Отчёта по лабораторной работе №6

Дисциплина: Архитектура компьютера

Камалиева Лия Дамировна.

Содержание

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM

# 2 Задание

1.1 Символьные и численные данные в NASM

1.2. Выполнение арифметических операций в NASM

1.3. Задание для самостоятельной работы

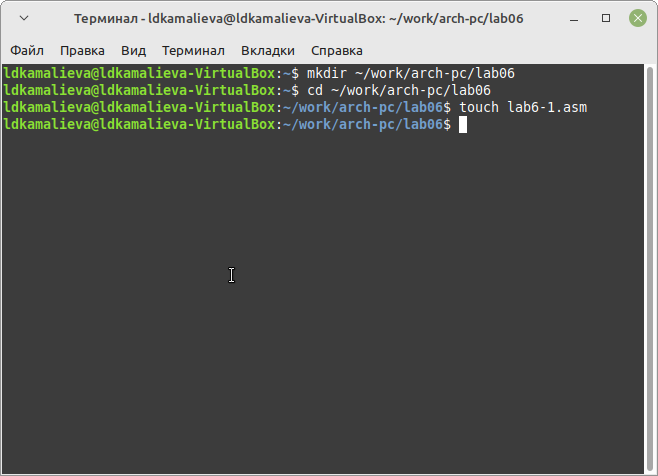
# 3 Теоретическое введение

Адресация в NASM Большинство инструкций на языке ассемблера требуют обработки операндов. Адрес операнда предоставляет место, где хранятся данные, подлежащие обработке. Это могут быть данные хранящиеся в регистре или в ячейке памяти. Далее рассмотрены все существующие способы задания адреса хранения операндов – способы адресации.

# 4 Выполнение лабораторной работы

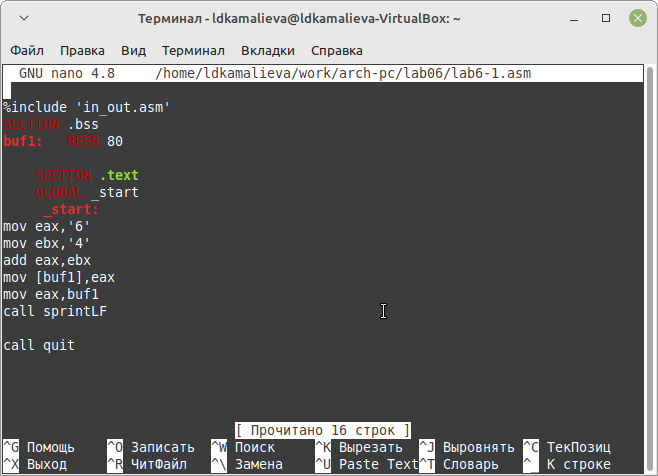
## 4.1 1.1 Символьные и численные данные в NASM

Шаг 1. Создаем каталог для программ лабраторной работы №6.



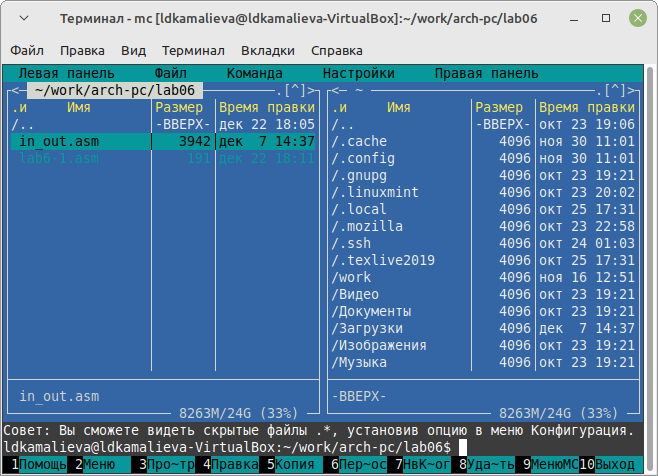
создание каталога

Шаг 2. Запишем программу вывода значения регистра eax



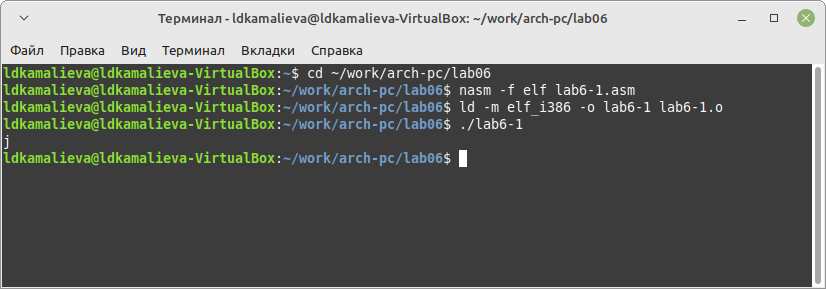
eax

Шаг 3. Для корректной работы программы подключаемый файл in\_out.asm



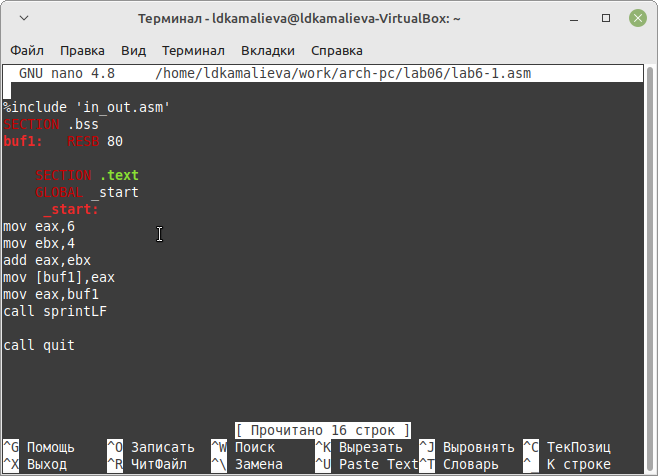
подключение in\_out.asm

Шаг 4.Создаем исполняемый файл и запускаем его



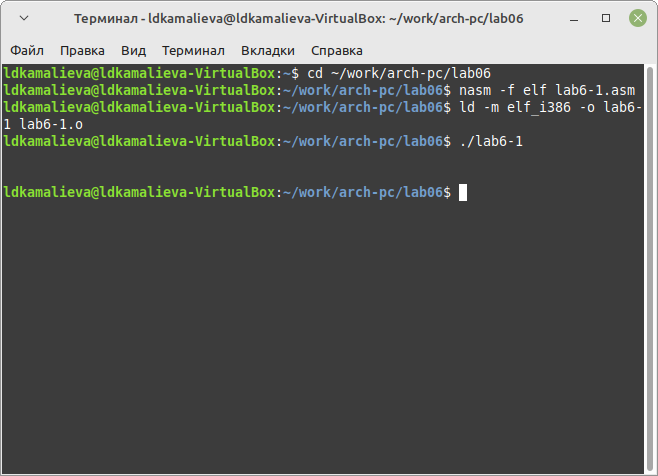
запуск

Шаг 5. Изменяем файл убираем ковычки



изменяем программу

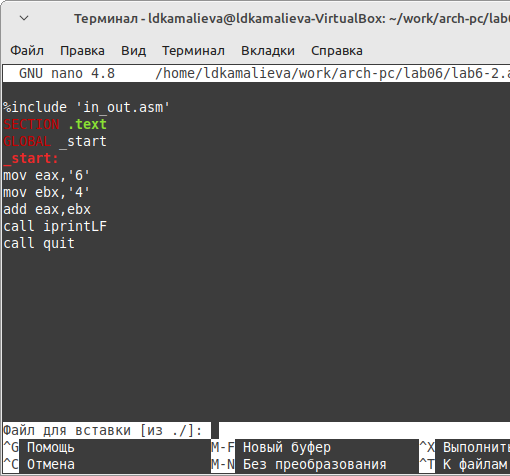
Шаг 6. Снова создаем исполняемый файл и запускаем его



запуск

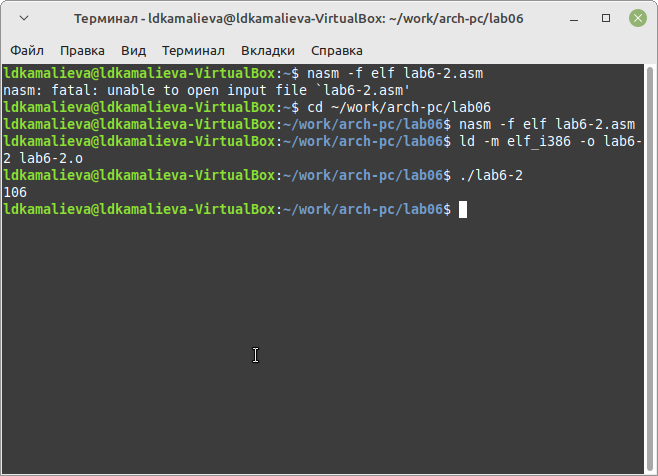
Шаг 7. Cоздаём файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и вводим в него текст программы из листинга

%include ‘in\_out.asm’ SECTION .text GLOBAL \_start \_start: mov eax,‘6’ mov ebx,‘4’ add eax,ebx call iprintLF call quit



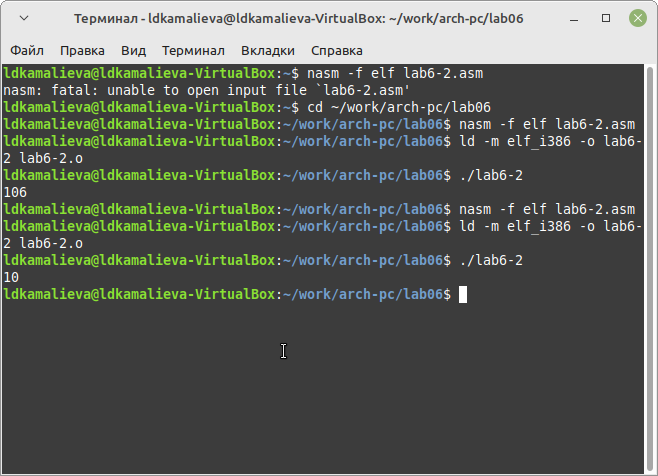
программа

Шаг 8. Проверяем программу



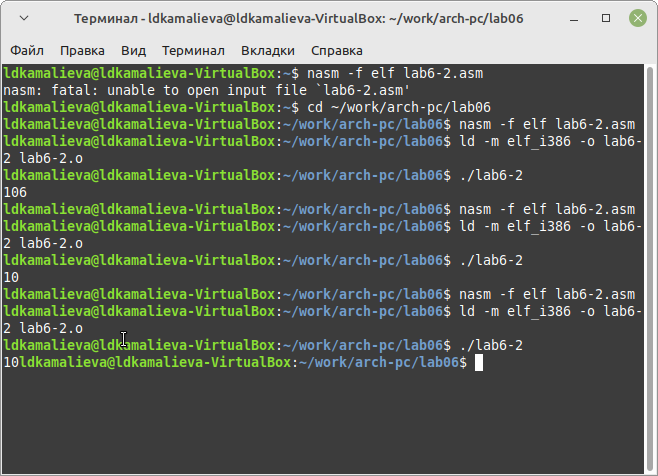
исполняемый файл

Шаг 9. Аналогично предыдущему примеру изменим символы на числа, создаём исполняемый файл и запускаем его



измененная программа

Шаг 10. Замените функцию iprintLF на iprint. Создаём исполняемый файл и запускаем его.



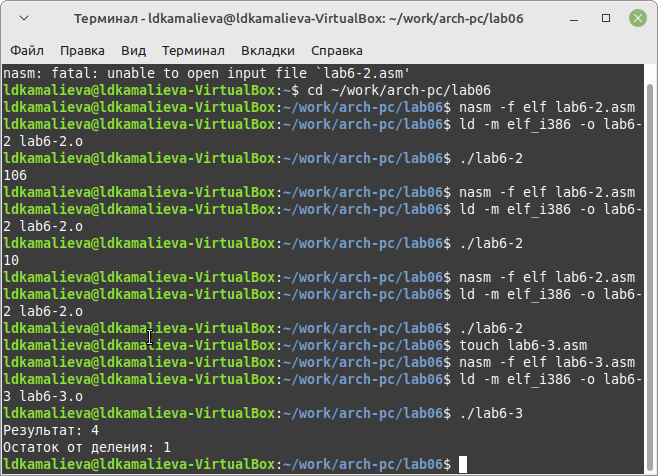
изменённая функция

## 4.2 1.2 Выполнение арифметических операций в NASM

### 4.2.1 В качестве примера выполнения арифметических операций в NASM приведем программу вычисления арифметического выражения F(x) = (5 \* 2 + 3)/3

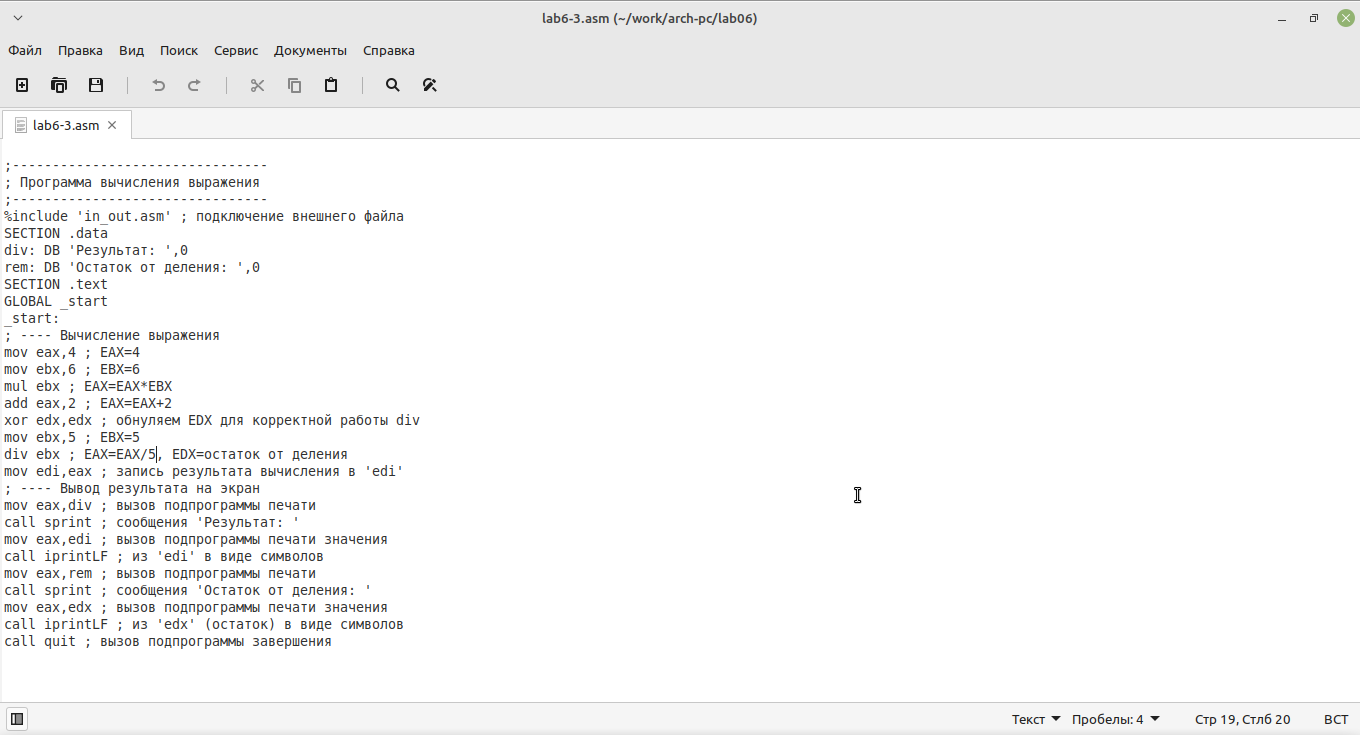
Шаг 1. Создаём файл lab6-3.asm в каталоге ~/work/arch-pc/lab06, пишем в нем программу и запускаем её.

%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,5 ; EAX=5 mov ebx,2 ; EBX=2 mul ebx ; EAX=EAX\*EBX add eax,3 ; EAX=EAX+3 xor edx,edx ; обнуляем EDX для корректной работы div mov ebx,3 ; 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



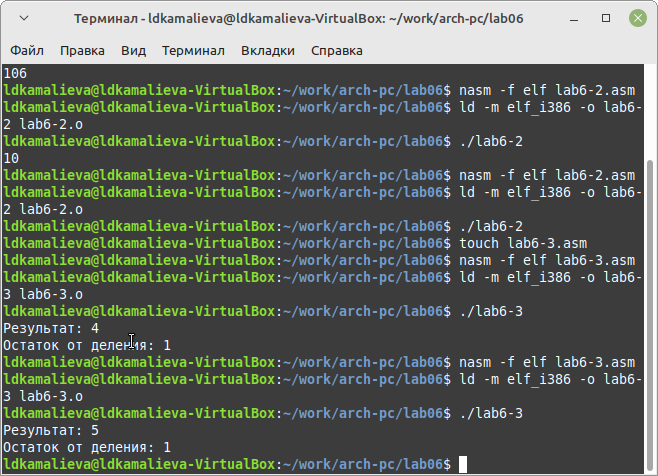
lab6-3.asm

Шаг 2. Изменяем текст программы для вычисления выражения 𝑓(𝑥) = (4 ∗ 6 + 2)/5.



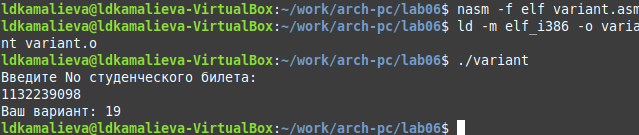
измененная программа lab6-3.asm

Шаг 3. Создаем исполняемый файл и проверьте его работу



запуск измененной программы lab6-3.asm

Шаг 4. Рассмотрим также программу, вычисляющую вриант задания по номеру студенческого билета, для этого создаем файл variant.asm. Создаём исполняемый файл и запускаем его.



запуск программы по расчету варианта

Ответы на вопросы: #### 1. Какие строки листинга 6.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’?

За вывод на экран отвечает строка: mov eax,rem, которая загружает в регистр адрес строки с текстом

#### 4.2.1.1 2. Для чего используется следующие инструкции?

mov ecx, x mov edx, 80 call sread

первая команда помещает x в регистр а вторая команда делает тоже самое, но с регистром edx и значением 80 команда call spread обеспечивает считывание студенчиского билета и присваивание его x

#### 4.2.1.2 3. Для чего используется инструкция “call atoi”?

эта функция переводит строковые данные в целочисленное значение

#### 4.2.1.3 4. Какие строки листинга 6.4 отвечают за вычисления варианта?

xor edx,edx mov ebx,20 div ebx inc edx

#### 4.2.1.4 5. В какой регистр записывается остаток от деления при выполнении инструкции “div ebx”?

edx, также мы его часто обнуляем

#### 4.2.1.5 6. Для чего используется инструкция “inc edx”?

команда inc edx увеличивает значение регистра на 1

####7. Какие строки листинга 6.4 отвечают за вывод на экран результата вычислений?

mov eax,edx call iprintLF

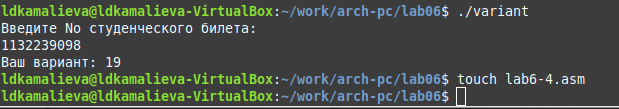
## 4.3 1.3 Задание для самостоятельной работы

Написать программу вычисления выражения y = F(x). Программа должна выводить выражение для вычисления, выводить запрос на ввод значения 𝑥, вычислять задан- ное выражение в зависимости от введенного 𝑥, выводить результат вычислений. Вид функции F(x) выбрать из таблицы 6.3 вариантов заданий в соответствии с номером полученным при выполнении лабораторной работы. Создайте исполняемый файл и проверьте его работу для значений x1 и x2 из 6.3 У меня это вариант №19

задание

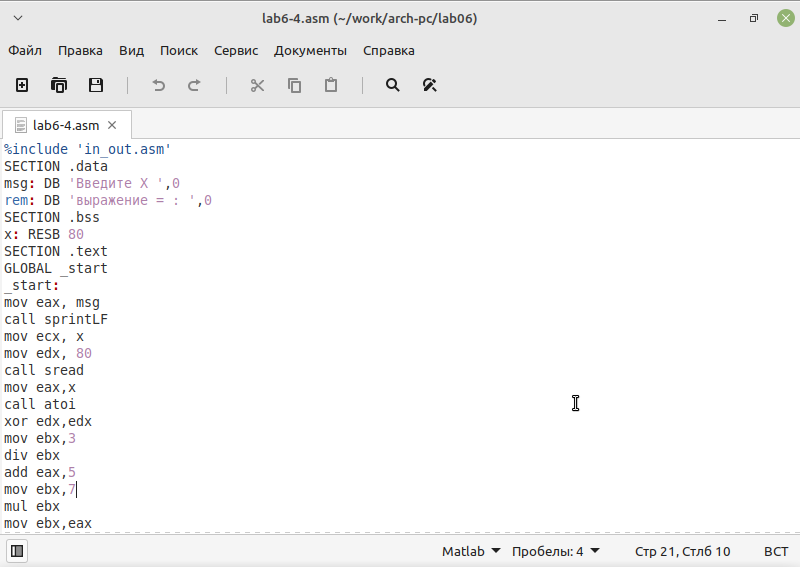
задание

Шаг 1. создаем файл lab6-4.asm



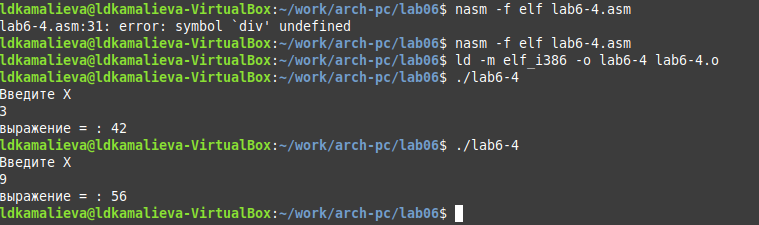
файл

Шаг 2. пишем программу по вычисению примера



программа

Шаг3. делаем проверку корней



проверка

# 5 Выводы

Вывод: я научилась писать арифметические инструкции в ассемблере NASM

# Список литературы