Отчёт по лабораторной работе №6

Архитектура компьютера

Морозова Мария Вячеславовна

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Задание

Написать программу вычисления выражения y=f(x).

3 Теоретическое введение

Схема команды целочисленного сложения add (от англ. addition - добавление) выполняет сложение двух операндов и записывает результат по адресу первого операнда. Команда add работает как с числами со знаком, так и без знака и выглядит следующим образом: add , Команда целочисленного вычитания sub (от англ. subtraction – вычитание) работает анало- гично команде add и выглядит следующим образом: sub , Еще одна команда, которую можно отнести к арифметическим командам это команда изменения знака neg: neg Умножение и деление, в отличии от сложения и вычитания, для знаковых и беззнаковых чисел производиться по-разному, поэтому существуют различные команды. Для беззнакового умножения используется команда mul (от англ. multiply – умножение): mul Для знакового умножения используется команда imul: imul Для деления, как и для умножения, существует 2 команды div (от англ. divide - деление) и idiv: div ; Беззнаковое деление idiv ; Знаковое деление

4 Выполнение лабораторной работы

Создала каталог для программ лабораторной работы No 6, перешла в него и создала файл lab6-1.asm. (рис. 4.1).

Рис. 4.1: Создание каталога и файла

Создала исполняемый файл и запустила его. (рис. 4.2).

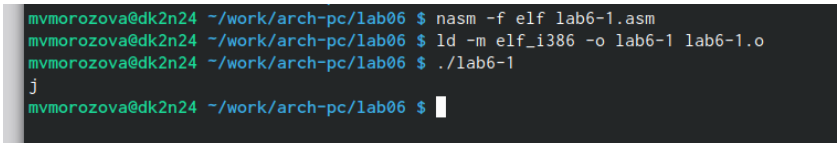

Рис. 4.2: Создание файла, запуск

Изменила текст программы и вместо символов, записала в регистры числа. (рис. 4.3).

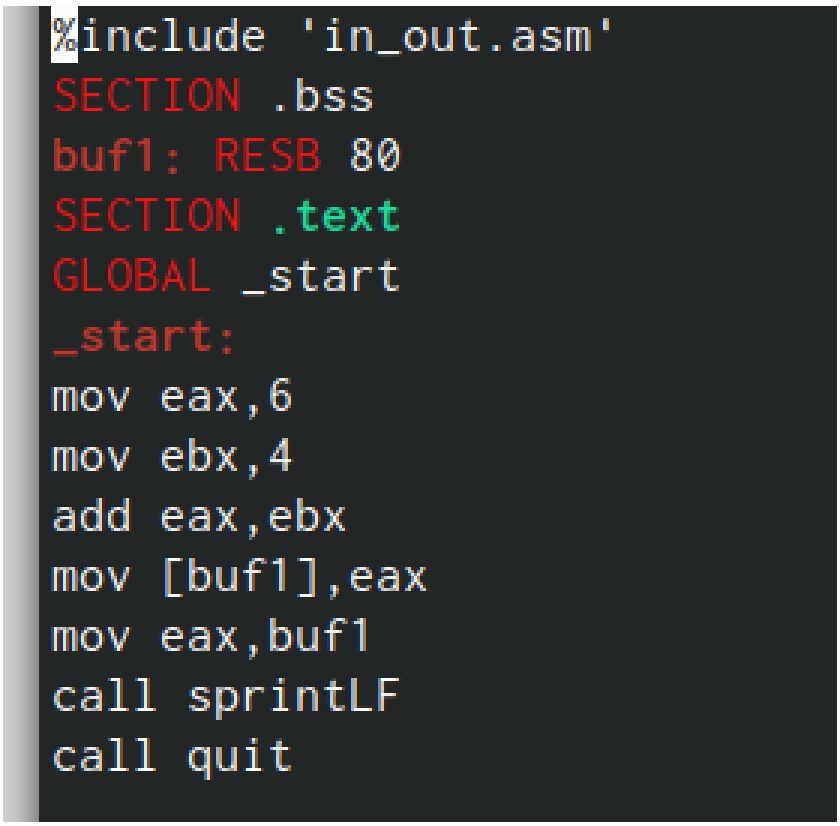

Рис. 4.3: Исправленный текст программы

Создала исполняемый файл и запустила его. (рис. 4.4).

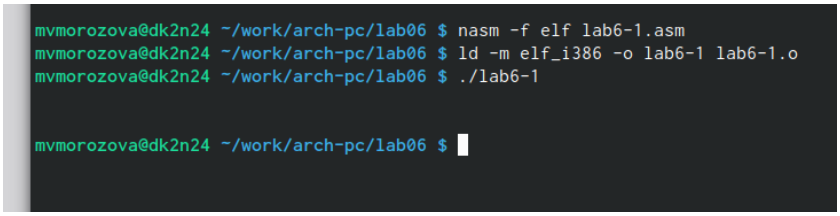

Рис. 4.4: Создание файла, запуск

Изменила текст программы. (рис. 4.5).

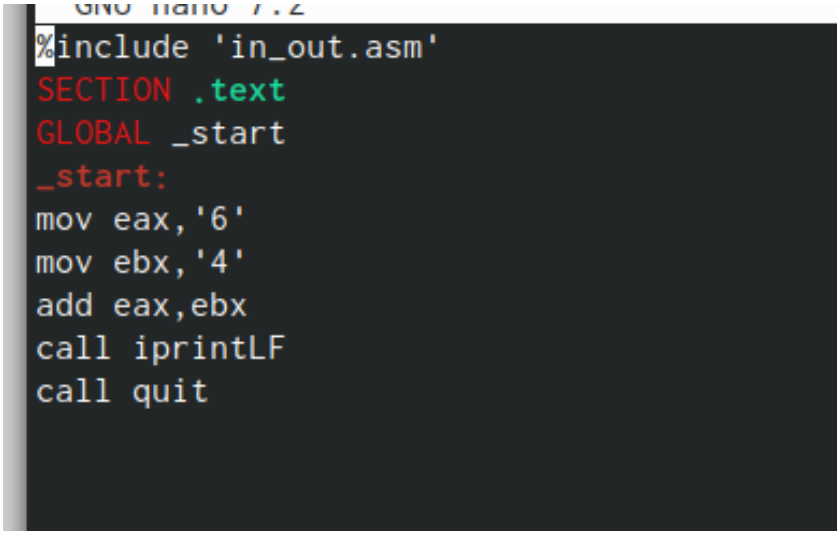

Рис. 4.5: Изменения

Создала файл, скомпоновала его и запустила. (рис. 4.6).

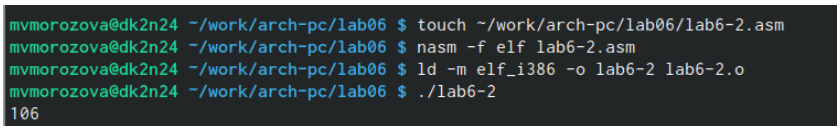

Рис. 4.6: Создание файла, запуск

Изменила текст программы. (рис. 4.7).

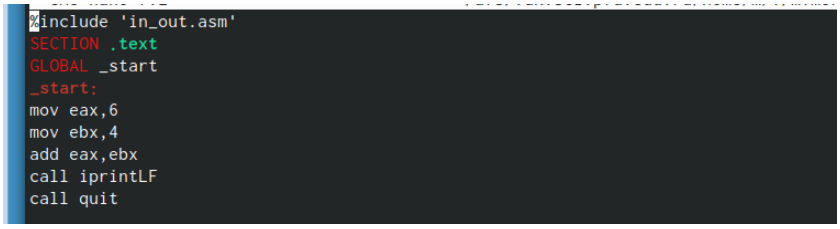

Рис. 4.7: Изменения

Создала исполняемый файл и запустила его. (рис. 4.8).

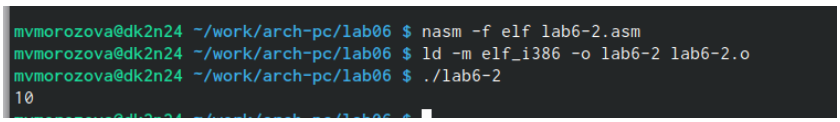

Рис. 4.8: Создание, запуск

Заменила iprintLF на iprint. (рис. 4.9).

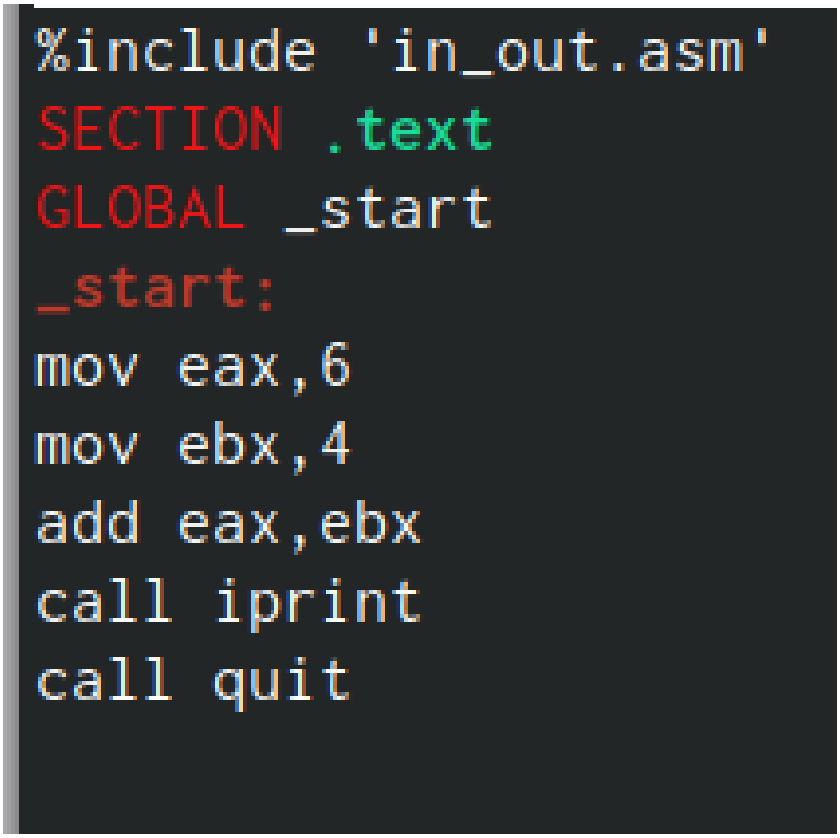

Рис. 4.9: Замена

Создала исполняемый файл и запустила его. (рис. 4.10).

Рис. 4.10: Создание, запуск

Создала файл lab6-3.asm в каталоге lab06. (рис. 4.11).

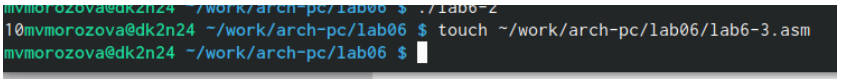

Рис. 4.11: Создание файла

Ввела текст программы из листинга. (рис. 4.12).

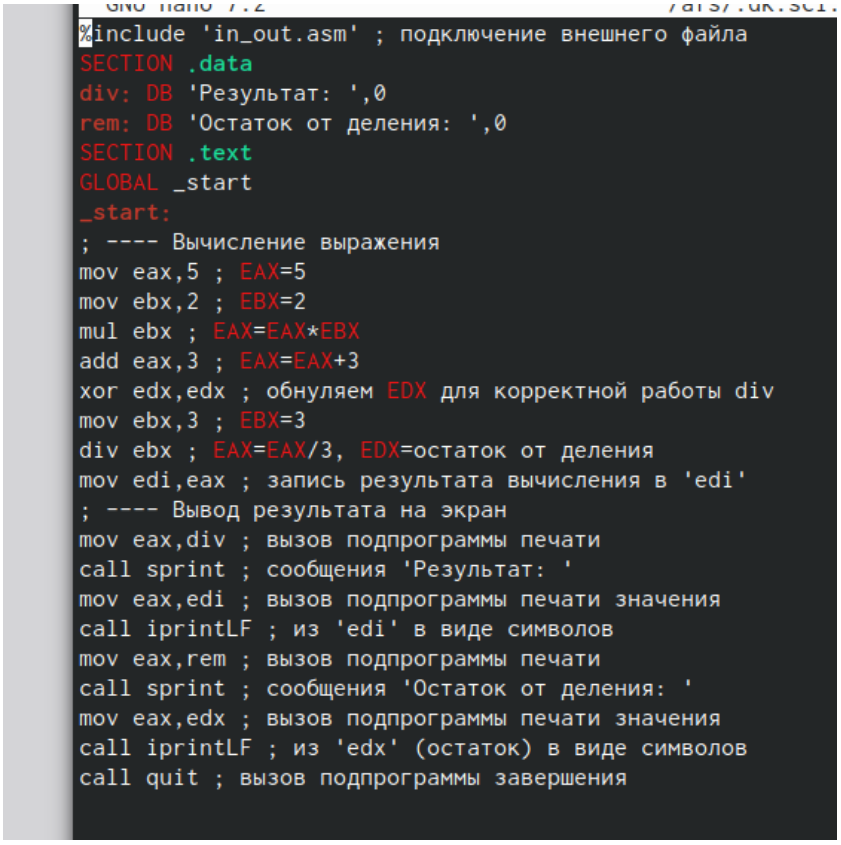

Рис. 4.12: Текст программы

Создала исполняемый файл и запустила его. (рис. 4.13).

```
mvmorozova@dk2n24 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
mvmorozova@dk2n24 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
mvmorozova@dk2n24 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 4
Остаток от деления: 1
mvmorozova@dk2n24 ~/work/arch-pc/lab06 $
```

Рис. 4.13: Компоновка, запуск

Изменила текст программы. (рис. 4.14).

```
 GNU nano 7.2                          /afs/.dk.sci.pfu.edu.ru/home/m/v/mvmo
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax,4 ; EAX=5
mov ebx,6 ; EBX=2
mul ebx ; EAX=EAX*EBX
add eax,2 ; EAX=EAX+3
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,5 ; EBX=3
div ebx ; EAX=EAX/3, EDX=остаток от деления
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax,div ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов
mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Остаток от деления: '
mov eax,edx ; вызов подпрограммы печати значения
call iprintLF ; из 'edx' (остаток) в виде символов
call quit ; вызов подпрограммы завершения
```

Рис. 4.14: Изменённый текст

Создала исполняемый файл, запустила его, чтобы проверить его работу. (рис. 4.15).

```
mvmorozova@dk2n24 ~/work/arch-pc/lab06 $ nasm -f elf lab6-3.asm
mvmorozova@dk2n24 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-3 lab6-3.o
mvmorozova@dk2n24 ~/work/arch-pc/lab06 $ ./lab6-3
Результат: 5
Остаток от деления: 1
```

Рис. 4.15: Создание файла, запуск

Создала файл variant.asm. (рис. 4.16).

```
mvmorozova@dk2n24 ~/work/arch-pc/lab06 $ touch ~/work/arch-pc/lab06/variant.asm
mvmorozova@dk2n24 ~/work/arch-pc/lab06 $
```

Рис. 4.16: Создание файла

Ввела текст из листинга. (рис. 4.17).

```
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите № студенческого билета: ',0
rem: DB 'Ваш вариант: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintLF
mov ecx, x
mov edx, 80
call sread
mov eax,x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, `eax=x`
xor edx,edx
mov ebx,20
div ebx
inc edx
mov eax,rem
call sprint
mov eax,edx
call iprintLF
call quit
```

Рис. 4.17: Текст программы

Создала исполняемый файл и запустила его, узнала номер варианта. (рис. 4.18).

```
mvmorozova@dk2n24 ~/work/arch-pc/lab06 $ nasm -f elf variant.asm
mvmorozova@dk2n24 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o variant variant.o
mvmorozova@dk2n24 ~/work/arch-pc/lab06 $ ./variant
Введите № студенческого билета:
1132236066
Ваш вариант: 7
mvmorozova@dk2n24 ~/work/arch-pc/lab06 $
```

Рис. 4.18: Создание файла, запуск

5 Ответы на вопросы

1. За вывод сообщения “Ваш вариант” отвечают строки: mov eax,rem call sprint
2. mov ecx, x - используют, чтобы положить адрес вводимой строки x в регистр ecx mov edx, 80 - запись в регистр edx длины вводимой строки call sread - вызов подпрограммы из внешнего файла, обеспечивающей ввод сообщения с клавиатуры
3. call atoi - вызов подпрограммы из внешнего файла, которая преобразует ascii-код символа в целое число и записывает результат в регистр eax
4. За вычисления варианта отвечают строки: xor edx,edx ; обнуление edx для корректной работы div mov ebx,20 ; ebx = 20 div ebx ; eax = eax/20, edx - остаток от деления inc edx ; edx = edx + 1
5. При выполнении инструкции div ebx остаток от деления записывается в регистр edx
6. Инструкция inc edx прибавляет 1 к значению регистра edx
7. За вывод на экран результатов вычислений отвечают строки: mov eax,edx call iprintLF

6 Выполнение самостоятельной работы

Написала программу для вычисления выражения y=f(x). (рис. 6.1).

```
call sprint ; вызов подпрограммы печати сообщения
mov ecx, x ; запись адреса переменной в ecx
mov edx, 80 ; запись длины вводимого значения в edx
call sread ; вызов подпрограммы ввода сообщения
mov eax,x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, `eax=x`
sub  eax,1; eax = eax-1 = x - 1
mov ebx,eax ; запись значения x-1 в регистр ebx
mul ebx; EAX=EAX*EBX = (x-1)*(x-1)
mov ecx,5 ; запись значения 5 в регистр ecx
mul ecx; EAX=EAX*ECX = (x-1)*(x-1)*5
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
```

Рис. 6.1: Текст программы

Проверка работы программы для x1. (рис. 6.2).

```
mvmorozova@dk8n81 ~/work/arch-pc/lab06 $ ./function
Введите значение переменной x: 3
Результат: 20mvmorozova@dk8n81 ~/work/arch-pc/lab06 $ mc
```

Рис. 6.2: Проверка

Проверка работы программы для x2. (рис. 6.3).

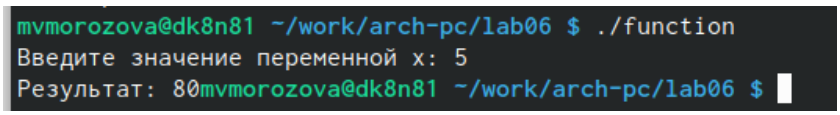

Рис. 6.3: Проверка

7 Листинги

```
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; секция инициированных данных
msg: DB 'Введите значение переменной x: ',0
rem: DB 'Результат: ',0
SECTION .bss ; секция не инициированных данных
x: RESB 80 ; Переменная, значение к-рой будем вводить с клавиатуры, выделенный ра
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
; ---- Вычисление выражения
mov eax, msg ; запись адреса выводимиого сообщения в eax
call sprint ; вызов подпрограммы печати сообщения
mov ecx, x ; запись адреса переменной в ecx
mov edx, 80 ; запись длины вводимого значения в edx
call sread ; вызов подпрограммы ввода сообщения
mov eax,x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, `eax=x`
sub  eax,1; eax = eax-1 = x - 1
mov ebx,eax ; запись значения x-1 в регистр ebx
mul ebx; EAX=EAX*EBX = (x-1)*(x-1)
mov ecx,5 ; запись значения 5 в регистр ecx
mul ecx; EAX=EAX*ECX = (x-1)*(x-1)*5
```

```
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprint ; из 'edi' в виде символов
call quit ; вызов подпрограммы завершения
```

8 Выводы

Были освоены арифметические инструкции языка ассемблера NASM.