

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.8.1 Архитектура вычислительных систем»

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

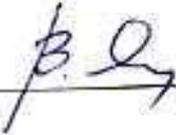
Очная

Год начала реализации программы (набора)

2014, 2015, 2016, 2017

г. Орск 2017

Методические указания для обучающихся по освоению дисциплины «Б.1.В.ДВ.8.1 Архитектура вычислительных систем» 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 по лабораторным и практическим работам

Изучение дисциплины «Программирование» 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

Работа с большими числами

Диапазон целых чисел, помещаемых в ЭВМ, ограничена разрядностью процессора. Например, в 16-ти разрядный процессор Intel 8086 можно поместить максимальное большое число без учета знака 65535, а с учетом знака от -32768 до +32767. В реальной жизни этого недостаточно. Возникает вопрос, как считать миллионы, миллиарды и т.д. на таком процессоре без использования математического сопроцессора. Для этого используются специальные алгоритмы, которые реализованы в математических библиотеках и доступны языкам высокого уровня. Данные алгоритмы детально рассматриваются в лекционном материале. Для хранения и работы с большими числами на процессорах с малой разрядностью используются одновременно несколько регистров. Здесь эти алгоритмы реализованы на языке Ассемблер, который позволяет детально, изнутри узнать принцип их работы.

Состав необходимого инструментария для программирования на языке Ассемблер и приемы работы с ними рассмотрены в лабораторной работе №1 дисциплины «Б.1.Б.15 Организация электронно-вычислительных машин и систем». Здесь используются те же приемы. Ниже приводится пример выполнения одного из вариантов поставленной задачи.

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

Задание: Вычислить результат выражения $x=580136*a^2-(1587431*c*d)/a*b^2$

Главная процедура программы представлена на рисунке 1.

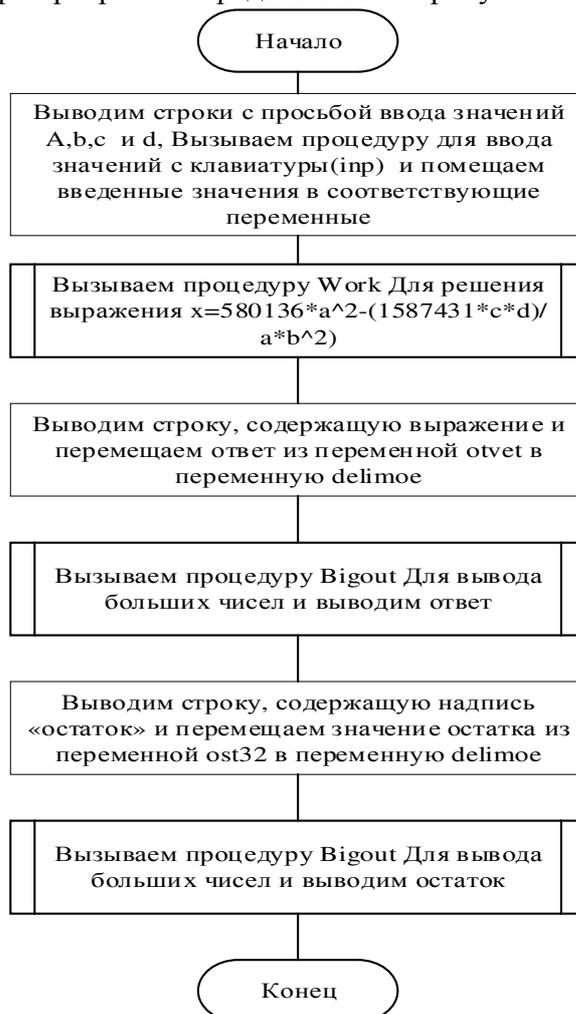


Рисунок 1 – Блок-схема процедуры main

На рисунке 2 отображена процедура вычисления результата уравнения.

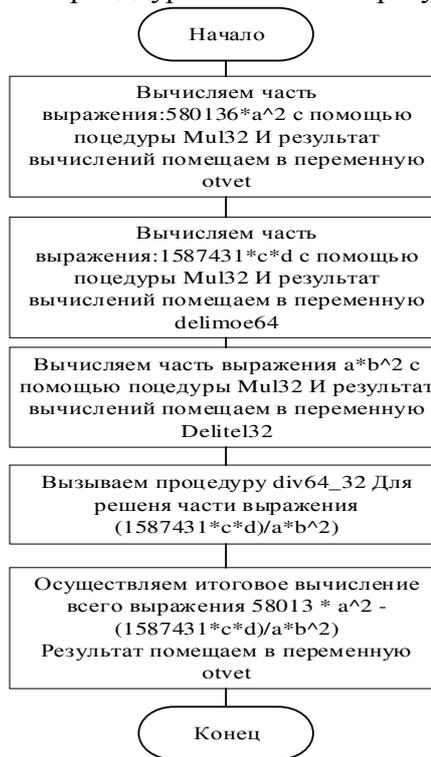


Рисунок 2 – Блок-схема процедуры work

Процедура вывода больших чисел представлена на рисунке 3.

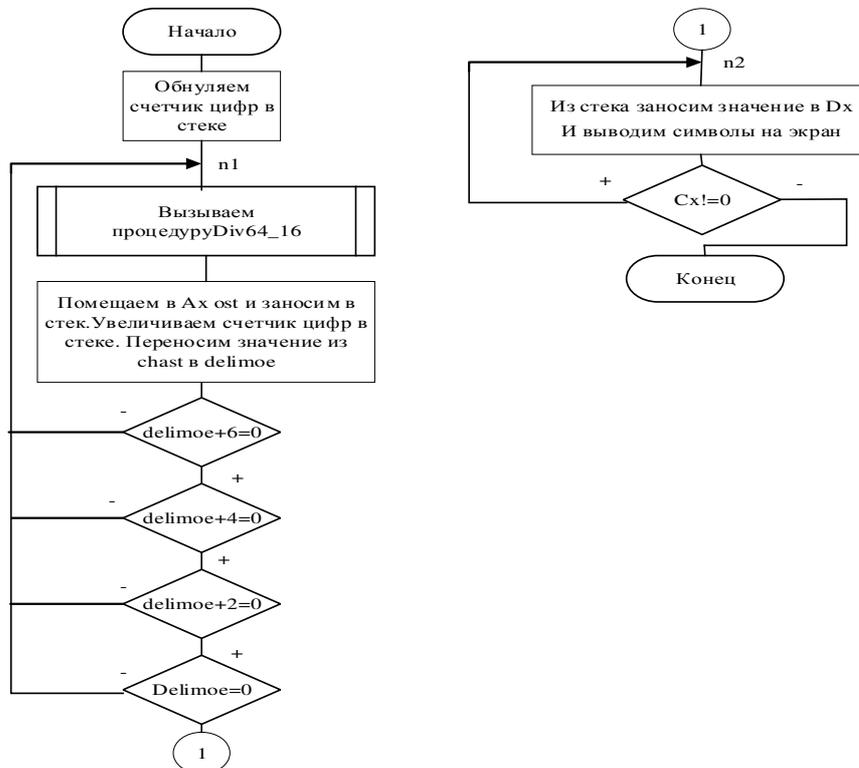


Рисунок 3 – Блок-схема процедуры bigout

Процедура умножения больших чисел представлена на рисунке 4.

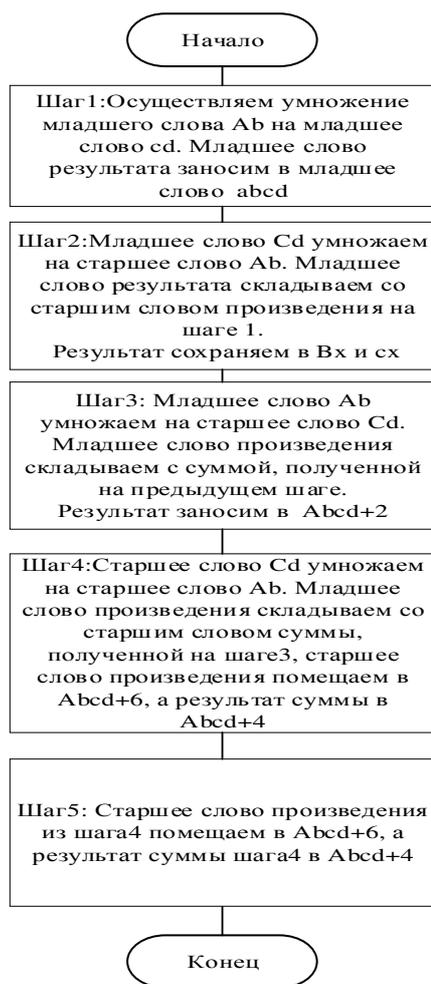


Рисунок 4 – Блок-схема процедуры mul32

Процедура деления большого числа на малое представлена на рисунке 5.

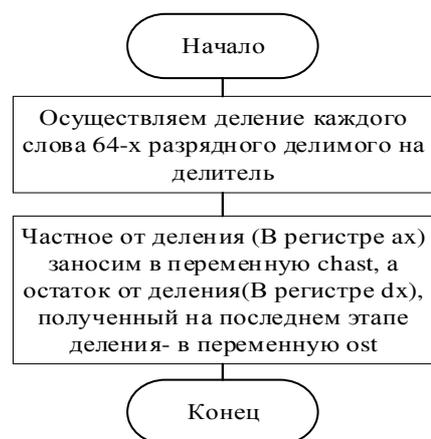


Рисунок 5 – Блок-схема процедуры div64_16

Процедура деления большого числа на большое число представлена на рисунке 6.

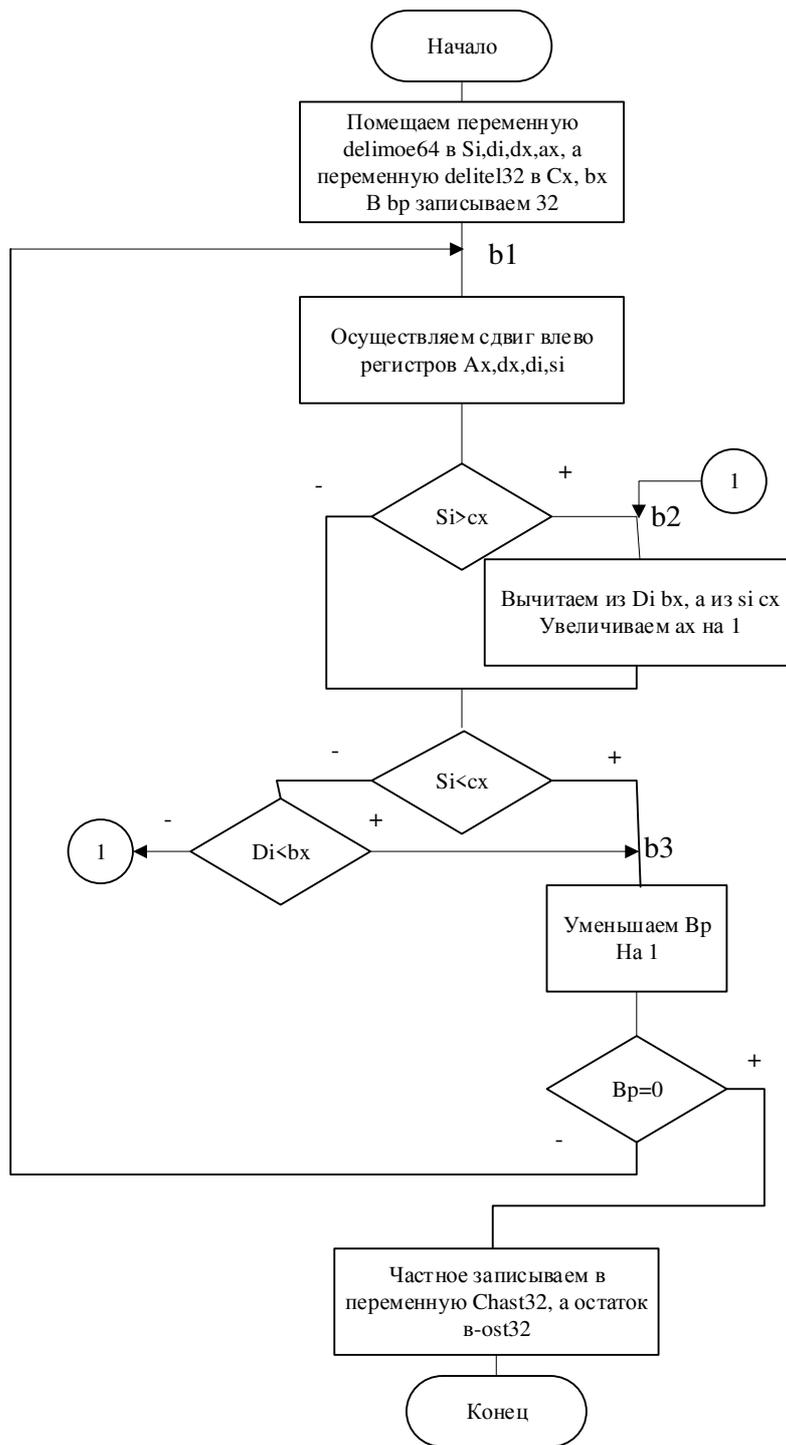


Рисунок 6 – Блок-схема процедуры div64_32

Процедура считывания данных с клавиатуры представлена на рисунке 7.

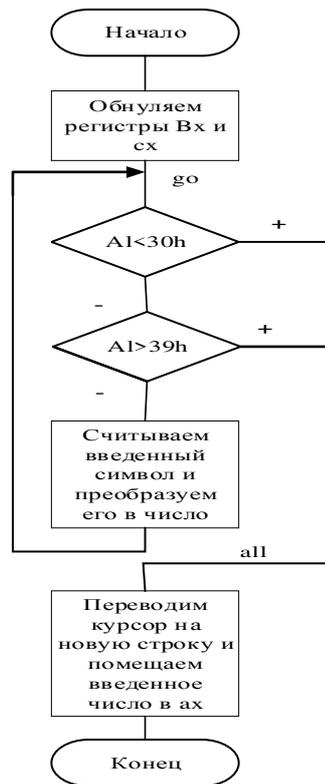


Рисунок 7 – Блок-схема процедуры inp

Программный код

```

text segment 'code'
assume cs:text, ds:data
;Основная процедура программы
main proc
    mov ax,data
    mov ds, ax
    lea dx,sa
    mov ah, 09h
    int 21h
    call inp
    mov a,ax
    lea dx,sb
    mov ah, 09h
    int 21h
    call inp
    mov b,ax
    lea dx,sc
    mov ah, 09h
    int 21h
    call inp
    mov c,ax
    lea dx,sd
    mov ah, 09h
    int 21h
    call inp
    mov d,ax
    call work
    lea dx,resstr
    mov ah,09h
    int 21h
    mov ax, word ptr otvet+2
  
```

```

n1:
    call div64_16
    mov ax, ost
    push ax
    inc cx
    mov ax, word ptr chast+6
    mov word ptr delimoe+6 ,ax
    mov ax, word ptr chast+4
    mov word ptr delimoe+4 ,ax
    mov ax, word ptr chast+2
    mov word ptr delimoe+2 ,ax
    mov ax, word ptr chast
    mov word ptr delimoe ,ax
    cmp word ptr delimoe+6,0
    jne n1
    cmp word ptr delimoe+4,0
    jne n1
    cmp word ptr delimoe+2,0
    jne n1
    cmp word ptr delimoe,0
    jne n1
n2:
    pop ax
    mov dx,ax
    mov ah,02h
    add dl,30h
    int 21h
    loop n2
    pop dx
    pop cx
    pop ax
  
```

```

mov word ptr delimoe+2,ax
mov ax, word ptr otvet
mov word ptr delimoe,ax
mov ax, word ptr otvet+4
mov word ptr delimoe+4,ax
mov ax, word ptr otvet+6
mov word ptr delimoe+6,ax
call bigout
lea dx,so
mov ah,09h
int 21h
mov ax, word ptr ost32+2
mov word ptr delimoe+2,ax
mov ax, word ptr ost32
mov word ptr delimoe,ax
mov word ptr delimoe+4,0
mov word ptr delimoe+6,0
call bigout
mov ax,4c00h
int 21h
main endp
;Процедура вычисления выражения
work proc
push ax
push bx
push cx
push dx

;580136*a^2
mov ax,8
mov word ptr ab+2,ax
mov ax,55848
mov word ptr ab,ax
mov ax,a
mul a
mov word ptr cd,ax
mov word ptr cd+2,dx
call mul32
mov ax, word ptr abcd
mov word ptr otvet,ax
mov ax, word ptr abcd+2
mov word ptr otvet+2,ax
mov ax, word ptr abcd+4
mov word ptr otvet+4,ax
mov ax, word ptr abcd+6
mov word ptr otvet+6,ax
;1587431*c*d
mov ax,24
mov word ptr ab+2,ax
mov ax,14567
mov word ptr ab,ax
mov ax,c
mul d
mov word ptr cd,ax
mov word ptr cd+2,dx
call mul32
mov ax,word ptr abcd+6
mov word ptr delimoe64+6,ax
mov ax,word ptr abcd+4
mov word ptr delimoe64+4,ax
mov ax,word ptr abcd+2
mov word ptr delimoe64+2,ax

```

```

ret
bigout endp
;Процедура деления 64-х разрядного числа на 16-
ти разрядное
div64_16 proc
push ax
push dx
mov ax,word ptr delimoe+6
xor dx,dx
div delitel
mov word ptr chast+6,ax
mov ax,word ptr delimoe+4
div delitel
mov word ptr chast+4,ax
mov ax,word ptr delimoe+2
div delitel
mov word ptr chast+2,ax
mov ax,word ptr delimoe
div delitel
mov word ptr chast,ax
mov ost,dx
pop dx
pop ax
ret
div64_16 endp
; Процедура деления 64-х разрядного числа на 32-х
разрядное
div64_32 proc
push ax
push bx
push cx
push dx
push si
push di
push bp
mov si,word ptr delimoe64+6
mov di,word ptr delimoe64+4
mov dx,word ptr delimoe64+2
mov ax,word ptr delimoe64
mov cx,word ptr delitel32+2
mov bx,word ptr delitel32
mov bp,32
b1:
shl ax,1
rcl dx,1
rcl di,1
rcl si,1
cmp si,cx
ja b2
jb b3
cmp di,bx
jb b3
b2:
sub di,bx
sbb si,cx
inc ax
adc dx,0
b3:
dec bp
cmp bp,0
jne b1

```

```

mov ax,word ptr abcd
mov word ptr delimo64,ax
;a*b^2
mov ax,b
mul b
mov word ptr ab+2,dx
mov word ptr ab,ax
mov ax,a
mov dx,0
mov word ptr cd+2,dx
mov word ptr cd,ax
call mul32
;(1587431*c*d)\a*b^2)
mov ax,word ptr abcd
mov dx,word ptr abcd+2
mov word ptr delitel32,ax
mov word ptr delitel32+2,dx
call div64_32
;58013 * a^2 - (1587431*c*d)\a*b^2)
mov ax,word ptr otvet
mov cx,word ptr chast32
sub ax,cx
mov word ptr otvet,ax
mov ax,word ptr otvet+2
mov cx,word ptr chast32+2
sbb ax,cx
mov word ptr otvet+2,ax

mov ax,word ptr otvet+4
mov cx,0
sbb ax,cx
mov word ptr otvet+4,ax
mov ax,word ptr otvet+6
mov cx,0
sbb ax,cx
mov word ptr otvet+6,ax
pop dx
pop cx
pop bx
pop ax
work endp
;Процедура произведения больших чисел
mul32 proc
push ax
push bx
push cx
push dx
mov ax,word ptr cd
mov bx,ax
mul word ptr ab
mov word ptr abcd,ax
mov cx,dx
mov ax,bx
mul word ptr ab+2
add ax,cx
adc dx,0
mov bx,ax
mov cx,dx
mov ax, word ptr cd+2
mul word ptr ab
add ax,bx
adc cx,dx

```

```

mov word ptr chast32+2,dx
mov word ptr chast32,ax
mov word ptr ost32+2,si
mov word ptr ost32,di
pop bp
pop di
pop si
pop dx
pop cx
pop bx
pop ax
ret
div64_32 endp
; Процедура считывания данных с клавиатуры
inp proc
push dx
push bx
xor cx,cx
xor bx, bx
mov cl,02h
mov ah,01h
go:
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
adc cx,0
add bl,al
adc bh,00h
jmp go
all:
mov ah,02h
mov dl,0Ah
int 21h
mov dl,0Dh
int 21h
mov ax,bx
pop bx
pop dx
ret
inp endp
text ends
data segment
a dw 0
b dw 0
c dw 0
d dw 0
nolic dq 0
ab dd 0
cd dd 0
abcd dq 0
buf dq 0
;div 64_16
delimo64 dq 0
delitel dw 10

```

```

mov word ptr abcd+2,ax
mov ax,word ptr cd+2
mul word ptr ab+2
add ax,cx
adc dx,0
mov word ptr abcd+4,ax
mov word ptr abcd+6,dx
pop dx
pop cx
pop bx
pop ax
ret
mul32 endp
; Процедура вывода больших чисел
bigout proc
push ax
push cx
push dx
xor cx, cx
chast dq 0
ost dw 0
;div 64_32
delimoe64 dq 0
delitel32 dd 0
chast32 dd 0
ost32 dd 0
otvet dq 0
resstr db 0ah,0dh,"580136 * a^2 -
(1587431*c*d)/a*b^2)=", "$"
so db " ostatok:", "$"
sa db "a=", "$"
sb db "b=", "$"
sc db "c=", "$"
sd db "d=", "$"
data ends
stack1 segment para stack 'stack1'
db 128 dup(?)
stack1 ends
end main

```

Тестирование программы

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

Пример работы программы изображён на рисунке 8.

```

GUI Turbo Assembler x64
a=15
b=10
c=10
d=100

580136 * a^2 - (1587431*c*d)/a*b^2)=129472313 ostatok:500

Program successfully executed !
Press any key to continue.

```

Рисунок 8 – Пример работы программы расчёта арифметического выражения

Варианты работ для самостоятельного выполнения:

- $$x = \frac{2460a^2b - 1370cd}{d^3} + c^3$$

$$2. \quad y = \frac{110370d^2}{c^3 + 876a^2b} - 11831a^3$$

$$3. \quad z = \frac{k^2bc - 878a^2}{18780k} + c^2b$$

$$4. \quad x = \frac{87810c^2d + ba^2}{148cd - k} + k^2$$

$$5. \quad y = \frac{22110138k^2}{abk - 197d} - d^3$$

$$6. \quad z = 580136a^2 - \frac{1587431dc}{ab^2}$$

$$7. \quad x = 138136c^2d - \frac{841a^2b - cd^2}{ab^2}$$

$$8. \quad y = 879003x^2b - \frac{1250023c^3}{c^3 + b}$$

$$9. \quad z = 4500341xy + \frac{1455638d - x^3}{d^3}$$

$$10. \quad x = \frac{274807d - b^3}{32405 - zd} + bd^3$$

$$11. \quad y = \frac{768490c^3 - 417555d^3}{ab + 23412c} + d^2$$

Лабораторная работа №2

Работа с вещественными числами. Математический сопроцессор Intel 8087 (FPU).

В лекционном материале подробно рассмотрены следующие вопросы:

- перевод десятичных вещественных чисел в двоичный вид и упаковки их в 32-х разрядное число с плавающей запятой типа float по стандарту IEEE 754;
- арифметические действия с числами с плавающей запятой в двоичном виде, где итогом является написание контрольной работы;
- внутреннее устройство математического сопроцессора Intel 8087 и типы поддерживаемых данных;
- схема соединения и принцип совместной работы центрального процессора Intel 8087 (CPU) и математического сопроцессора Intel 8087 (FPU);
- математические команды (F-команды) сопроцессора и пример программы.

Здесь приводится пример итоговой программы для математического сопроцессора – индивидуальное творческое задание.

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

Задание: Вычислить результат выражения $z = ((1455.638 * d - x^3) / d^3) + 4500.341 * x * y$

Основная процедура программы предоставлена на рисунке 9.

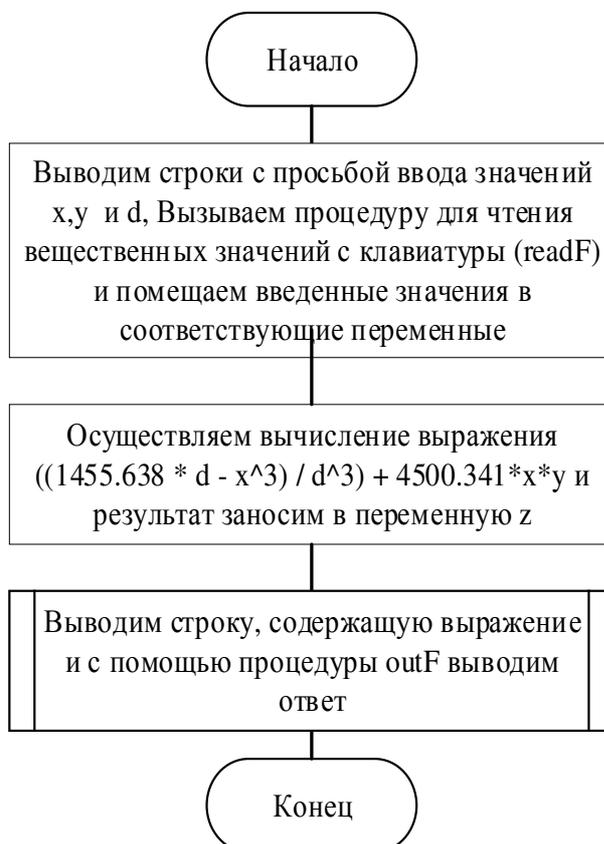


Рисунок 9 – Блок-схема процедуры main

На рисунке 10 отображена процедура вывода вещественного числа.

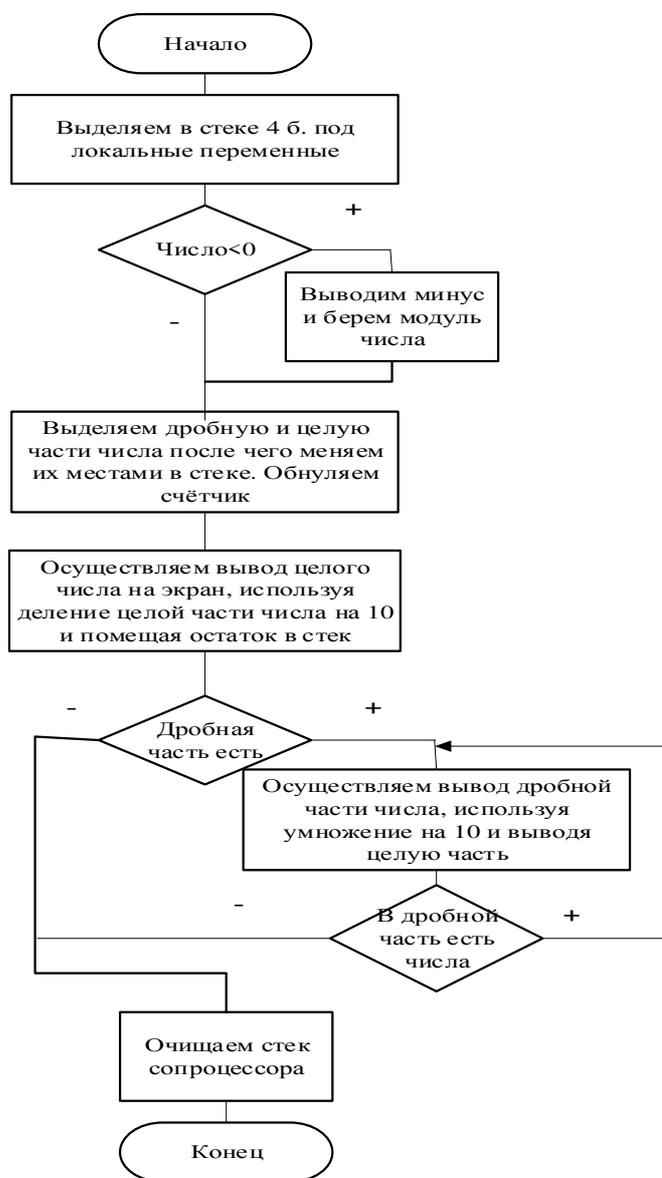


Рисунок 10 – Блок-схема процедуры outF

Процедура ввода вещественного числа предоставлена на рисунке 11.

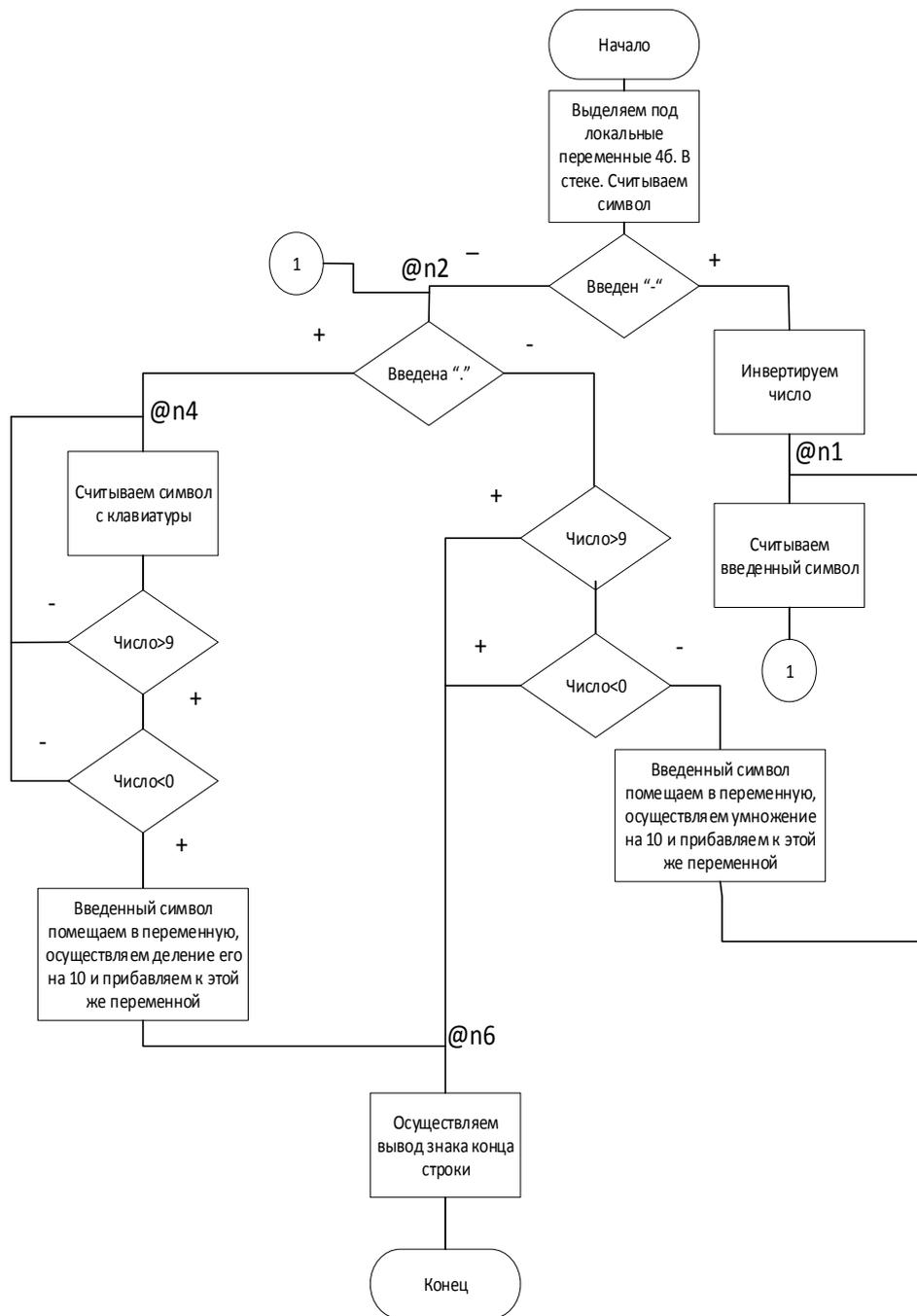


Рисунок 11 – Блок-схема процедуры readF

Программный код

```
.286
codesg segment
    Assume cs: codesg, ds:datasg
start:
;Основная процедура программы
    main proc
        mov ax,datasg
        mov ds,ax
        finit
        lea dx,strx
        mov ah, 09h
        int 21h
        call readF
```

```
int 29h
    mov cx, lengf
    @p4:
        fimul ten
        fxch st(1)
        fld st(1)
        fprem
        fsub st(2), st
        fxch st(2)
        fistp temp
        mov ax, temp
        or al, 30h
    int 29h
```

```

fst x
lea dx, strd
mov ah, 09h
int 21h
call readF
fst d
lea dx, stry
mov ah, 09h
int 21h
call readF
fst y
;x^3
fld x
fld x
fmul st(0), st(1)
fmul st(0), st(1)
fst x3
;1455,638*d
fld Numb1
fld d
fmul st(0), st(1)
fst Numb1
;(1455,638*d - x^3)
fld x3
fld Numb1
fsub st(0), st(1)
fst Numb1
;d^3
fld d
fld d
fmul st(0), st(1)
fmul st(0), st(1)
fst d3
;(1455,638*d - x^3) / d^3
fld d3
fld Numb1
fdiv st(0), st(1)
fst Numb1
;x*y
fld x
fld y
fmul st(0), st(1)
fst x
;x*y*4500.341
fld Numb2

```

```

fld x
fmul st(0), st(1)
fst Numb2
;((1455.638 * d - x^3) / d^3) + 4500.341*x*y
fld Numb2
fld Numb1
fadd st(0), st(1)
fst z
lea dx, resultstr
mov ah, 09h
int 21h
call outF
fwait
mov ax, 4c00h
int 21h
main endp

```

```

fxch st(1)
fst
fstsw ax
sahf
loopne @p4
@p5:
fstp
fstp st
leave
ret 2
OutF endp
;Процедура чтения числа
readF proc
enter 4, 0
mov ten, 10
fldz
mov ah, 01h
int 21h
xor bx, bx
cmp al, '-'
jne @n2
xor bx, 1
@n1:
mov ah, 01h
int 21h
@n2:
cmp al, '.'
je @n3
cmp al, '9'
ja @n6
sub al, '0'
jb @n6
cbw
mov temp, ax
fimul ten
fiadd temp
jmp @n1
@n3:
fld1
@n4:
mov ah, 01h
int 21h
cmp al, '9'
ja @n5
sub al, '0'
jb @n5
cbw
mov temp, ax
fidiv ten
fld st(0)
fimul temp
faddp st(2), st
jmp short @n4
@n5:
fstp st(0)
@n6:
mov al, 10
int 29h
or bx, bx
jz @n7
fchs

```

;Процедура вывода числа

```
OutF proc
    enter 4, 0
    mov ten, 10
    fstst
    fstsw ax
    sahf
    jnc @p1
    mov al, '-'
    int 29h
    fchs
    @p1:
    fld1
    fld st(1)
    fprem
    fsub st(2), st
    fxch st(2)
    xor cx, cx
    @p2:
    fidiv ten
    fxch st(1)
    fld st(1)
    fprem
    fsub st(2), st
    fimul ten
    fistp temp
    push temp
    inc cx
    fxch st(1)
    fstst
    fstsw ax
    sahf
    jnz @p2
    @p3:
    pop ax
    add al, '0'
    int 29h
    loop @p3
    fstp st
    fxch st(1)
    fstst
    fstsw ax
    sahf
    jz @p5
    mov al, '!
```

```
@n7:
    leave
    ret
readF endp
codesg ends
datasg segment
    x dd 0
    y dd 0
    z dd 0
    d dd 0
    x3 dd 0
    d3 dd 0
    Numb2 dd 4500.341
    Numb1 dd 1455.638
    Org 20h
    temp dw 0
    ten dw 10
    lengf dw 3
    strx db "x=", "$"
    stry db "y=", "$"
    strd db "d=", "$"
    resultstr db "((1455.638 * d - x^3) / d^3) +
4500.341*x*y=", "$"
    psvh db 0Ah, 0Dh, '$'
datasg ends
stacksg segment
    db 128 (?)
stacksg ends
end start
```

Тестирование программы

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

Пример работы программы изображён на рисунке 12.

```
GUI Turbo Assembler x64
x=2.4
d=5.4
y=2.1
((1455.638 * d - x^3) / d^3) + 4500.341*x*y=22731.548
Program successfully executed !
Press any key to continue.
```

Рисунок 12 – Пример работы программы расчёта арифметического выражения

Варианты работ для самостоятельного выполнения:

1.
$$x = \frac{24,60a^2b - 13,70cd}{d^3} + c^3$$

2.
$$y = \frac{110,370d^2}{c^3 + 8,76a^2b} - 11,831a^3$$

3.
$$z = \frac{k^2bc - 8,78a^2}{18,780k} + c^2b$$

4.
$$x = \frac{87,810c^2d + ba^2}{14,8cd - k} + k^2$$

5.
$$y = \frac{221,10138k^2}{abk - 1,97d} - d^3$$

$$6. \quad z = 580,136a^2 - \frac{1587,431dc}{ab^2}$$

$$7. \quad x = 138,136c^2d - \frac{84,1a^2b - cd^2}{ab^2}$$

$$8. \quad y = 87,9003x^2b - \frac{12500,23c^3}{c^3 + b}$$

$$9. \quad z = 4500,341xy + \frac{1455,638d - x^3}{d^3}$$

$$10. \quad x = \frac{274,807d - b^3}{32,405 - zd} + bd^3$$

$$11. \quad y = \frac{768,490c^3 - 417,555d^3}{ab + 23,412c} + d^2$$

Лабораторная работа №3

Работа с RISC процессорами.

Современные микропроцессоры, в основном, делятся на два типа:

- CISC – процессоры со сложным набором команд, н.п. семейство процессоров компании Intel, Motorola, DEC и т.д.;

- RISC – процессоры с упрощенным (укороченным) набором команд, н.п. процессоры ARM, MIPS и т.д.

По количеству инсталляции (по популярности), RISC-процессоры превосходят CISC-процессоры из-за широкого распространения мобильных устройств.

В лекционном материале подробно рассмотрены следующие вопросы:

- внутреннее устройство 32-х разрядного процессора MIPS;
- файловый регистр – номера и имена регистров и их специальное назначение;
- система адресации и ассемблерные команды;
- типы данных и системные вызовы syscall.

В данной и следующей лабораторной работе будут повторены задания из лабораторных работ №1 и №2, но для процессора MIPS с учетом их тонкостей. В качестве программного инструментария будет использоваться эмулятор процессора MIPS для Windows – программа spim.exe. Исходный текст программы можно набирать в любом редакторе.

Цель работы: Написать программу вычисления результата математического выражения средствами языка программирования Assembler для RISC процессора.

Задание: Вычислить результат выражения $x = 580136 * a^2 - (1587431 * c * d) / a * b^2$

Блок-схема программы предоставлена на рисунке 13.

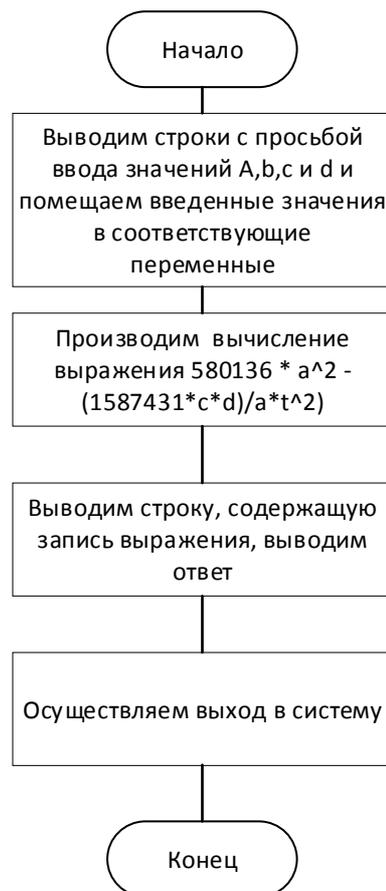


Рисунок 13 – Блок-схема процедуры main

Программный код

<pre> .data x: .double 0.0 a: .double 0.0 t: .double 0.0 c: .double 0.0 d: .double 0.0 ch1: .double 580136.0 ch2: .double 1587431.0 psvk: .asciiiz "\n" stra: .asciiiz "a=" </pre>	<pre> # Вычисление части выражения: a*(t^2) l.d \$f4, a l.d \$f6, t mul.d \$f6, \$f6, \$f4 s.d \$f6, t # Вычисление части выражения: (1587431*c*d)/a*t^2) l.d \$f4, d l.d \$f6, t div.d \$f4, \$f4, \$f6 </pre>
--	--

```

strt: .asciiiz "t="
strc: .asciiiz "c="
strd: .asciiiz "d="
strx: .asciiiz "580136 * a^2 - (1587431*c*d)/a*t^2)= "
.text
main:
#Ввод a
la $a0,stra
li $v0,4
syscall
li $v0,7
syscall
s.d $f0, a
# Ввод b
la $a0,strt
li $v0,4
syscall
li $v0,7
syscall
s.d $f0, t
# Ввод c
la $a0,strc
li $v0,4
syscall
li $v0,7
syscall
s.d $f0, c
# Ввод d
la $a0,strd
li $v0,4
syscall
li $v0,7
syscall
s.d $f0, d
#Вычисление части выражения: c*d
l.d $f4, c
l.d $f6, d
mul.d $f6, $f6, $f4
s.d $f6, c
# Вычисление части выражения: 1587431*(c*d)
l.d $f4, ch2
l.d $f6, c
mul.d $f6, $f6, $f4
s.d $f6, d
# Вычисление части выражения: t^2
l.d $f4, t
mul.d $f4, $f4, $f4
s.d $f4, t

```

```

s.d $f4, d
# Вычисление части выражения: a^2
l.d $f4, a
mul.d $f4, $f4, $f4
s.d $f4, a
# Вычисление части выражения: 580136*(a^2)
l.d $f4, ch1
l.d $f6, a
mul.d $f6, $f6, $f4
s.d $f6, a
# Вычисление выражения 580136 * a^2 -
(1587431*c* d) / a*t^2)
l.d $f4, a
l.d $f6, d
sub.d $f4, $f4, $f6
s.d $f4, x
#Вывод строки выражения и ответа
la $a0,strx
li $v0,4
syscall

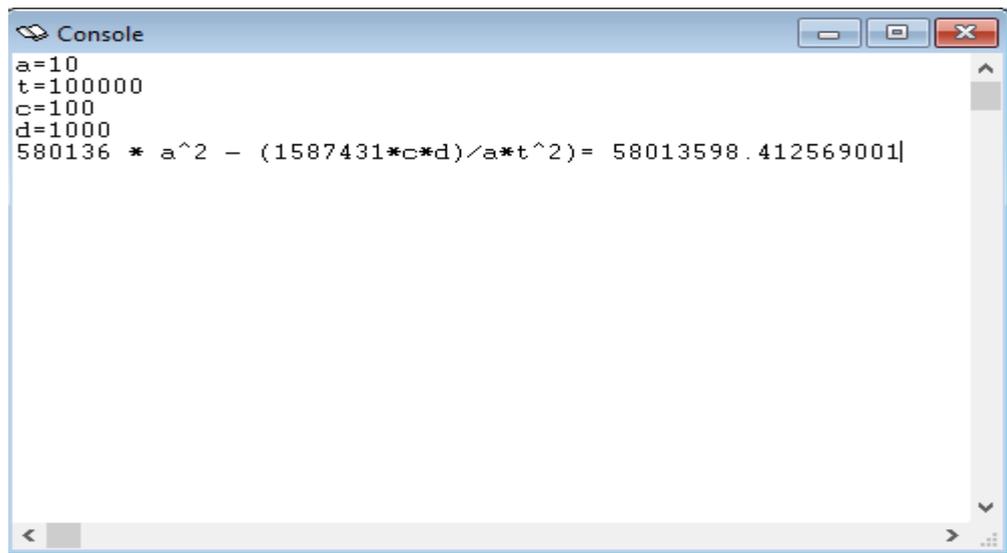
l.d $f12, x
li $v0, 3
syscall
#Выход в ОС
li $v0, 10
syscall
.end main

```

Тестирование программы

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

Пример работы программы изображён на рисунке 14.



```
Console
a=10
t=100000
c=100
d=1000
580136 * a^2 - (1587431*c*d)/a*t^2)= 58013598.412569001
```

Рисунок 14 – Пример работы программы расчёта арифметического выражения

Варианты работ для самостоятельного выполнения:

Те же самые, что и в лабораторной работе №1.

Лабораторная работа №4

Работа с числами с плавающей запятой на процессоре MIPS

Цель работы: Написать программу вычисления результата математического выражения средствами языка программирования Assembler для RISC процессора.

Задание: Вычислить результат выражения $z = ((1455.638 * d - x^3) / d^3) + 4500.341 * x * y$

Блок-схема программы предоставлена на рисунке 15.

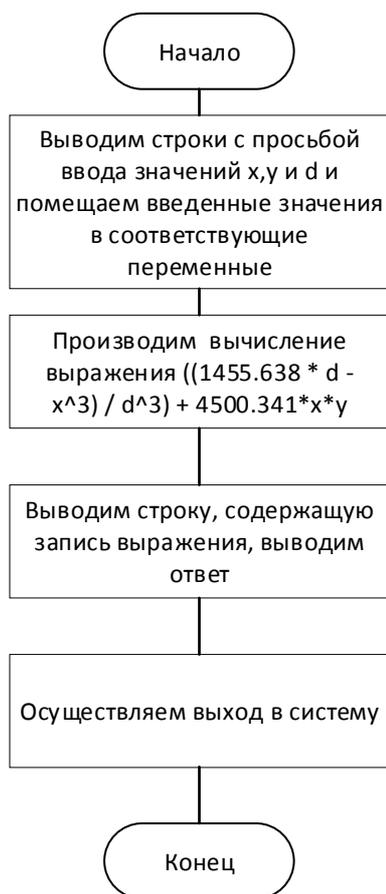


Рисунок 15 – Блок-схема процедуры main

Программный код

```

.data
z: .double 0.0
x: .double 0.0
x1: .double 0.0
y: .double 0.0
d: .double 0.0
d1: .double 0.0
d2: .double 0.0
ch1: .double 1455.638
ch2: .double 4500.341
psvk: .asciiz "\n"
strx: .asciiz "x="
stry: .asciiz "y="
strd: .asciiz "d="
strz: .asciiz "((1455.638 * d - x^3) / d^3) + 4500.341*x*y="
.text
main:
    #Ввод x
    la $a0,strx
    li $v0,4
    syscall
    li $v0,7
    syscall
    s.d $f0, x
    # Ввод y
    la $a0,stry
    # Вычисление части выражения: d^3
    l.d $f4, d
    mul.d $f4, $f4, $f4
    s.d $f4, d2
    l.d $f4, d
    l.d $f6,d2
    mul.d $f6, $f6, $f4
    s.d $f6, d2
    # Вычисление части выражения: 1455.638 * d - x^3
    l.d $f4, d1
    l.d $f6, x1
    sub.d $f4, $f4, $f6
    s.d $f4, d1
    # Вычисление части выражения: (1455.638 * d -
    x^3) / d^3
    l.d $f4, d1
    l.d $f6, d2
    div.d $f4, $f4, $f6
    s.d $f4, d2
    # Вычисление части выражения: x*y
    l.d $f4, x
    l.d $f6, y
    mul.d $f6, $f6, $f4
    s.d $f6, y
    # Вычисление части выражения: 4500.341*x*y
    l.d $f4, ch2
    l.d $f6, y
  
```

```

li $v0,4
syscall
li $v0,7
syscall
s.d $f0, y
# Ввод d
la $a0, strd
li $v0,4
syscall
li $v0,7
syscall
s.d $f0, d
#Вычисление части выражения: x^3
l.d $f4, x
mul.d $f4, $f4, $f4
s.d $f4, x1
l.d $f4, x
l.d $f6, x1
mul.d $f6, $f6, $f4
s.d $f6, x1
# Вычисление части выражения: 1455.638 * d
l.d $f4, ch1
l.d $f6, d
mul.d $f6, $f6, $f4
s.d $f6, d1

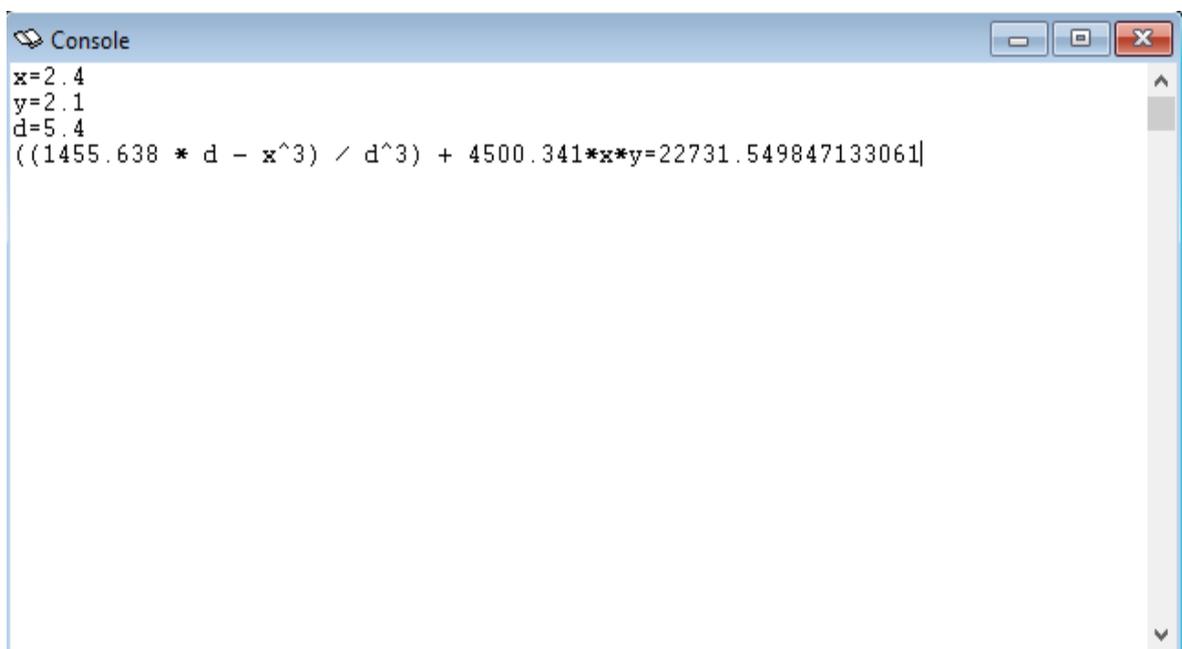
mul.d $f6, $f6, $f4
s.d $f6, y
# Вычисление выражения ((1455.638 * d - x^3) /
d^3) + 4500.341*x*y
l.d $f4, d2
l.d $f6, y
add.d $f4, $f4, $f6
s.d $f4, z
#Вывод строки с выражением и ответа
la $a0, strz
li $v0,4
syscall
l.d $f12, z
li $v0, 3
syscall
#Выход в ОС
li $v0, 10
syscall
.end main

```

Тестирование программы

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

Пример работы программы изображён на рисунке 16.



```

Console
x=2.4
y=2.1
d=5.4
((1455.638 * d - x^3) / d^3) + 4500.341*x*y=22731.549847133061

```

Рисунок 16 – Пример работы программы расчёта арифметического выражения

Варианты работ для самостоятельного выполнения:

Те же самые, что и в лабораторной работе №2.

3 Методические указания по самостоятельной работе

Для успешного освоения курса «Архитектура вычислительных систем» 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 «Архитектура вычислительных систем» проводится в форме экзамена. Для подготовки к итоговому контролю знаний по дисциплине «Архитектура вычислительных систем» 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>

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. 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://tverhthk.ru/library/predmets/vichislitelnaa_tehnika/Arhitektura_vychislitelnyh_sistem_i_kompjutersnyh_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>

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: ➤ № 41ГК/13 от 25.12.2012 г.; ➤ № 33ГК/13 от 24.12.2013 г.
Офисный пакет	Microsoft Office	
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
Интерпретатор команд языка программирования	SPIM	Свободное ПО, http://spimsimulator.sourceforge.net/

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.