Отчёт по лабораторной работе №6

Архитектура компьютера

Морозова Мария Вячеславовна

Содержание

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

# 2 Задание

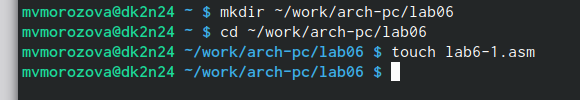
Написать программу вычисления выражения y=f(x).

# 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

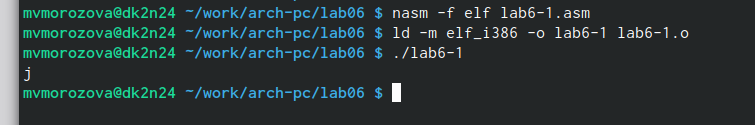
# 4 Выполнение лабораторной работы

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



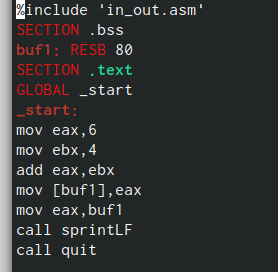
Создание каталога и файла

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



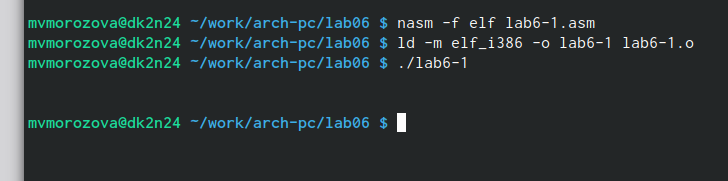
Создание файла, запуск

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



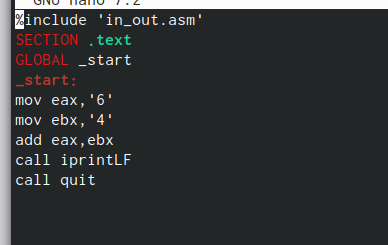
Исправленный текст программы

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



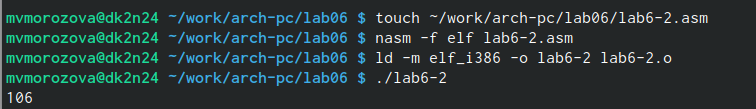
Создание файла, запуск

Изменила текст программы. (рис. ??).



Изменения

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



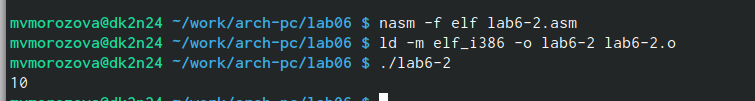
Создание файла, запуск

Изменила текст программы. (рис. ??).



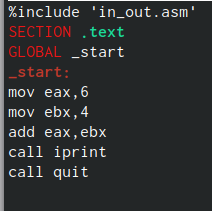
Изменения

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



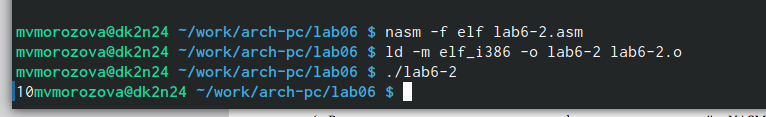
Создание, запуск

Заменила iprintLF на iprint. (рис. ??).



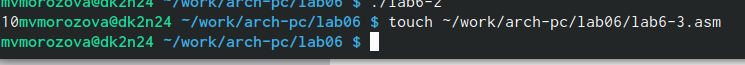
Замена

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



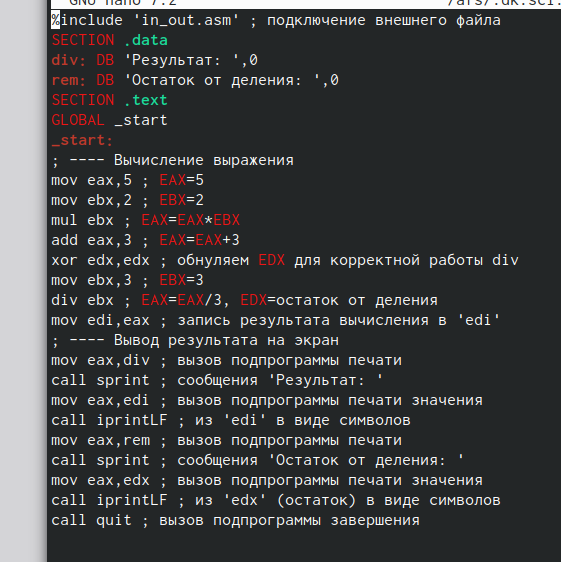
Создание, запуск

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



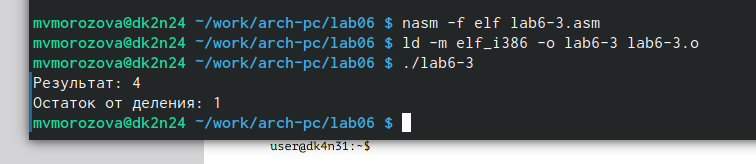
Создание файла

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



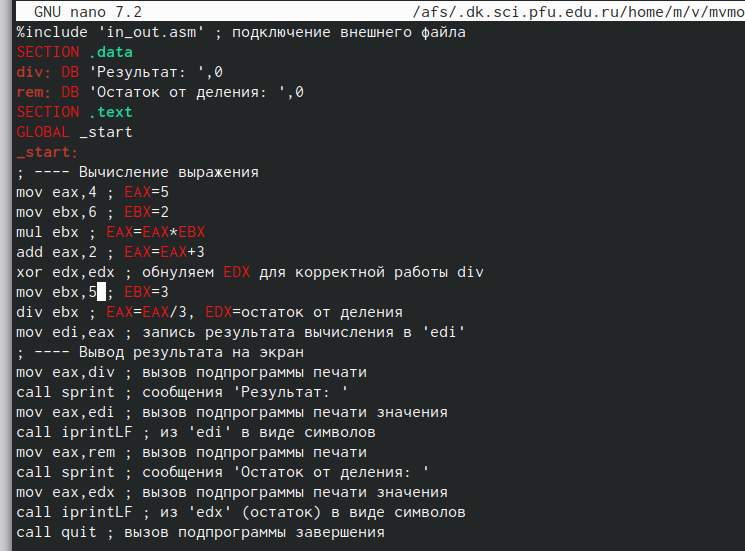
Текст программы

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



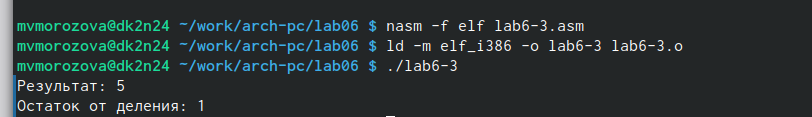
Компоновка, запуск

Изменила текст программы. (рис. ??).



Изменённый текст

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



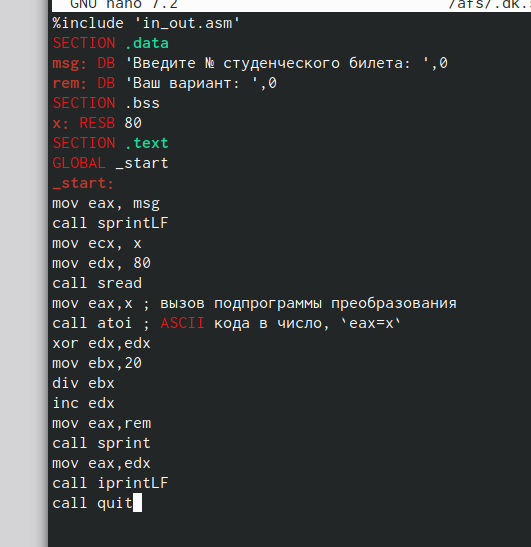
Создание файла, запуск

Создала файл variant.asm. (рис. ??).

Создание файла

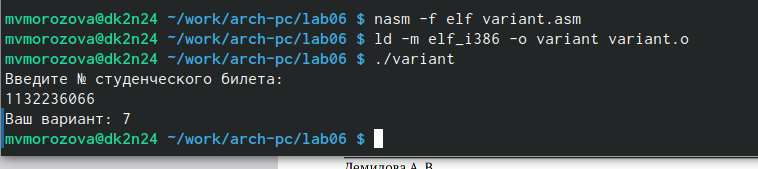
Создание файла

Ввела текст из листинга. (рис. ??).



Текст программы

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



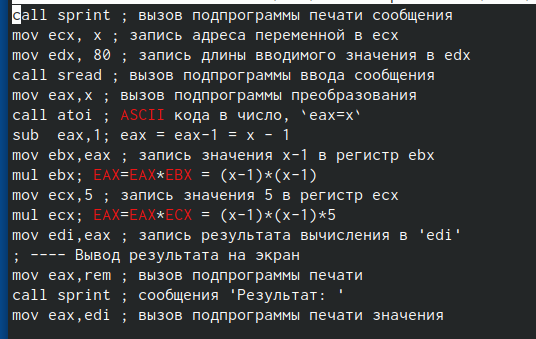
Создание файла, запуск

# 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

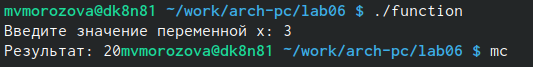
# 6 Выполнение самостоятельной работы

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



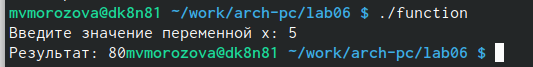
Текст программы

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



Проверка

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



Проверка

# 7 Листинги

%include 'in\_out.asm' ; подключение внешнего файла  
SECTION .data ; секция инициированных данных  
msg: DB 'Введите значение переменной х: ',0  
rem: DB 'Результат: ',0  
SECTION .bss ; секция не инициированных данных  
x: RESB 80 ; Переменная, значение к-рой будем вводить с клавиатуры, выделенный размер - 80 байт  
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.