw ?
  trig db 3
  trig2 db 3
  str1 db '(207*d -b^2)/(205-d) +a*b= ', '$'
  sa db'a:', '$'
  sb db'b:', '$'
  sd db'd:', '$'

```

```

x1 db 1
x2 db 5
y1 db 23
y2 db 24

data ends;
Code segment
    assume cs:code, ds:data
    start:
;Главная процедура программы, отвечает за вывод видеостраниц на экран и обработку
нажатий клавиш.
main proc
    mov ax, data
    mov ds, ax
;Вызываем процедуру вычисления результата выражения
    call mpr
vid0:
    mov ah,05h
    mov al,vs
    int 10h
;Вызываем процедуру вывода видеостраницы на экран, проверяем переключатель
    call video
    cmp trig2,0
    jne tds1
; Вызываем процедуру считывания символа нажатой клавиши
    call pkey
    tds1:
;Вызываем процедуру переключения видеостраниц
    call changeway
; В зависимости от нажатой клавиши и
выбранной видеостраницы изменяются цвета
    cmp vs,1
    je vid1
vid1:
    cmp vs, 0
    je vid2
    cmp vs, 1
    je vid3
    cmp vs, 2
    je vid4
    cmp vs, 3
    je vid5
    cmp vs, 4
    je vid6
    cmp vs, 5
    je vid7
    cmp vs, 6
    je vid8
vid2:
    mov ax,x
    call outstr
    mov b1,10h
    mov b2,24h
    jmp vid0
vid3:
    mov ax,x
    call outstr
    mov b1,0A0h
    mov b2,0E1h
    jmp vid0
vid4:

```

```

    mov ax,x
    call outstr
    mov b1,0B0h
    mov b2,75h
    jmp vid0
vid5:
    mov ax,x
    call outstr
    mov b1,0A0h
    mov b2,74h
    jmp vid0
vid6:
    mov ax,x
    call outstr
    mov b1,19h
    mov b2,40h
    jmp vid0
vid7:
    mov ax,x
    call outstr
    mov b1,30h
    mov b2,95h
vid8:
    mov ax,x
    call outstr
    mov b1,71h
    mov b2,0D0h
    jmp vid0
main endp
;Процедура вывода видеостраницы
video proc
;По циклу, вызывается процедура move,которая изменяет координаты прямоугольника на
видео странице
    cicle:
        call move
; Далее происходит перерисовка видеостраницы
    mov ah,06h
    mov al,0
    mov bh,b1
    mov ch,0
    mov cl,0
    mov dh,24
    mov dl,79
    int 10h
    mov ah,06h
    mov al,0
    mov bh,b2
    mov ch,y1
    mov cl,x1
    mov dh,y2
    mov dl,x2
    int 10h
    mov ax,seg text1
    mov es,ax
    mov ah,13h
    mov al,0
    mov bh,vs
    mov bl,b2
    mov cx,textlen
    mov dh,y1
    mov dl,x1

```

```

    mov bp,offset text1
    int 10h
    mov ah,02h
    mov bh,vs
    mov dh,0
    mov dl,79
    int 10h
    mov al,vs
    lea bx,tab1
    xlat
    mov bh,vs
    mov cx,1
    mov ah,0Ah
    int 10h
; Цикл работает до нажатия на любую клавишу
    mov ah,01h
    int 16h
    jnz v1
; Для плавности анимации вызывается процедура программной задержки
    call delay
    cmp trig2,1
    jne td
    jmp v1
td:
    jmp cicle
v1:
    delay proc
    mov cx, time
outer1:
    push cx
    mov cx,0
inner1:
    loop inner1
    pop cx
    loop outer1
    ret
delay endp
    ret
video endp
mpr 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
mpg endp
;Процедура считывания числовых данных с клавиатуры
inr 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,c1
add bx,dx
add bl,al
adc bh,00h
jmp go
;После того как введён не числовой символ записываем результат в регистр ax и смещаем
;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
;Процедура обработки нажатия клавиши
pkey proc
; Считываем символ клавиши при нажатие
    mov ah,0h
    int 16h
; Анализируем и выбираем одно из действий
    cmp al,'1'
    je t1
    cmp al,'2'
    je t2
;Если нажата другая клавиша, выходим из программы
    mov ah,4ch
    int 21h
; При нажатии на клавишу 1, в переменную переключатель заносится число 0. Согласно
данной
переменной осуществляется движение прямоугольника слева на право.
t1:
    mov ah, 0
    mov trig, ah
    jmp f
; При нажатии на клавишу 2- переключение видеостраниц, при достижении последней
происходит сброс номера видеостраницы на первую
t2:
    mov ah, vs
    add ah,1
    mov vs,ah
    cmp vs,8
    je t3
    jne f
t3:
    mov ah,1
    mov vs,ah
    jmp f
f:
    mov ah,05h
    mov al,0
    int 10h
    ret
pkey endp
;Процедура переключения видеостраниц
changeway proc
;В переменную переключатель заносится 0, далее переключение видеостраниц, при
достижении последней происходит сброс номера видеостраницы на первую
    mov ah,0
    mov trig2,ah
    mov ah, vs
    add ah,1

```

```

mov vs,ah
cmp vs,8
je t21
jne f1
t21:
    mov ah,1
    mov vs,ah
    jmp f1
f1:
mov ah,05h
mov al,0
int 10h
ret
changeway endp
; Процедура перемещения прямоугольника по видеостранице
move proc
;Проверяем достиг ли края прямоугольник, если да - обрабатываем данное событие
изменяя его координаты на начальные. Изменяем переменные переключатели
cmp y1,0
je tr1
jne nxt
tr1:
    mov ah,0
    mov trig,ah
    mov ah,1
    mov trig2,ah
    mov x1,ah
    mov ah,5
    mov x2,ah
    mov ah,22
    mov y1,ah
    mov ah,23
    mov y2,ah
;Перемещаем прямоугольник
nxt:
    mov ah,x1
    add ah,3
    mov x1,ah
    mov ah,x2
    add ah,3
    mov x2,ah
    mov ah,y1
    sub ah,1
    mov y1,ah
    mov ah,y2
    sub ah,1
    mov y2,ah
    ret
move endp
code ends

stack1 segment
    db 128 (?)
stack1 ends
end start

```

Тестирование программы

При запуске программы мы должны ввести данные в переменные, после ввода данных происходит вывод видеостраницы с прямоугольником в левом нижнем углу, на котором отображён результат вычисления арифметического выражения. Данный прямоугольник будет медленно перемещаться по диагонали в сторону правого верхнего

угла. При достижении правого края видеостраница будет меняться, а прямоугольник продолжит движение с нижнего левого края новой видеостраницы. Так же видеостраница меняется при нажатии клавиши 2. При нажатии любой другой клавиши работа программы завершается.

Пример работы программы изображён на рисунках 12 и 13.

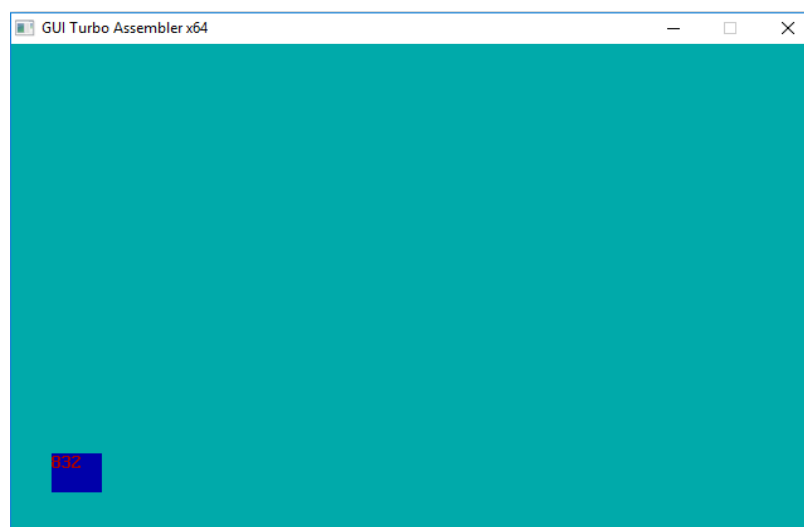


Рисунок 12 – Пример работы программы для работы с видеостраницами

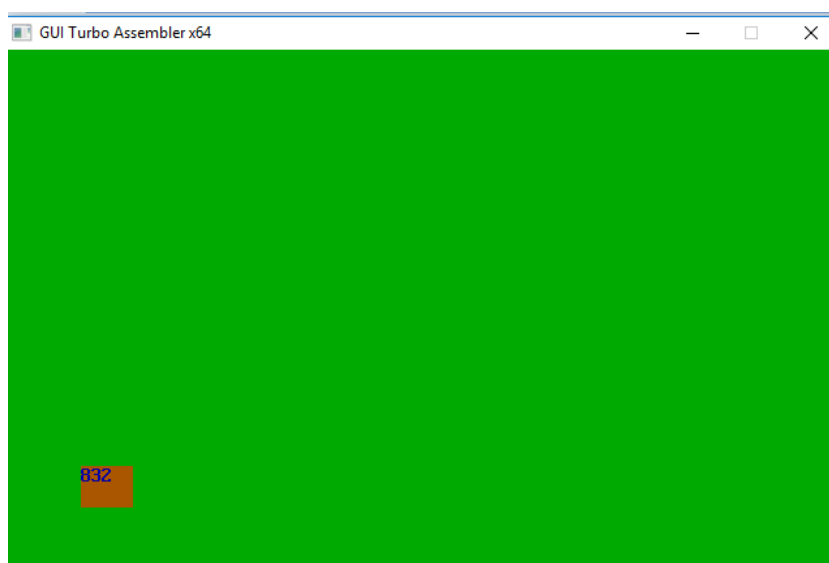


Рисунок 13 – Пример работы программы для работы с видеостраницами

3 Методические указания по самостоятельной работе

Для успешного освоения курса «Б.1.В.ДВ.9.2 Машинно-зависимые языки программирования» 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.ДВ.9.2 Машинно-зависимые языки программирования» проводится в форме экзамена. Для подготовки к итоговому контролю знаний по дисциплине «Б.1.В.ДВ.9.2 Машинно-зависимые языки программирования» 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. 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. (15)

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 Контент, 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 «МИР ПК + DVD».
4. Журнал «ВЕСТНИК КОМПЬЮТЕРНЫХ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ».

5.4 Интернет-ресурсы

1. Питер Абель «Язык Ассемблера для IBM PC и программирования» - Режим доступа: <http://pkm.kai.ru/argo/archive/docs/016/0160/01602.pdf>

2. Ассемблер для начинающих – Режим доступа: <http://av-assembler.ru/asm/afd/introduction.htm>

5.5 Программное обеспечение, профессиональные базы данных и информационные справочные системы современных информационных технологий

Тип программного обеспечения	Наименование	Схема лицензирования, режим доступа
Операционная система	Microsoft Windows	Подписка Campus and School Agreement (CASA) по государственному контракту:

Тип программного обеспечения	Наименование	Схема лицензирования, режим доступа
Офисный пакет	Microsoft Office	➤ № 41ГК/13 от 25.12.2012 г.; ➤ № 33ГК/13 от 24.12.2013 г.
Текстовый редактор	Notepad++	Свободное ПО, https://notepad-plus-plus.org/
Набор средств разработки программного обеспечения	Turbo Assembler	Входит в состав программных продуктов Turbo Pascal 7.0 for DOS и Borland C++ 3.1 for DOS
Интегрированная среда разработки программного обеспечения	Turbo Pascal 7.0 for DOS	Образовательная лицензия по государственному контракту № 34/10 от 10.12.2010 г., лицензия на рабочее место

6 Материально-техническое обеспечение дисциплины

Учебные аудитории для проведения занятий лекционного типа, семинарского типа, для проведения групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Для проведения лабораторных и практических работ используются компьютерный класс (ауд. № 4-113, 4-116, 4-117), оборудованный средствами оргтехники, программным обеспечением, персональными компьютерами, объединенными в сеть с выходом в Интернет.

Аудитории оснащены комплектами ученической мебели, техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой, подключенной к сети «Интернет», и обеспечением доступа в электронную информационно-образовательную среду Орского гуманитарно-технологического института (филиала) ОГУ (ауд. № 4-307).

Наименование помещения	Материально-техническое обеспечение
Учебные аудитории: - для проведения занятий лекционного типа, семинарского типа, - для групповых и индивидуальных консультаций; - для текущего контроля и промежуточной аттестации	Учебная мебель, классная доска, мультимедийное оборудование (проектор, экран, ноутбук с выходом в сеть «Интернет»)
Компьютерные классы № 4-113, 4-116, 4-117	Учебная мебель, компьютеры (29) с выходом в сеть «Интернет», проектор, экран, лицензионное программное обеспечение
Помещение для самостоятельной работы обучающихся, для курсового проектирования (выполнения курсовых работ)	Учебная мебель, компьютеры (3) с выходом в сеть «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Орского гуманитарно-технологического института (филиала) ОГУ, программное обеспечение

Для проведения занятий лекционного типа используются следующие наборы демонстрационного оборудования и учебно-наглядные пособия:

- презентации к курсу лекций.