Отчёт по лабораторной работе №6

Дисциплина: архитектура компьютера

Бизев Никита Владимирович

Содержание

# 1 Цель работы

Цель данной лабораторной работы - освоение арифметческих инструкций языка ассемблера NASM.

# 2 Задание

1. Символьные и численные данные в NASM
2. Выполнение арифметических операций в NASM
3. Выполнение заданий для самостоятельной работы

# 3 Теоретическое введение

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. - Регистровая адресация – операнды хранятся в регистрах и в команде используются имена этих регистров, например: mov ax,bx. - Непосредственная адресация – значение операнда задается непосредственно в команде, Например: mov ax,2. - Адресация памяти – операнд задает адрес в памяти. В команде указывается символическое обозначение ячейки памяти, над содержимым которой требуется выполнить операцию.

Ввод информации с клавиатуры и вывод её на экран осуществляется в символьном виде. Кодирование этой информации производится согласно кодовой таблице символов ASCII. ASCII – сокращение от American Standard Code for Information Interchange (Американский стандартный код для обмена информацией). Согласно стандарту ASCII каждый символ кодируется одним байтом. Среди инструкций NASM нет такой, которая выводит числа (не в символьном виде). Поэтому, например, чтобы вывести число, надо предварительно преобразовать его цифры в ASCII-коды этих цифр и выводить на экран эти коды, а не само число. Если же выводить число на экран непосредственно, то экран воспримет его не как число, а как последовательность ASCII-символов – каждый байт числа будет воспринят как один ASCII-символ – и выведет на экран эти символы. Аналогичная ситуация происходит и при вводе данных с клавиатуры. Введенные данные будут представлять собой символы, что сделает невозможным получение корректного результата при выполнении над ними арифметических операций. Для решения этой проблемы необходимо проводить преобразование ASCII символов в числа и обратно

# 4 Выполнение лабораторной работы

## 4.1 Символьные и численные данные в NASM

Создаю файл lab6-1.asm (рис. 1).

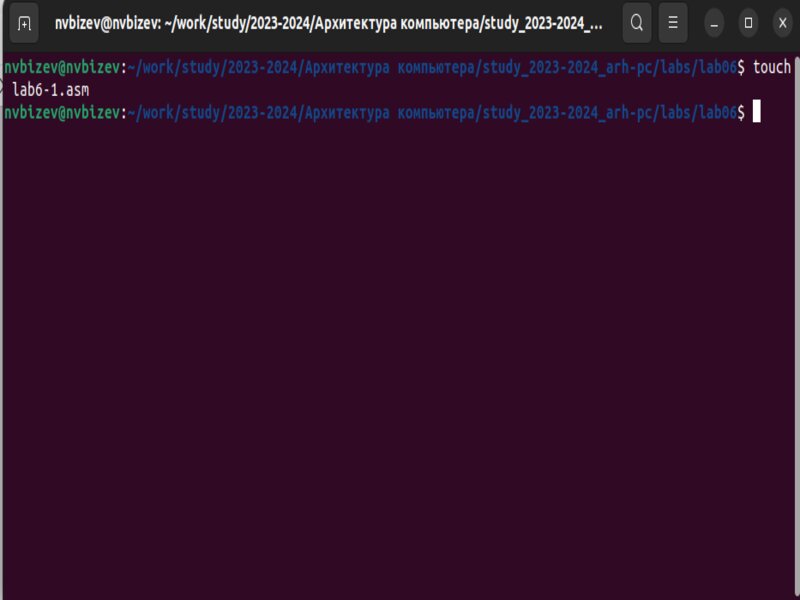


Рис. 1: Создание lab6-1.asm

Ввожу в файл lab6-1.asm текст программы из листинга 6.1 (рис. 2).

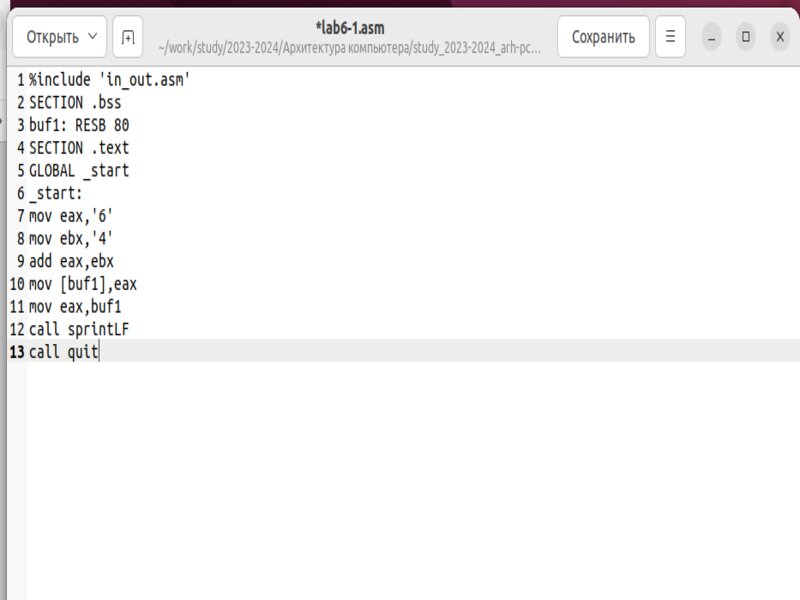


Рис. 2: Ввод данных из листинга в lab6-1.asm

Создаю исполняемый файл lab6-1.asm (рис. 3).

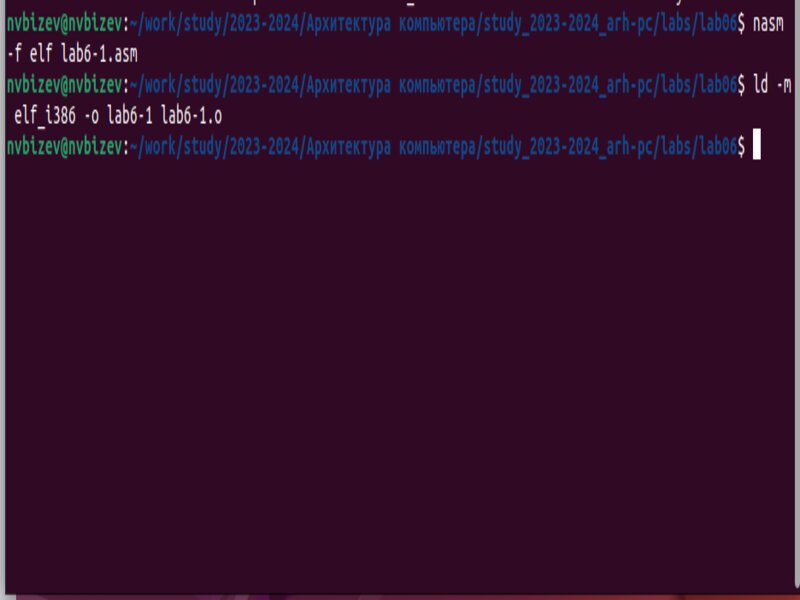


Рис. 3: Создание исполяемого файла

Запускаю файл. (рис. 4).

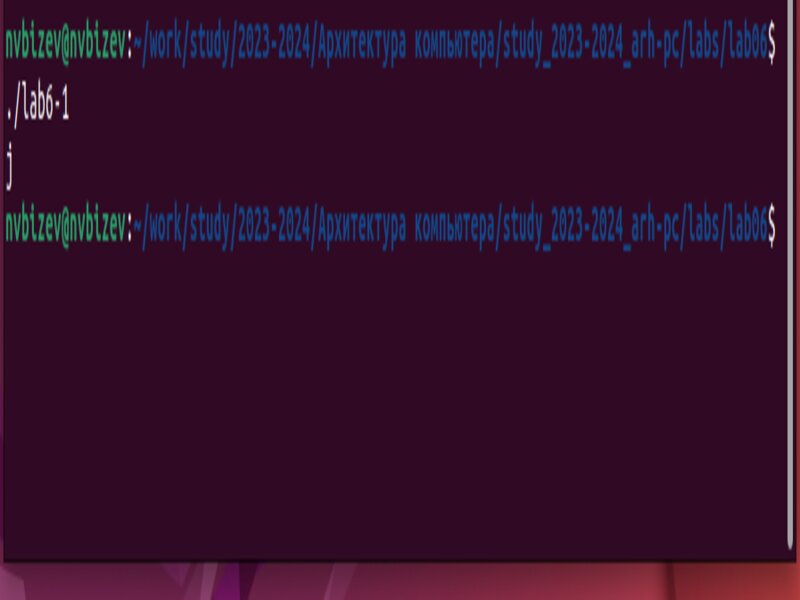


Рис. 4: Запуск файла

Создаю файл lab6-1-1.asm и изменяю в нем символы ‘6’, ‘4’ на числа 6 и 4. (рис. 5).

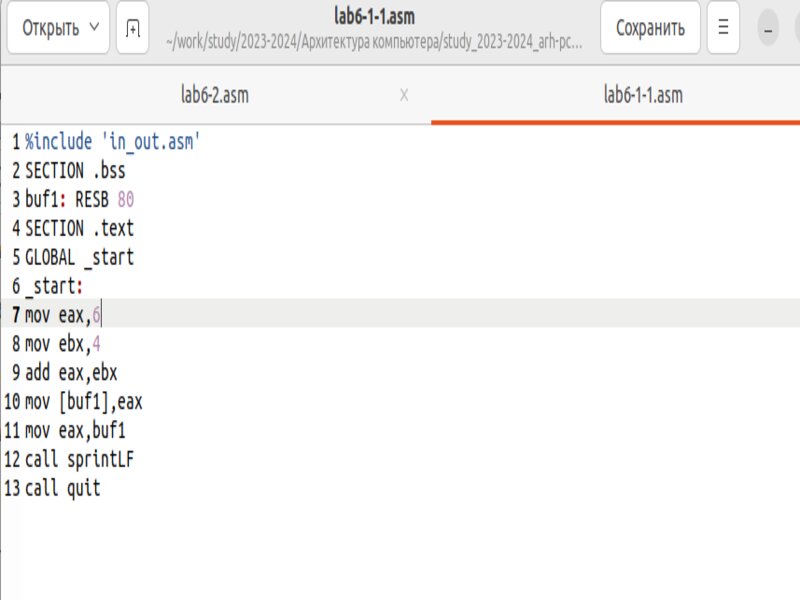


Рис. 5: Изменяю файл lab6-1-1.asm

Запускаю файл. (рис. 6).

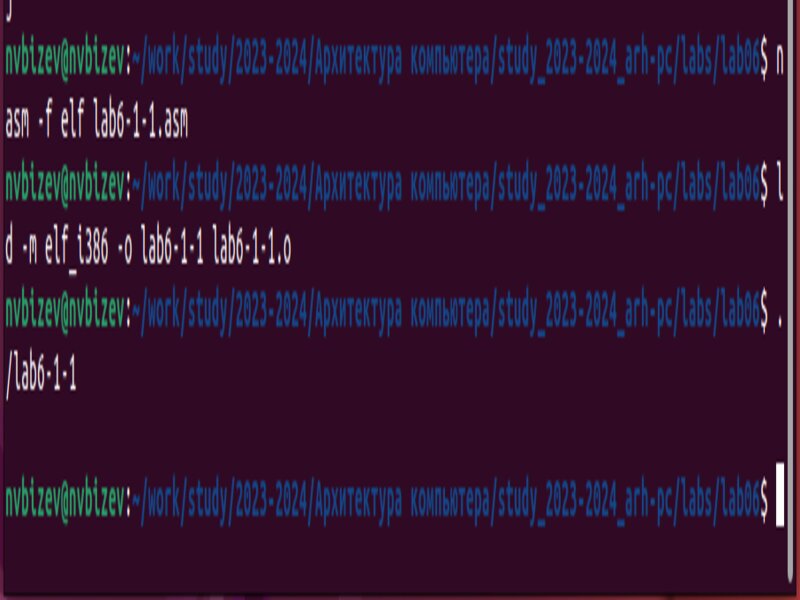


Рис. 6: Запуск файла lab6-1.asm

Теперь вывелся символ с кодом 10, это символ перевода строки, этот символ не отображается при выводе на экран.

Создаю файл lab6-2.asm (рис. 7).

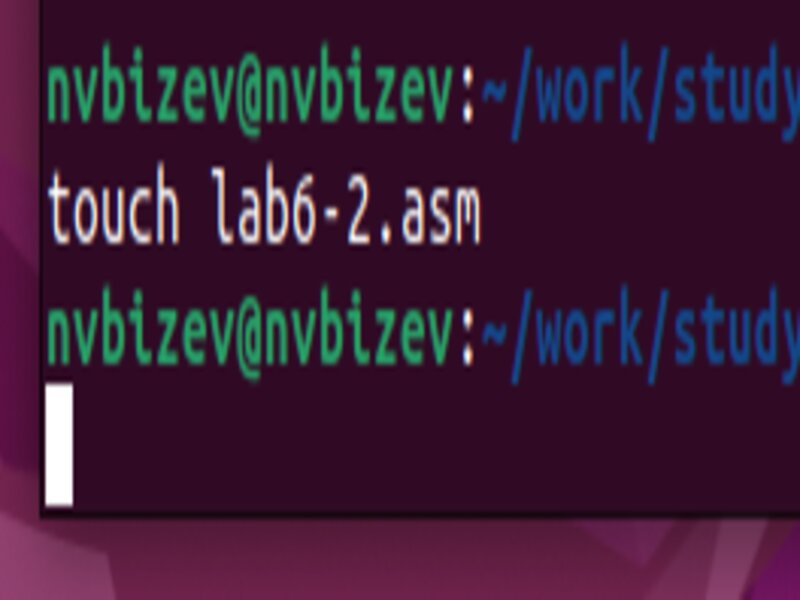


Рис. 7: Создание lab6-2.asm

Ввожу в файл lab6-2.asm текст программы из листинга 6.2.(рис. 8)

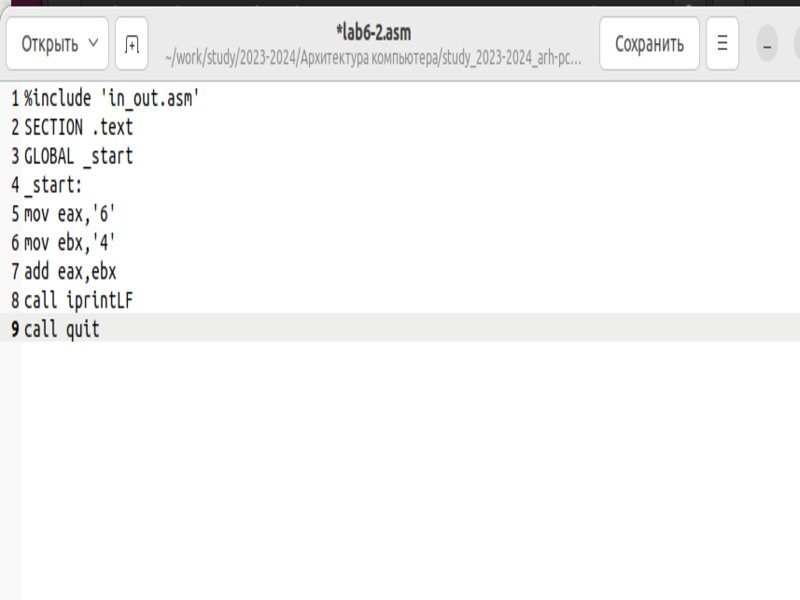


Рис. 8: Ввод текста программы в файл

Создаю исполняемый файл и вывожу результат работы программы. (рис. 9).

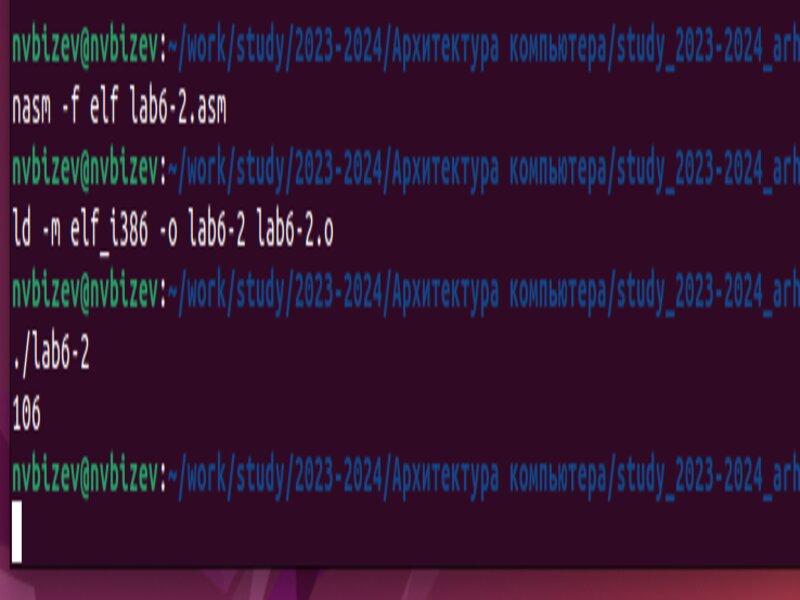


Рис. 9: Вывод работы программы

Аналагично предыдущему примеру изменяю символы на числа. (рис. 10).

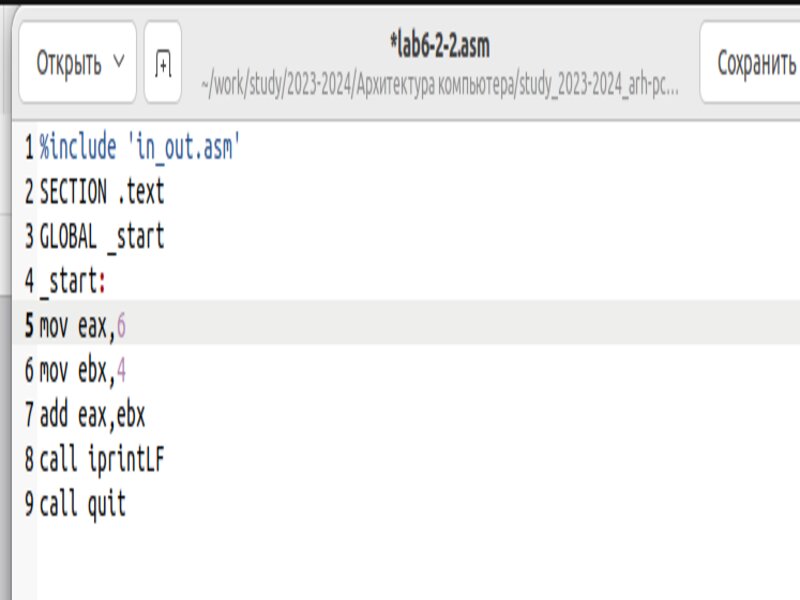


Рис. 10: Замена символов на числа

Вывожу результат с заменой. (рис. 11).

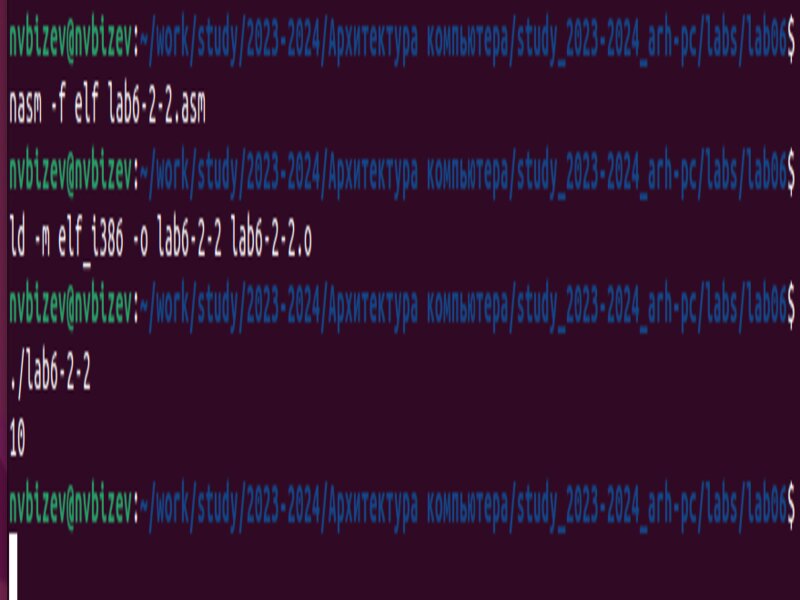


Рис. 11: Вывод работы программы

Результат работы программы - 10.

Создаю файл lab6-2-2 и заменяю функцию inprintLF на iprint. (рис. 12).

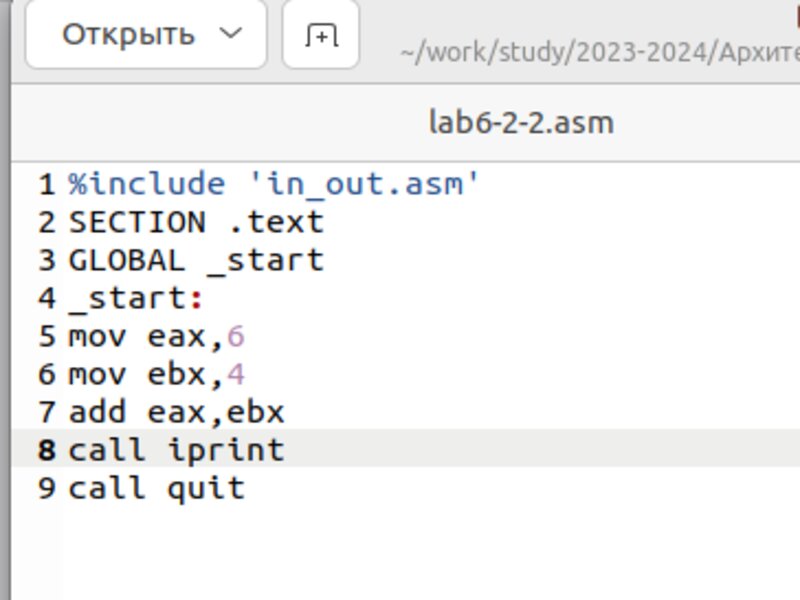


Рис. 12: Замена функции inprintLF на iprint

Запускаю программу. (рис. 13).

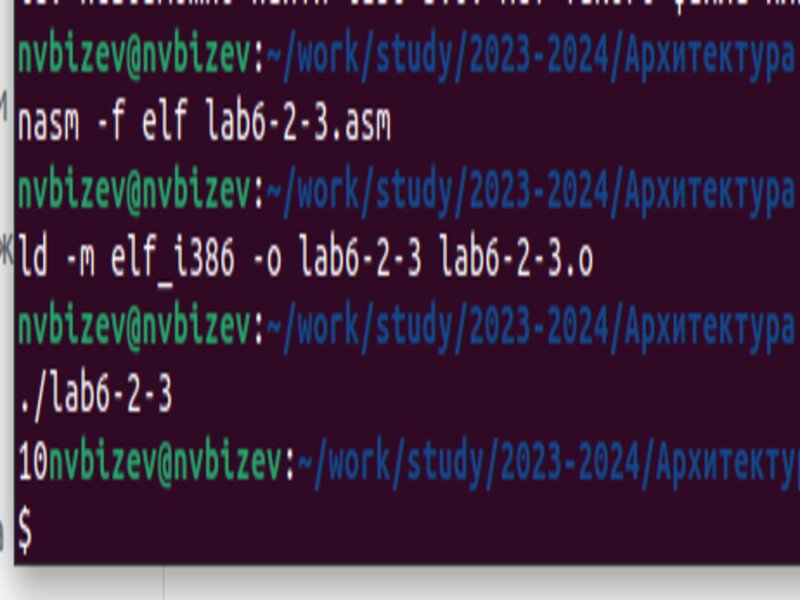


Рис. 13: Запуск программы

При использовании функции iprintLF результат выводится на следующую строку, в то время как при использовании функции iprint результат выводится сразу.

## 4.2 Выполнение арифметических операций в NASM

Создаю файл lab6-3.asm (рис. 14).

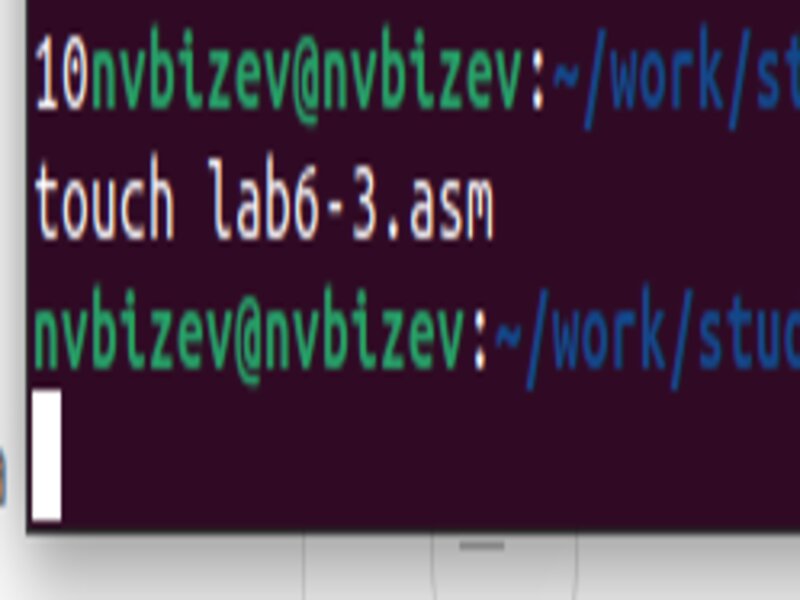


Рис. 14: Создание lab6-3.asm

Открываю файл в текстовом редакторе и ввожу текст программы из листинга 6.3.(рис. 15).

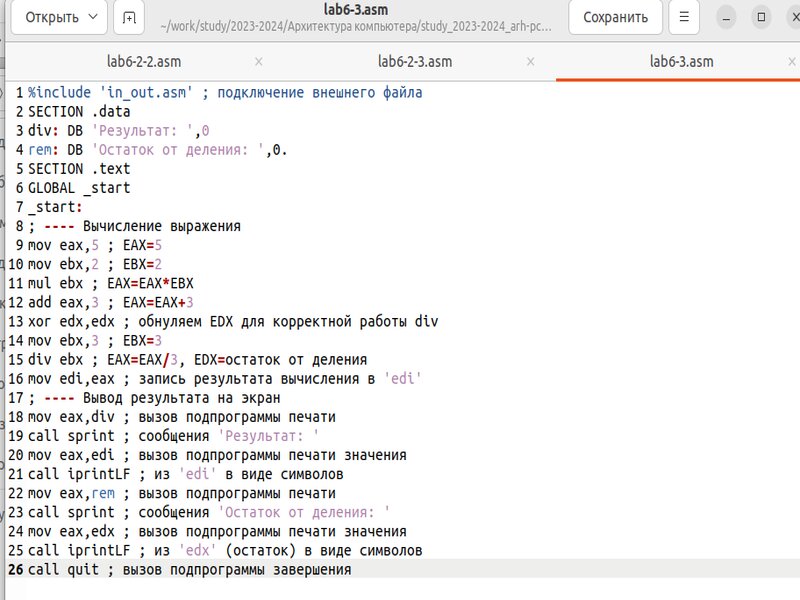


Рис. 15: Редактирование файла

Создаю исполняемый файл lab6-3 и запускаю его. (рис. 16).

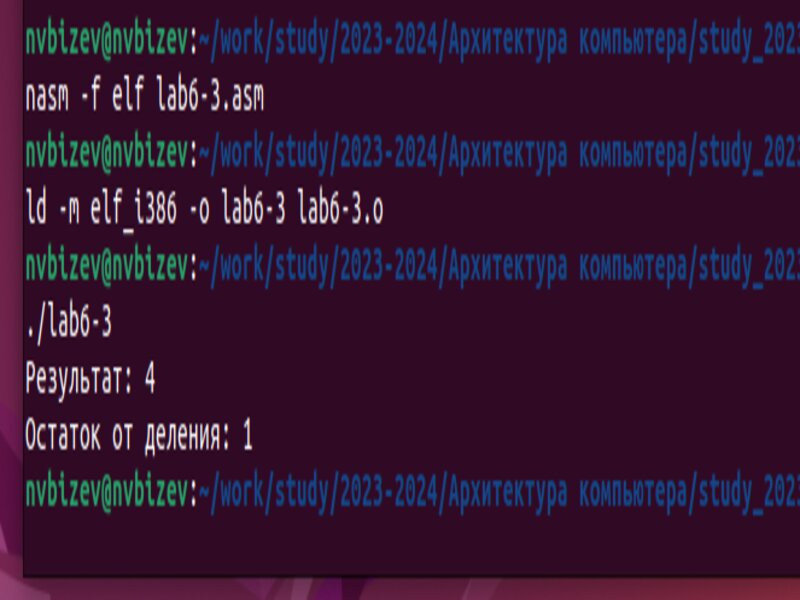


Рис. 16: Создание и запуск lab6-3

Результат программы удовлетворяет f(x) = (5 \* 2 + 3)/3. Теперь я заменю это выражение на f(x) = (4 \* 6 + 2)/5.

Изменяю файл чтобы он удовлетворял выше указанному выражению. (рис. 17).

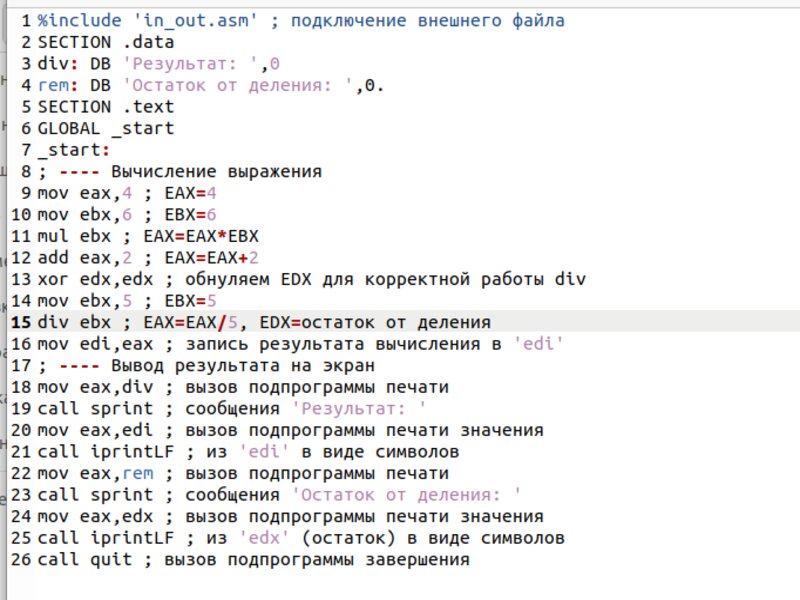


Рис. 17: Редактирование файла

Создаю исполняемый файл lab6-3 и запускаю его. (рис. 18).

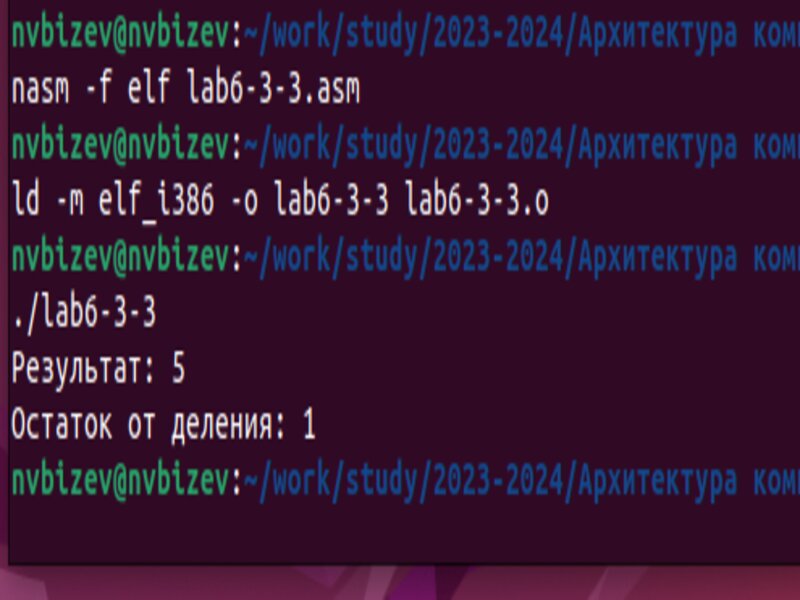


Рис. 18: Создание и запуск lab6-3

Результат работы - 5, отстаток же - 1.

Рассмотрим другой пример.

Создаю файл variant.asm (рис. 19).

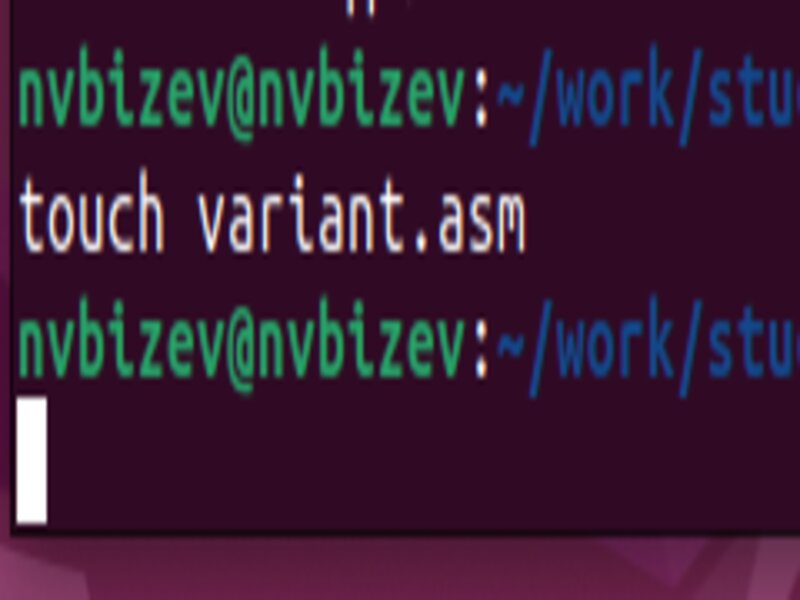


Рис. 19: Создание variant.asm

Ввожу текст программы из листинга 6.4 в файл. (рис. 20).

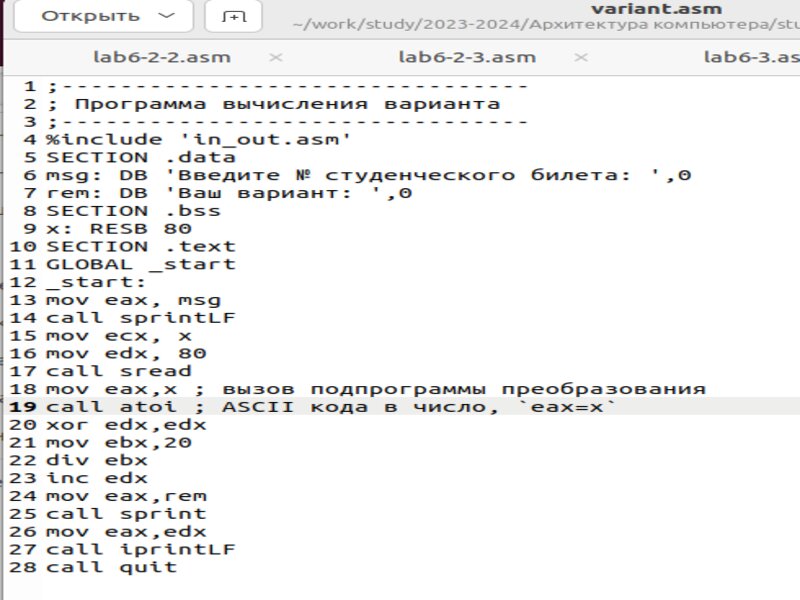


Рис. 20: Ввод текста программы

Программа запрашивает № студенческого билета и основываясь на нем выводит мой вариант. Номер моего студенческого билет - 1132230806.

Создаю исполняемый файл variant и запускаю его. (рис. 21).

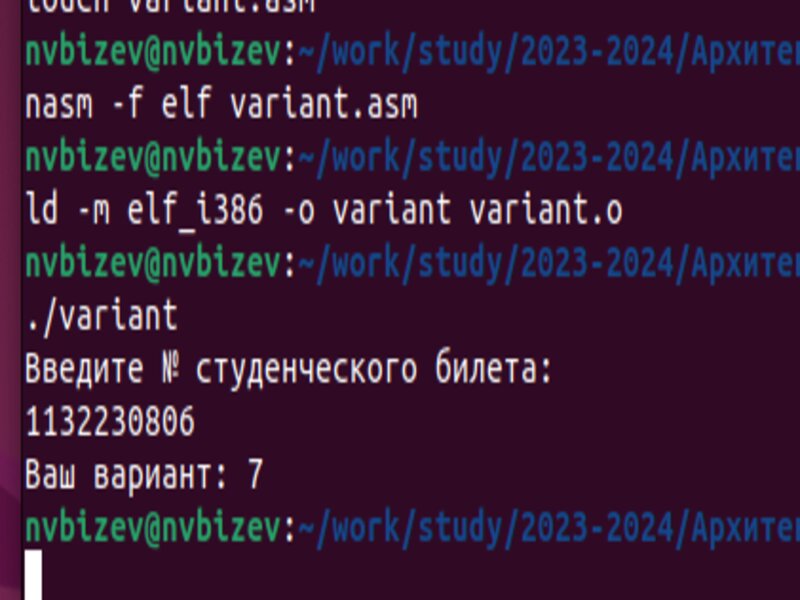


Рис. 21: Создание и запуск variant

### 4.2.1 Ответы на вопросы по программе

1. За вывод сообщения “Ваш вариант” отвечают строки кода:

mov eax,rem  
call sprint

1. Инструкция mov ecx, x используется, 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
2. call atoi используется для вызова подпрограммы из внешнего файла, которая преобразует ascii-код символа в целое число и записывает результат в регистр eax
3. 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

1. При выполнении инструкции div ebx остаток от деления записывается в регистр edx
2. Инструкция inc edx увеличивает значение регистра edx на 1
3. За вывод на экран результатов вычислений отвечают строки:

mov eax,edx  
call iprintLF

# 5 Выводы

При выполнении данной лабораторной работы я освоила арифметические инструкции языка ассемблера NASM.

# Список литературы

1. https://esystem.rudn.ru/pluginfile.php/2089086/mod\_resource/content/0/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%966.%20%D0%90%D1%80%D0%B8%D1%84%D0%BC%D0%B5%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5%20%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%B8%20%D0%B2%20NASM..pdf