

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УДМУРТСКОЙ РЕСПУБЛИКИ

**Автономное профессиональное образовательное учреждение Удмуртской Республики
«Техникум радиоэлектроники и информационных технологий»**

**Практические работы
по МДК.02.03 «Ассемблер для микропроцессорных систем»**

Разработали
преподаватель:

А.Н. Тетерин

Ижевск, 2016

Assembler. Практическая работа №1. “Разработка линейных программ”.

Цели:

- Научиться использовать инструментарий создания программ на Ассемблере;
- Научиться составлять простейшие линейные программы на Ассемблере.

Ход работы:

1. Изучить (повторить) теоретический материал.
4. Получить (уточнить) у преподавателя номер своего варианта.
5. Выбрать из списка номер вашего варианта:
6. Прочитать условие задачи, если возникли вопросы по условию, задайте их преподавателю. В этой работе и во всех последующих данные занимают слово (целые со знаком).
7. Внимательно изучить пример программы.
8. Разработать программу для своего варианта, выполнив последовательно все действия, подробно описанные в «Средства разработки программ».
9. Протестировать программу. Записать в отчет заданные исходные данные и полученные результаты (в шестнадцатеричной и десятичной системе счисления).
10. Оформить отчет по лабораторной работе - в начале asm-файла записать как комментарий:
 - номер, название, цель работы;
 - номер варианта, условие задачи;
 - результаты тестирования программы (значения исходных данных и полученные результаты в шестнадцатеричной и десятичной системе счисления).
 - дата окончания работы, Фамилия, Имя, группа.
11. Сдать преподавателю выполненную работу.

Пример программы:

```
; Составить программу для вычисления выражения:  $x = y + (z - u * v) / w$ ; данные занимают слово (целые со знаком); =====  
  
    .Model Small      ;определение модели памяти  
    .Stack 100h      ;начало сегмента стека  
  
    .Data            ;начало сегмента данных  
X    dw ?            ;определение данных  
y    dw 5            ;исходные данные  
z    dw -10  
u    dw 2  
w    dw -2  
v    dw 3  
  
    .Code            ;начало сегмента кода  
  
begin:  
    mov ax,@Data     ;загрузка в регистр ds адреса сегмента данных  
    mov ds,ax  
  
    mov ax,u  
    imul v           ;u*v  
    mov bx,ax  
    mov ax,z         ;z-u*v  
    sub ax,bx  
    cwd              ;подготовка к делению  
    idiv w           ;(z-u*v)/w  
    add ax,y         ;y+(z-u*v)/w  
    mov x,ax         ;запись результата в x  
  
    mov ah,4ch       ;выход из программы  
    int 21h  
  
end begin
```

Индивидуальные задания

Вариант 1.

Составить программу для вычисления выражения:

$$x = y - \frac{w + u \cdot v}{z}$$

Протестировать программу при следующих значениях исходных данных:

- a) $y = 1; w = 4; u = 3; v = 5; z = 2$
b) $y = 12; w = -2; u = 2; v = 3; z = -10$

Вариант 2.

Составить программу для вычисления выражения:

$$x = \frac{y + z}{u} + v \cdot w$$

Протестировать программу при следующих значениях исходных данных:

- a) $y = 2; w = 3; u = 10; v = 5; z = 18$
b) $y = 2; w = 3; u = 10; v = -2; z = 18$

Вариант 3.

Составить программу для вычисления выражения:

$$x = \frac{y \cdot z}{u} + v - w$$

Протестировать программу при следующих значениях исходных данных:

- a) $y = 1; w = 4; u = 2; v = 5; z = 3$
b) $y = -2; w = 4; u = 2; v = 5; z = 3$

Вариант 4.

Составить программу для вычисления выражения:

$$x = y + z - \frac{u \cdot v}{w}$$

Протестировать программу при следующих значениях исходных данных:

- a) $y = 12; w = 4; u = 8; v = 2; z = 10$
b) $y = 12; w = 4; u = 8; v = 2; z = -20$

Вариант 5.

Составить программу для вычисления выражения:

$$x = \frac{y}{z} + u \cdot v - w$$

Протестировать программу при следующих значениях исходных данных:

- a) $y = 10; w = 8; u = 4; v = 3; z = 2$
b) $y = 10; w = 40; u = 4; v = 3; z = 2$

Вариант 6.

Составить программу для вычисления выражения:

$$x = (y - z) \cdot u + \frac{v}{w}$$

Протестировать программу при следующих значениях исходных данных:

- a) $y = 5; w = 3; u = 2; v = 3; z = 1$
b) $y = 5; w = 3; u = -10; v = 3; z = 1$

Вариант 7.

Составить программу для вычисления выражения:

$$x = \frac{y - z + u}{v} + w$$

Протестировать программу при следующих значениях исходных данных:

- a) $y = 1; w = 5; u = 3; v = 4; z = 2$
- b) $y = 1; w = -12; u = 3; v = 4; z = 2$

Вариант 8.

Составить программу для вычисления выражения:

$$x = \frac{y}{z} + u \cdot v - w$$

Протестировать программу при следующих значениях исходных данных:

- a) $y = 5; w = 1; u = 3; v = 2; z = 4$
- b) $y = 15; w = 1; u = 3; v = 2; z = 4$

Вариант 9.

Составить программу для вычисления выражения:

$$x = -y + z + \frac{u \cdot v}{w}$$

Протестировать программу при следующих значениях исходных данных:

- a) $y = 2; w = -1; u = 8; v = 5; z = 3$
- b) $y = 2; w = -1; u = 8; v = 5; z = 10$

Вариант 10.

Составить программу для вычисления выражения:

$$x = (y - z) \cdot u + \frac{v}{w}$$

Протестировать программу при следующих значениях исходных данных:

- a) $y = 8; w = 5; u = 4; v = 1; z = 2$
- b) $y = 8; w = 15; u = 4; v = 1; z = 2$

Вариант 11.

Составить программу для вычисления выражения:

$$x = y - z + \frac{u \cdot v}{w}$$

Протестировать программу при следующих значениях исходных данных:

- a) $y = 12; w = 3; u = 2; v = -2; z = -10$
- b) $y = 12; w = 3; u = 2; v = -2; z = 10$

Вариант 12.

Составить программу для вычисления выражения:

$$x = \frac{y - z + u}{v} + w$$

Протестировать программу при следующих значениях исходных данных:

- a) $y = 12; w = 30; u = 20; v = -20; z = -100$
- b) $y = 12; w = 30; u = 20; v = -20; z = 100$

Assembler. Практическая работа №2. “Разработка программ с ветвлениями”.

Цели:

- Научиться программировать структуры ветвлений на Ассемблере.

Ход работы:

1. Изучить (повторить) теоретический материал.
4. Получить (уточнить) у преподавателя номер своего варианта.
5. На данную работу предлагается 2 задачи:
 - первая - программирование простых условий;
 - вторая - программирование составных условий.
6. Выбрать из списка номер вашего варианта:
7. Прочитать условие задач, если возникли вопросы по условию, задайте их преподавателю.
8. Внимательно изучить пример программы.
9. Разработать программы для своего варианта.
10. Протестировать программы.
11. Оформить отчет по лабораторной работе - в начале asm-файлов записать как комментарий:
 - номер, название, цель работы;
 - номер варианта, условие задачи;
 - результаты тестирования программы (значения исходных данных и полученные результаты в шестнадцатеричной и десятичной системе счисления.
 - дата окончания работы, Фамилия, Имя, группа.
12. Сдать преподавателю выполненную работу.

Пример программы:

; Составить программу вычисления функции у:

```
;          (a+b)/a,          a=b
;   y =    (a+b)/b,          a<b
;          (a+b)/(a*b), a>b
;=====
```

```
        .model small
        .Stack 100h
        .data
a       dw -1
b       dw -2
y       dw ?
;=====
```

```
        .code
begin:  mov ax,@data
        mov ds,ax
        mov ax,a
        add ax,b ;y:=a+b
        mov y,ax
        mov ax,a
        cmp ax,b
        je var1 ;на вариант a=b
        jl var2 ;на вариант a<b
var3:   mov ax,a ;a>b
        imul b
        mov bx,ax
        mov ax,y
        cwd
        idiv bx
        jmp exit
```

```

var2: mov ax,y      ;a<b
      cwd
      idiv b
      jmp exit
var1: mov ax,y      ;a=b
      cwd
      idiv a

exit:  mov y,ax     ;запись y
      mov ah,4ch   ;выход в ОС
      int 21h

end begin

```

Индивидуальные задания

Вариант 1.

1. Составить программу для вычисления функции $y(a,b)$:

$$y = \begin{cases} \frac{a+b}{a}, & a=b \\ \frac{4 \cdot a - b}{b}, & a < b \\ \frac{a + 2 \cdot b}{a \cdot b}, & a > b \end{cases}$$

Протестировать программу при следующих значениях исходных данных: а) $a = 2; b = 2$ б) $a = 1; b = 2$ в) $a = 1; b = -2$

2. Составить программу для вычисления функции $y(x)$:

$$y(x) = \begin{cases} x^2, & -10 \leq x \leq 5 \\ 2x, & 5 < x \leq 10 \\ x, & \text{иначе} \end{cases}$$

Протестировать программу при следующих значениях исходных данных: а) $x = 2$; б) $x = 6$; в) $x = 11$

Вариант 2.

1. Составить программу для вычисления функции $y(x)$:

$$y(x) = \begin{cases} x/2, & x - \text{четное} \\ 2x, & x - \text{нечетное} \end{cases}$$

Протестировать программу при следующих значениях исходных данных: а) $x = 4$; б) $x = 5$

2. Составить программу для вычисления функции $y(a,b)$:

$$y = \begin{cases} \frac{a \cdot b}{2}, & a < 0, b < 0 \\ \frac{a \cdot b}{3}, & a > 0, b > 0 \\ \frac{a \cdot b}{4}, & \text{иначе} \end{cases}$$

Протестировать программу при следующих значениях исходных данных: а) $a = -3; b = -4$ б) $a = 2; b = 2$ в) $a = 2; b = -4$

Вариант 3.

1. Составить программу для вычисления функции $y(x)$:

$$y(x) = \begin{cases} x/3, & x - \text{кратно } 3 - \text{м} \\ 3x, & \text{иначе} \end{cases}$$

Протестировать программу при следующих значениях исходных данных: а) $x = 6$; б) $x = 5$

2. Составить программу для вычисления функции $y(a,b)$:

$$y = \begin{cases} (a+b) \cdot a, & a < 0, b < 0 \\ (a+b) \cdot b, & a > 0, b > 0 \\ (a+b) \cdot b \cdot a, & \text{иначе} \end{cases}$$

Протестировать программу при следующих значениях исходных данных: а) $a = -2; b = -2$ б) $a = 2; b = 3$ в) $a = 2; b = -3$

Вариант 4.

1. Составить программу для вычисления функции $y(x)$:

$$y(x) = \begin{cases} \frac{x}{4}, & \text{если } x \text{ кратно } 4 \\ 4x, & \text{иначе} \end{cases}$$

Протестировать программу при следующих значениях исходных данных: а) $x = 8$; б) $x = 5$

2. Составить программу для вычисления функции $y(a,b)$:

$$y = \begin{cases} \frac{a-b}{a}, & a < 0, b > 0 \\ \frac{a-b}{b}, & a > 0, b < 0 \\ \frac{a+b}{a-b}, & \text{иначе} \end{cases}$$

Протестировать программу при следующих значениях исходных данных: а) $a = -2; b = 4$ б) $a = 2; b = -4$ в) $a = 6; b = 4$

Вариант 5.

1. Составить программу для вычисления функции $y(x)$:

$$y(x) = \begin{cases} x + 10, & \text{если } x < 0 \\ x - 10, & \text{иначе} \end{cases}$$

Протестировать программу при следующих значениях исходных данных: а) $x = -8$; б) $x = 25$

2. Составить программу для вычисления функции $y(a,b)$:

$$y = \begin{cases} a + b, & a < 0, b < 0 \\ a - b, & a > 0, b < 0 \\ a \cdot b, & \text{иначе} \end{cases}$$

Протестировать программу при следующих значениях исходных данных: а) $a = -2; b = -3$ б) $a = 5; b = -2$ в) $a = 2; b = 3$

Вариант 6.

1. Составить программу для вычисления функции $y(a,b)$:

$$y = \begin{cases} \frac{a}{a+b}, & a < b \\ \frac{b^2}{a+b}, & a > b \\ \frac{a \cdot b}{a+b}, & a = b \end{cases}$$

Протестировать программу при следующих значениях исходных данных: а) $a = -2; b = 4$ б) $a = 6; b = 5$ в) $a = 2; b = 2$

2. Составить программу для вычисления функции $y(x)$:

$$y(x) = \begin{cases} 2x^2 + 1, & -5 \leq x \leq 0 \\ 2x^2 - 1, & 0 < x \leq 10 \\ 2x^2, & \text{иначе} \end{cases}$$

Протестировать программу при следующих значениях исходных данных: а) $x = -2$; б) $x = 6$; в) $x = -6$

Вариант 7.

1. Составить программу для вычисления функции $y(x)$:

$$y(x) = \begin{cases} 2x + 1, & \text{если } x < 10 \\ 3x - 1, & \text{иначе} \end{cases}$$

Протестировать программу при следующих значениях исходных данных: а) $x = 8$; б) $x = 12$

2. Составить программу для вычисления функции $y(a,b)$:

$$y = \begin{cases} a^2 + b^2, & a \geq b, a > 0, b > 0 \\ a^2 - b^2, & a \leq b, a < 0, b < 0 \\ a^2 \cdot b^2, & \text{иначе} \end{cases}$$

Протестировать программу при следующих значениях исходных данных: а) $a = 2$; б) $a = -3$; в) $a = -2$; г) $a = -2$; б) $b = 3$

Вариант 8.

1. Составить программу для вычисления функции $y(x)$:

$$y(x) = \begin{cases} x^2 + 1, & x < 0 \\ 2x - 1, & x > 0 \\ 1, & x = 0 \end{cases}$$

Протестировать программу при следующих значениях исходных данных: а) $x = -2$; б) $x = 2$; в) $x = 0$

2. Составить программу для вычисления функции $y(a,b)$:

$$y = \begin{cases} \frac{b^3}{a^2}, & a \leq b, a > 0, b > 0 \\ \frac{b^2}{a^3}, & a \geq b, a < 0, b < 0 \\ \frac{b^2}{a^2}, & \text{иначе} \end{cases}$$

Протестировать программу при следующих значениях исходных данных: а) $a = 1$; б) $a = -1$; в) $a = -3$; г) $a = 2$; б) $b = 3$; в) $b = -4$

Вариант 9.

1. Составить программу для вычисления функции $y(a,b)$:

$$y = \begin{cases} 2 \cdot a + 3 \cdot b, & a < b \\ 3 \cdot b - 2 \cdot a, & a > b \\ 2 \cdot a^2 + b, & a = b \end{cases}$$

Протестировать программу при следующих значениях исходных данных: а) $a = 2$; б) $a = 1$; в) $a = -5$; г) $a = 2$; б) $b = 2$

2. Составить программу для вычисления функции $y(x)$:

$$y(x) = \begin{cases} x^2 + x - 1, & 0 \leq x \leq 5 \\ 2x^2 - x + 1, & 5 < x \leq 10 \\ 3x^2, & \text{иначе} \end{cases}$$

Протестировать программу при следующих значениях исходных данных: а) $x = -2$; б) $x = 2$; в) $x = 6$

Вариант 10.

1. Составить программу для вычисления функции $y(a,b)$:

$$y = \begin{cases} 12 \cdot a^2 + 8 \cdot b^2, & a < b \\ 10 \cdot a^2 - 4 \cdot b^2, & a > b \\ 15 \cdot a^2 - 2 \cdot b^2, & a = b \end{cases}$$

Протестировать программу при следующих значениях исходных данных: а) $a = 15; b = 16$ б) $a = 12; b = 10$ в) $a = 5; b = 5$

2. Составить программу для вычисления функции $y(x)$:

$$y(x) = \begin{cases} 2x + 2, & x \leq -5 \text{ или } x \geq 5 \\ 3x + 3, & -5 < x \leq 0 \\ x + 1, & 0 < x < 5 \end{cases}$$

Протестировать программу при следующих значениях исходных данных: а) $x = -5$; б) $x = 0$; в) $x = 2$

Вариант 11.

1. Составить программу для вычисления функции $y(a,b)$:

$$y = \begin{cases} \frac{(a-b)^2}{a+b}, & a > b \\ \frac{a+b}{a-b}, & a < b \\ \frac{a \cdot b}{a+b}, & a = b \end{cases}$$

Протестировать программу при следующих значениях исходных данных: а) $a = 4; b = 1$ б) $a = 1; b = 4$ в) $a = 4; b = 4$

2. Составить программу для вычисления функции $y(x)$:

$$y(x) = \begin{cases} x, & x \leq 0 \text{ или } x \geq 10 \\ x^3, & 0 < x \leq 5 \\ x^2, & 5 < x < 10 \end{cases}$$

Протестировать программу при следующих значениях исходных данных: а) $x = -5$; б) $x = 1$; в) $x = 6$

Вариант 12.

1. Составить программу для вычисления функции $y(x)$:

$$y(x) = \begin{cases} x^2 - 1, & x < 5 \\ 2x + 1, & x > 5 \\ x, & x = 5 \end{cases}$$

Протестировать программу при следующих значениях исходных данных: а) $x = 0$; б) $x = 6$; в) $x = 5$

2. Составить программу для вычисления функции $y(a,b)$:

$$y = \begin{cases} a+b, & a, b \text{ — четные} \\ a-b, & a, b \text{ — нечетные} \\ a \cdot b, & \text{иначе} \end{cases}$$

Протестировать программу при следующих значениях исходных данных: а) $a = 2; b = 4$ б) $a = 1; b = 3$ в) $a = 2; b = 5$

Assembler. Лабораторная работа №3.
“Разработка программ с циклами”.

Цели:

- Научиться программировать структуры циклов на Ассемблере.

Ход работы:

1. Изучить (повторить) теоретический материал: Тема 10. Программирование циклов.
 4. Получить (уточнить) у преподавателя номер своего варианта.
 5. На данную работу предлагается 2 задачи:
 - первая - программирование простого цикла (4 балла);
 - вторая - программирование вложенных циклов (6 баллов).
 6. Выбрать из списка номер вашего варианта:
 7. Прочитать условие задач, если возникли вопросы по условиям, задайте их преподавателю.
 8. Внимательно изучить примеры программ.
 9. Разработать программы для своего варианта.
 10. Протестировать программы.
 11. Оформить отчет по лабораторной работе - в начале asm-файла записать как комментарий:
 - номер, название, цель работы;
 - номер варианта, условие задачи;
 - результаты тестирования программы (значения исходных данных и полученные результаты в шестнадцатеричной и десятичной системе счисления).
 - дата окончания работы, Фамилия, Имя, группа.
9. Сдать преподавателю выполненную работу.
-

Пример программы:

;Вычисление суммы N первых четных ;чисел.

```
=====
.model small
.stack 100h
.data
N    dw 5 ;2+4+6+8+10
S    dw ?
=====
.code
begin: mov ax,@data
        mov ds,ax
        mov cx,N    ;cx:=N
        mov bx,2    ;bx:=2
        mov ax,0    ;ax:=0
for:    jcxz exit
        add ax,bx   ;ax:=ax+bx
        add bx,2    ;bx:=bx+2
        loop for   ;cx:=cx-1; если cx>0 на for:
exit:   mov S,ax    ;S:=ax
        mov ah,4ch
        int 21h
end    begin
```

Пример программы 2:

;Подбор натуральных x,у (одну комбинацию, ;наибольшее у при наибольшем x), ;удовлетворяющих равенству 2x+3y=12.

;Найденные x, у записать в память.

```
.model small
```

```

        .stack 100h
        .data
a       dw 2
b       dw 3
c       dw 12
x       dw ?
y       dw ?
        .code
begin:   mov ax,@data
         mov ds,ax
         mov cx,6           ;цикл по x от 6 до 1
for_x:  push cx
         mov cx,4           ;цикл по y от 4 до 1
for_y:  pop ax              ;ax:=x
         push ax            ;сохранить значение стека
         mul a              ;ax:=x*a
         mov bx,ax          ;bx:=ax
         mov ax,cx          ;ax:=y
         mul b              ;ax:=y*b
         add ax,bx          ;ax:=a*x+b*y
         cmp ax,c           ;если ax=c то
         je break          ;перейти на break
         loop for_y        ;y:=y-1,если y>0 перейти на for_y
         pop cx            ;cx:=x (из стека)
         loop for_x        ;x:=x-1, если x>0
         jmp exit          ;перейти на for_x
         ; => перейти на exitbreak:
         mov y,cx          ;запись найденной
         ;комбинации в память
         pop ax            ;y=2
         mov x,ax          ;x=3
exit:   mov ah,4ch ;выход в ОС
        int 21h
end     begin

```

Индивидуальные задания

Вариант 1.

1. Составить программу вычисления суммы первых N натуральных четных чисел. Протестировать программу при N=5.
2. Составить программу, подбирающую натуральные X и Y (одну комбинацию, наибольшее Y при наибольшем X), удовлетворяющие равенству: $2 \cdot X + 3 \cdot Y = 120$. Найденные X, Y записать в память.

Вариант 2.

1. Составить программу вычисления произведения первых N натуральных четных чисел. Протестировать программу при N=4.
2. Составить программу, подбирающую натуральные X и Y (одну комбинацию, наименьшее Y при наименьшем X), удовлетворяющие равенству: $2 \cdot X + 3 \cdot Y = 120$. Найденные X, Y записать в память.

Вариант 3.

1. Составить программу вычисления суммы первых N натуральных нечетных чисел. Протестировать программу при N=5.

2. Составить программу, подбирающую натуральные X и Y (одну комбинацию, наименьшее Y при наименьшем X), удовлетворяющие равенству: $X^2 + Y^2 = 1201$. Найденные X, Y записать в память.

Вариант 4.

1. Составить программу вычисления произведения первых N натуральных нечетных чисел. Протестировать программу при $N=4$.
2. Составить программу вычисления двойной суммы:

$$\sum_{i=1}^n \sum_{j=1}^m (i + j)$$

Протестировать программу при: $n=3; m = 5$.

Вариант 5.

1. Составить программу вычисления суммы первых N натуральных чисел. Протестировать программу при $N=5$.
2. Составить программу вычисления двойной суммы:

$$\sum_{i=1}^n \sum_{j=1}^m (i - 2 * j)$$

Протестировать программу при: $n=3; m = 5$.

Вариант 6.

1. Составить программу вычисления произведения первых N натуральных чисел (факториала N). Протестировать программу при $N=5$.
2. Составить программу вычисления двойной суммы:

$$\sum_{i=1}^n \sum_{j=1}^m (3 * i - j)$$

Протестировать программу при: $n=3; m = 5$.

Вариант 7.

1. Составить программу вычисления суммы N натуральных чисел, хранящихся в памяти как массив A. Протестировать программу при: $N=5; A=(2, 5, 1, 3, 7)$.
2. Составить программу вычисления суммы произведения:

$$\sum_{i=1}^n \prod_{j=1}^m (3 * i - j)$$

Протестировать программу при: $n=3; m = 5$.

Вариант 8.

1. Составить программу вычисления произведения N натуральных чисел, хранящихся в памяти как массив. Протестировать программу при: $N=5; A=(2, 5, 1, 3, 7)$.
2. Составить программу вычисления суммы произведения:

$$\sum_{i=1}^n \prod_{j=1}^m (i + j)$$

Протестировать программу при: $n=3; m = 5$.

Вариант 9.

1. Составить программу вычисления суммы 6-ти байт памяти от адреса start_adr в сегменте данных. Протестировать программу при следующем определении в сегменте данных:
`start_adr db 'Program'`
2. Составить программу вычисления произведения суммы:

$$\prod_{j=1}^m \sum_{i=1}^n (3 * i - j)$$

Протестировать программу при: $n=3$; $m = 5$.

Вариант 10.

1. Составить программу вычисления произведения 6-ти байт памяти от адреса start_adr в сегменте данных. Протестировать программу при следующем определении в сегменте данных:
`start_adr db '14.03'`

2. Составить программу вычисления произведения суммы:

$$\prod_{j=1}^m \sum_{i=1}^n (i + j)$$

Протестировать программу при: $n=3$; $m = 2$.

Вариант 11.

1. Составить программу вычисления суммы 5-ти слов в памяти от адреса start_adr в сегменте данных. Протестировать программу при следующем определении в сегменте данных:
`start_adr dw F1h, A10h, Ah, ADh, 123h`

2. Составить программу вычисления двойного произведения:

$$\prod_{j=1}^m \prod_{i=1}^n (i + j)$$

Протестировать программу при: $n=3$; $m = 2$.

Вариант 12.

1. Составить программу вычисления произведения 5-ти слов в памяти от адреса start_adr в сегменте данных. Протестировать программу при следующем определении в сегменте данных:
`start_adr dw 1h, 2h, Ah, Bh, 100h`

2. Составить программу вычисления двойного произведения:

$$\prod_{j=1}^m \prod_{i=1}^n (2 * i + j)$$

Протестировать программу при: $n=3$; $m = 2$.

Assembler. Лабораторная работа №4.
“Разработка и использование подпрограмм”.

Цели:

- Научиться создавать и использовать подпрограммы на Ассемблере.

Ход работы:

1. Изучить (повторить) теоретический материал: Тема 11. Организация подпрограмм.
 2. Получить (уточнить) у преподавателя номер своего варианта.
 3. Выбрать из списка номер своего варианта:
 4. Внимательно прочитать условие задачи, если возникли вопросы по условию, задайте их преподавателю.
 5. Изучить примеры программ.
 6. Разработать программу для своего варианта.
 7. Протестировать программу.
 8. Оформить отчет по лабораторной работе - в начале asm-файла записать как комментарий:
 - номер, название, цель работы;
 - номер варианта, условие задачи;
 - результаты тестирования программы (значения исходных данных и полученные результаты в шестнадцатеричной и десятичной системе счисления).
 - дата окончания работы, Фамилия, Имя, группа.
 9. Сдать преподавателю выполненную работу.
-

Пример 1.

Составить программу вычисления:

$$y = n! - m!$$

Для вычисления факториала создать подпрограмму (процедуру Factor с возвращением результата).

Процедуру Factor записать в текущем файле; Передачу параметра в процедуру и возврат результата организовать через регистры cx и ax соответственно.

```
.model small
.stack 100h
.data
n    dw 5
m    dw 4
y    dw ?
.code
begin:  mov ax,@data
        mov ds,ax
        mov cx,m           ;cx:=m
        call Factor        ;вычислить m!
        mov bx,ax          ;bx:=m!
        mov cx,n           ;cx:=n
        call Factor        ;вычислить n!
        sub ax,bx          ;ax:=n!-m!
        mov y,ax           ;y:=ax
        mov ah,4ch         ;выход из программы
        int 21h

Factor proc           ;подпрограмма
        mov ax,1           ;ax:=1
for:    imul cx            ;ax:=ax*cx
        loop for           ;cx:=cx-1
        ret                ;возврат в основную программу
Factor endp
end begin
```

Пример 2.

Составить программу вычисления:

$$y = \sum_{i=1}^n 2 \cdot i - \sum_{i=1}^m 3 \cdot i$$

Для вычисления суммы создать процедуру Summa;

Процедуру Summa разместить во внешнем файле; Передачу параметров (предел суммирования) передать через стек;

Результат вернуть в регистре ax.

Текст основного файла (main.asm)

```
.model small
.stack 100h
extrn Summa:proc
.data
n    dw 7
m    dw 5
y    dw ?
.code
begin:  mov ax,@data
        mov ds,ax
        push m           ;передача параметра m
        call Summa      ;вычислить S2 (в ax)
        mov bx,3
        imul bx         ;ax:=ax*3
        mov bx,ax       ;bx:=3*S2
        push n          ;передача параметра n
        call Summa      ;вычислить S1 (в ax)
        mov dx,2
        imul dx         ;ax:=ax*2
        sub ax,bx       ;ax:=2*S1-3*S2
        mov y,ax        ;y:=ax
        mov ah,4ch      ;выход из программы
        int 21h
end begin
```

Текст внешнего файла (sum.asm)

```
.model small
.code
begin:
Summa proc
Public Summa
    push bp             ;"пролог"-для доступа
    mov bp,sp          ;к параметрам в стеке
    mov cx,[bp+4]      ;cx:=параметр
                        ;(предел суммирования)
    mov ax,0           ;ax:=0
for:    add ax,cx       ;ax:=ax+cx
        loop for       ;cx:=cx-1
        pop bp         ;возврат из процедуры и
        ret 2          ;очистка стека от параметра
Summa endp
end begin
```

Выполнить трансляцию файлов main.asm и sum.asm по отдельности;

Для компоновки дать команду: tlink main sum

Вариант 1.

Составить подпрограмму для вычисления суммы:

$$\sum_{i=1}^n (2 \cdot i \cdot x)$$

Подпрограмму разместить в текущем файле. Параметры (x, n) в подпрограмму передать через регистры. Результат вернуть в регистре ax. В основной программе организовать вызов подпрограммы для следующих наборов исходных данных:

a) $x = 2; n = 2$; b) $x = -2; n = 3$; c) $x = 7; n = 12$

Вариант 2.

Составить подпрограмму для вычисления суммы:

$$\sum_{i=0}^n ((2 \cdot i + 1) \cdot x)$$

Подпрограмму разместить во внешнем файле. Параметры (x, n) в подпрограмму передать через регистры. Результат вернуть в регистре ax. В основной программе организовать вызов подпрограммы для следующих наборов исходных данных:

a) $x = 2; n = 2$; b) $x = -2; n = 3$; c) $x = 7; n = 9$

Вариант 3.

Составить подпрограмму для вычисления суммы:

$$\sum_{i=1}^n (i^2 x)$$

Подпрограмму разместить в текущем файле. Параметры (x, n) в подпрограмму передать через стек. Результат вернуть в регистре ax. В основной программе организовать вызов подпрограммы для следующих наборов исходных данных:

a) $x = 2; n = 2$; b) $x = -2; n = 3$; c) $x = 9; n = 8$

Вариант 4.

Составить подпрограмму для вычисления суммы:

$$\sum_{i=1}^n (2^i x)$$

Подпрограмму разместить во внешнем файле. Параметры (x, n) в подпрограмму передать через стек. Результат вернуть в регистре ax. В основной программе организовать вызов подпрограммы для следующих наборов исходных данных:

a) $x = 2; n = 2$; b) $x = -2; n = 3$; c) $x = 10; n = 8$

Вариант 5.

Составить подпрограмму для вычисления суммы:

$$\sum_{i=1}^n (3 \cdot i \cdot x)$$

Подпрограмму разместить в текущем файле. Параметры (x, n) в подпрограмму передать через общую область памяти. Результат вернуть в регистре ax. В основной программе организовать вызов подпрограммы для следующих наборов исходных данных:

a) $x = 2; n = 2$; b) $x = -2; n = 3$; c) $x = 5; n = 8$

Вариант 6.

Составить подпрограмму для вычисления суммы:

$$\sum_{i=1}^n (i(x+1))$$

Подпрограмму разместить во внешнем файле. Параметры (x, n) в подпрограмму передать через общую область памяти. Результат вернуть в регистре ax. В основной программе организовать вызов подпрограммы для следующих наборов исходных данных:

a) $x = 2; n = 2$; b) $x = -2; n = 3$; c) $x = 5; n = 12$

Вариант 7.

Составить подпрограмму для вычисления суммы:

$$\sum_{i=1}^n (i \cdot (x-1))$$

Подпрограмму разместить в текущем файле. Параметры (x, n) в подпрограмму передать через

регистры. Результат вернуть в регистре ax. В основной программе организовать вызов подпрограммы для следующих наборов исходных данных:

a) $x = 2; n = 2$; b) $x = -2; n = 3$; c) $x = 15; n = 12$

Вариант 8.

Составить подпрограмму для вычисления суммы:

$$\sum_{i=0}^n ((2i + 1) \cdot (2x + 1))$$

Подпрограмму разместить во внешнем файле. Параметры (x, n) в подпрограмму передать через регистры. Результат вернуть в регистре ax. В основной программе организовать вызов подпрограммы для следующих наборов исходных данных:

a) $x = 2; n = 2$; b) $x = -2; n = 3$; c) $x = -20; n = 5$

Вариант 9.

Составить подпрограмму для вычисления суммы:

$$\sum_{i=1}^n (i + x)$$

Подпрограмму разместить в текущем файле. Параметры (x, n) в подпрограмму передать через стек. Результат вернуть в регистре ax. В основной программе организовать вызов подпрограммы для следующих наборов исходных данных:

a) $x = 2; n = 2$; b) $x = -2; n = 3$; c) $x = -12; n = 30$

Вариант 10.

Составить подпрограмму для вычисления суммы:

$$\sum_{i=1}^n (i - x)$$

Подпрограмму разместить во внешнем файле. Параметры (x, n) в подпрограмму передать через стек. Результат вернуть в регистре ax. В основной программе организовать вызов подпрограммы для следующих наборов исходных данных:

a) $x = 2; n = 2$; b) $x = -2; n = 3$; c) $x = -4; n = 7$

Вариант 11.

Составить подпрограмму для вычисления суммы:

$$\sum_{i=1}^n (i^2 + x)$$

Подпрограмму разместить во внешнем файле. Параметры (x, n) в подпрограмму передать через стек. Результат вернуть в регистре ax. В основной программе организовать вызов подпрограммы для следующих наборов исходных данных:

a) $x = 2; n = 2$; b) $x = -2; n = 3$; c) $x = -28; n = 5$

Вариант 12.

Составить подпрограмму для вычисления суммы:

$$\sum_{i=1}^n (i^2 - x)$$

Подпрограмму разместить в текущем файле. Параметры (x, n) в подпрограмму передать через регистры. Результат вернуть в регистре ax. В основной программе организовать вызов подпрограммы для следующих наборов исходных данных:

a) $x = 2; n = 2$; b) $x = -2; n = 3$; c) $x = 100; n = 10$

Assembler. Лабораторная работа №5.

«Программирование с использованием взаимодействия Паскаль - Ассемблер».

Цели:

- Научиться создавать комбинированные программы, совмещающая язык высокого уровня Паскаль и язык низкого уровня Ассемблер.

Ход работы:

1. Изучить (повторить) теоретический материал:

Тема 12. Связь ассемблера с языками высокого уровня.

2. Получить (уточнить) у преподавателя номер своего варианта.

3. Выбрать из списка номер своего варианта:

4. Внимательно прочитать условие задачи, если возникли вопросы по условию, задайте их преподавателю.

5. Внимательно изучить пример.

6. Разработать программу для своего варианта.

7. Протестировать программу.

8. Оформить отчет по лабораторной работе - в начале asm-файла записать как комментарий:

- номер, название, цель работы;
- номер варианта, условие задачи;
- результаты тестирования программы (значения исходных данных и полученные результаты в шестнадцатеричной и десятичной системе счисления).
- дата окончания работы, Фамилия, Имя, группа.

9. Сдать преподавателю выполненную работу.

Пример.

Составить программу вычисления:

$$y = \sum_{i=1}^n 2 \cdot i - \sum_{i=1}^m 3 \cdot i$$

Основную программу разработать на языке Паскаль;

Для вычисления суммы создать функцию Summa(n,k:integer):integer на Ассемблере;

Функцию Summa разместить во внешнем obj- файле;

Файл primer.pas:

```
Program PasAsm;
```

```
Function Summa(n,k:integer):integer;external;
```

```
Var n,m,y:integer;
```

```
{$L asmmod.obj} {подключение файла}
```

```
Begin
```

```
Write('Введите n,m: ');
```

```
Readln(n,m);
```

```
y:=Summa(n,2)-Summa(m,3);
```

```
Writeln('y=',y);
```

```
Readln;
```

```
End.
```

Файл asmmod.asm:

```
.model small
```

```
.code
```

```
begin:
```

```
Summa proc
```

```
Public Summa
```

```
push bp
```

```
mov bp,sp
```

```
mov cx,[bp+6] ;cx:=n
```

```
mov ax,0 ;ax:=0
```

```
for: add ax,cx ;ax:=ax+cx
```

```

loop for          ;cx:=cx-1
imul [bp+4]      ;ax:=ax*k
pop bp
ret 4
Summa endp
end begin

```

Индивидуальные задания

Вариант 1.

Составить программу для вычисления выражения:

$$y = \sum_{i=1}^n a \cdot i - \sum_{i=1}^m b \cdot i$$

Основную программу разработать на языке Паскаль, в ней организовать ввод исходных данных и вывод результата. Для вычисления суммы разработать подпрограмму на Ассемблере и использовать её в основной программе. Протестировать программу при следующих значениях исходных данных: $a = 2$; $b = 3$; $n = 14$; $m = 3$

Вариант 2.

Составить программу для вычисления выражения:

$$y = \frac{n!}{m!(n-m)!}$$

Основную программу разработать на языке Паскаль, в ней организовать ввод исходных данных и вывод результата. Для вычисления факториала разработать подпрограмму на Ассемблере и использовать её в основной программе. Протестировать программу при следующих значениях исходных данных: $n = 12$; $m = 9$

Вариант 3.

Составить программу для вычисления выражения:

$$y = 2 \cdot (n! - m!)$$

Основную программу разработать на языке Паскаль, в ней организовать ввод исходных данных и вывод результата. Для вычисления факториала разработать подпрограмму на Ассемблере и использовать её в основной программе. Протестировать программу при следующих значениях исходных данных: $n = 7$; $m = 5$

Вариант 4.

Составить программу для вычисления выражения:

$$y = \prod_{i=1}^n (2i + 1) - \prod_{i=1}^m (2i)$$

Основную программу разработать на языке Паскаль, в ней организовать ввод исходных данных и вывод результата. Для вычисления произведения разработать подпрограмму на Ассемблере и использовать её в основной программе. Протестировать программу при следующих значениях исходных данных: $n = 5$; $m = 5$

Вариант 5.

Составить программу для вычисления выражения:

$$y = \frac{\sum_{i=1}^n (a \cdot i + b)}{\sum_{i=1}^m (c \cdot i + d)}$$

Основную программу разработать на языке Паскаль, в ней организовать ввод исходных данных и вывод результата. Для вычисления суммы разработать подпрограмму на Ассемблере и использовать её в основной программе. Протестировать программу при следующих значениях исходных данных: $a = 2$; $b = 1$; $c = 3$; $d = 2$; $n = 120$; $m = 10$

Вариант 6.

Составить программу для вычисления выражения:

$$y = \frac{n! + m!}{(n - m + 1)!}$$

Основную программу разработать на языке Паскаль, в ней организовать ввод исходных данных и вывод результата. Для вычисления факториала разработать подпрограмму на Ассемблере и использовать её в основной программе. Протестировать программу при следующих значениях исходных данных: $n = 7$; $m = 5$

Вариант 7.

Составить программу для вычисления выражения:

$$y = \frac{\prod_{i=1}^n (a \cdot i)}{\prod_{i=1}^m (b \cdot i)}$$

Основную программу разработать на языке Паскаль, в ней организовать ввод исходных данных и вывод результата. Для вычисления произведения разработать подпрограмму на Ассемблере и использовать её в основной программе. Протестировать программу при следующих значениях исходных данных: $a = 2$; $b = 1$; $n = 6$; $m = 4$

Вариант 8.

Составить программу для вычисления выражения:

$$y = \sum_{i=1}^n (3 \cdot i) - \sum_{i=n-m}^m i$$

Основную программу разработать на языке Паскаль, в ней организовать ввод исходных данных и вывод результата. Для вычисления суммы разработать подпрограмму на Ассемблере и использовать её в основной программе. Протестировать программу при следующих значениях исходных данных: $n = 40$; $m = 30$

Вариант 9.

Составить программу для вычисления выражения:

$$y = \sum_{i=n}^m (2i + 1) \cdot \sum_{i=k}^l 2i$$

Основную программу разработать на языке Паскаль, в ней организовать ввод исходных данных и вывод результата. Для вычисления суммы разработать подпрограмму на Ассемблере и использовать её в основной программе. Протестировать программу при следующих значениях исходных данных: $k = 4$; $l = 12$; $n = 3$; $m = 7$

Вариант 10.

Составить программу для вычисления выражения:

$$y = \sum_{i=1}^n ((3i)!) - \sum_{i=1}^m ((2i)!)$$

Основную программу разработать на языке Паскаль, в ней организовать ввод исходных данных и вывод результата. Для вычисления суммы и факториала разработать подпрограммы на Ассемблере и использовать их в основной программе. Протестировать программу при следующих значениях исходных данных: $n = 2$; $m = 2$

Вариант 11.

Составить программу для вычисления выражения:

$$y = \sum_{i=1}^n 2 \cdot i - \sum_{i=1}^m 3 \cdot i$$

Основную программу разработать на языке Паскаль, в ней организовать ввод исходных данных и вывод результата. Для вычисления суммы разработать подпрограмму на Ассемблере и использовать её в основной программе. Протестировать программу при следующих значениях исходных данных: $n = 14$; $m = 3$

Вариант 12.

Составить программу для вычисления выражения:

$$y = 3 \cdot n! - 2 \cdot m!$$

Основную программу разработать на языке Паскаль, в ней организовать ввод исходных данных и вывод результата. Для вычисления факториала разработать подпрограмму на Ассемблере и использовать её в основной программе. Протестировать программу при следующих значениях исходных данных: $n = 4$; $m = 3$

Assembler. Лабораторная работа №6.

“Организация ввода-вывода данных.

Обработка строк с использованием цепочечных команд”.

Цели:

- Научиться программировать ввод-вывод данных на ассемблере;
- Научится использовать цепочечные команды для обработки строковых данных на ассемблере.

Ход работы:

1. Изучить (повторить) теоретический материал:

Тема 13. Использование системных ресурсов. Организация ввода-вывода данных.

Тема 14. Цепочечные команды. Обработка строк.

2. Получить (уточнить) у преподавателя номер своего варианта.

3. Выбрать из списка номер своего варианта:

4. Внимательно прочитать условие задачи, если возникли вопросы по условию, задайте их преподавателю.

5. Внимательно изучить пример.

6. Разработать программу.

7. Протестировать программу.

8. Оформить отчет по лабораторной работе - в начале asm-файла записать как комментарий:

- номер, название, цель работы;
- номер варианта, условие задачи;
- результаты тестирования программы (значения исходных данных и полученные результаты в шестнадцатеричной и десятичной системе счисления.
- дата окончания работы, Фамилия, Имя, группа.

9. Сдать преподавателю выполненную работу.

Пример программы:

;Дана строка текста. Заменить в нем символ 'и' на 'i'.

;Подсчитать количество замен. Организовать ввод данных с клавиатуры

;и вывод результатов на экран.

;=====

.model small

.stack 100h

extrn write:proc ;внешняя процедура для вывода чисел

.data

;параметры для поля ввода:

Enter_String db 80 ;максимальная длина

Len_S db ? ;реальная длина

Str_Field db 80 dup(' ') ;поле ввода - цепочка

Ns db ' \$' ;поле для размещения числа в строковом представлении для последующего вывода на экран

S2 db 13,10,'Введите строку: \$'

S3 db 13, 10, 'Обработанная строка:',13,10,'\$'

S4 db 13,10,'Для завершения нажмите <Enter>.\$'

S5 db 13,10,'Количество произведенных замен <i> на <i> = \$'

k dw 0

public Enter_String,Len_S,Str_Field,Ns

;=====

.code

begin:

mov ax,@data ;инициализация сегментных регистров

mov ds,ax

mov es,ax

```

mov ah,9 ;вывод подсказки ввода
lea dx,S2
int 21h
mov ah,10 ;ввод строки
lea dx,Enter_String
int 21h
cld ;направление обработки по возрастанию адресов
lea di,Str_Field ;адрес цепочки
mov cl,Len_S ;длина цепочки
for: ;начало цикла
mov al,'и' ;образец поиска
jcxz vyvod ;если в cx ноль, то переход на вывод
repne scasb ;сканирование цепочки
je замена ;если символ-образец найден в цепочке, то перейти на блок обработки
(замены)
vyvod: ;вывод результата
mov al,'$' ;запись ограничителя вывода в конец цепочки
stosb
mov ah,9 ;вывод 'Обработанная строка:'
lea dx,S3
int 21h
mov ah,9 ;вывод обработанной строки-цепочки
lea dx,Str_Field
int 21h
mov ah,9 ;вывод 'Количество произведенных замен <i> на <i>='
lea dx,S5
int 21h
mov ax,k
call write ;вывод числа k
mov ah,9 ;вывод 'Для завершения нажмите <Enter>.'
lea dx,S4
int 21h
mov ah,10 ;ожидание нажатия клавиши <Enter>
lea dx,Str_Field
int 21h
jmp exit ;переход на выход из программы
замена: ;замена найденного символа
inc k
dec di
mov al,'i'
stosb
jmp for ;переход на начало цикла
exit: ;выход из программы
mov ah,4ch
int 21h
end begin

```

Индивидуальные задания

Вариант 1.

Составить программу, определяющую в строке количество заданных символов. Организовать ввод исходных данных (исходную строку и символ) с клавиатуры и вывод результата (количество символов) на экран. Для вывода численного значения можно использовать процедуру *write* из файла *c:tasm/bin/primery/rw.asm*.

Вариант 2.

Составить программу, заменяющую в строке один заданный символ на другой и определяющую количество произведенных замен. Организовать ввод исходных данных (исходную строку и два символа) с клавиатуры и вывод результатов (обработанную строку и количество замен) на экран. Для вывода численного значения можно использовать процедуру *write* из файла *c:tasm/bin/primery/rw.asm*.

Вариант 3.

Составить программу, заменяющую в строке два заданных символа (идущих подряд) на два других и подсчитывающую количество произведенных замен. Организовать ввод исходных данных (исходную строку и две пары символов) с клавиатуры и вывод результатов (обработанную строку и количество замен) на экран. Для вывода численного значения можно использовать процедуру *write* из файла *c:tasm/bin/primery/rw.asm*.

Вариант 4.

Составить программу, заменяющую в строке два заданных символа на один другой и подсчитывающую количество произведенных замен. Организовать ввод исходных данных (исходную строку, пару заменяемых символов и символ для замены) с клавиатуры и вывод результатов (обработанную строку и количество замен) на экран. Для вывода численного значения можно использовать процедуру *write* из файла *c:tasm/bin/primery/rw.asm*.

Вариант 5.

Составить программу, заменяющую в строке один заданный символ на два других и подсчитывающую количество произведенных замен. Организовать ввод исходных данных (исходную строку, заменяемый символ и пару символов для замены) с клавиатуры и вывод результатов (обработанную строку и количество замен) на экран. Для вывода численного значения можно использовать процедуру *write* из файла *c:tasm/bin/primery/rw.asm*.

Вариант 6.

Составить программу, удаляющую в строке заданный символ и подсчитывающую количество удалений. Организовать ввод исходных данных (исходную строку и удаляемый символ) с клавиатуры и вывод результатов (обработанную строку и количество удалений) на экран. Для вывода численного значения можно использовать процедуру *write* из файла *c:tasm/bin/primery/rw.asm*.

Вариант 7.

Составить программу, удаляющую в строке два, идущих подряд заданных символа и подсчитывающую количество удалений. Организовать ввод исходных данных (исходную строку и пару удаляемых символов) с клавиатуры и вывод результатов (обработанную строку и количество удалений) на экран. Для вывода численного значения можно использовать процедуру *write* из файла *c:tasm/bin/primery/rw.asm*.

Вариант 8.

Составить программу, вставляющую в строку за заданным символом другой заданный символ и подсчитывающую количество произведенных вставок. Организовать ввод исходных данных (исходную строку и два символа) с клавиатуры и вывод результатов (обработанную строку и количество вставок) на экран. Для вывода численного значения можно использовать процедуру *write* из файла *c:tasm/bin/primery/rw.asm*.

Вариант 9.

Составить программу, вставляющую в строку перед заданным символом два других заданных символа и подсчитывающую количество произведенных вставок. Организовать ввод исходных данных (исходную строку, символ и пару символов) с клавиатуры и вывод результатов (обработанную строку и количество вставок) на экран. Для вывода численного значения можно использовать процедуру *write* из файла *c:tasm/bin/primery/rw.asm*.

Вариант 10.

Составить программу, определяющую является ли заданная строка "перевертышом" (читается одинаково слева - направо и справа - налево). Организовать ввод исходных данных (исходную строку) с клавиатуры и вывод результата (сообщение "да" или "нет") на экран.

Вариант 11.

Составить программу, определяющую количество слов в заданной текстовой строке. Слова разделяются между собой одним, или несколькими пробелами. Организовать ввод исходных данных (исходную строку) с клавиатуры и вывод результатов (количество слов в строке) на экран.

Для вывода численного значения можно использовать процедуру *write* из файла *c:tasm/bin/primery/rw.asm*.

Вариант 12.

Составить программу, определяющую количество предложений в заданном тексте. Предложения разделяются между собой точкой, вопросительным, или восклицательным знаком. Организовать ввод исходных данных (исходный текст) с клавиатуры и вывод результата (количество предложений в тексте) на экран. Для вывода численного значения можно использовать процедуру *write* из файла *c:tasm/bin/primery/rw.asm*.
