Отчёт по лабораторной работе №6

Дисциплина: архитектура компьютера

Хатамов Эзиз

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. Далее рассмотрены все существующие способы задания адреса хранения операндов – способы адресации. Существует три основных способа адресации: • **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. Например, определим переменную intg DD 3 – это означает, что задается область памяти размером 4 байта, адрес которой обозначен меткой intg. В таком случае, команда mov eax,[intg] копирует из памяти по адресу intg данные в регистр eax. В свою очередь команда mov [intg],eax запишет в память по адресу intg данные из регистра eax. Также рассмотрим команду mov eax,intg В этом случае в регистр eax запишется адрес intg. Допустим, для intg выделена память начиная с ячейки с адресом 0x600144, тогда команда mov eax,intg аналогична команде mov eax,0x600144 – т.е. эта команда запишет в регистр eax число 0x600144.

# 4 Выполнение лабораторной работы

## 4.1 Символьные и численные данные в NASM

Для начала я создал каталог для программам лабораторной работы №6, потом перешел на этот каталог и создал файл lab6-1.asm (рис. 1).

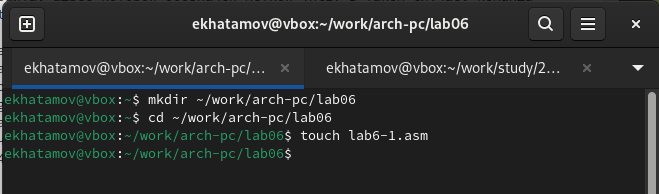


Рис. 1: Создания каталога и файла

Потом я зашел на Midnight Commander и скопировал in\_out.asm в каталог с файлом lab6-1.asm с помощью клавищи F5

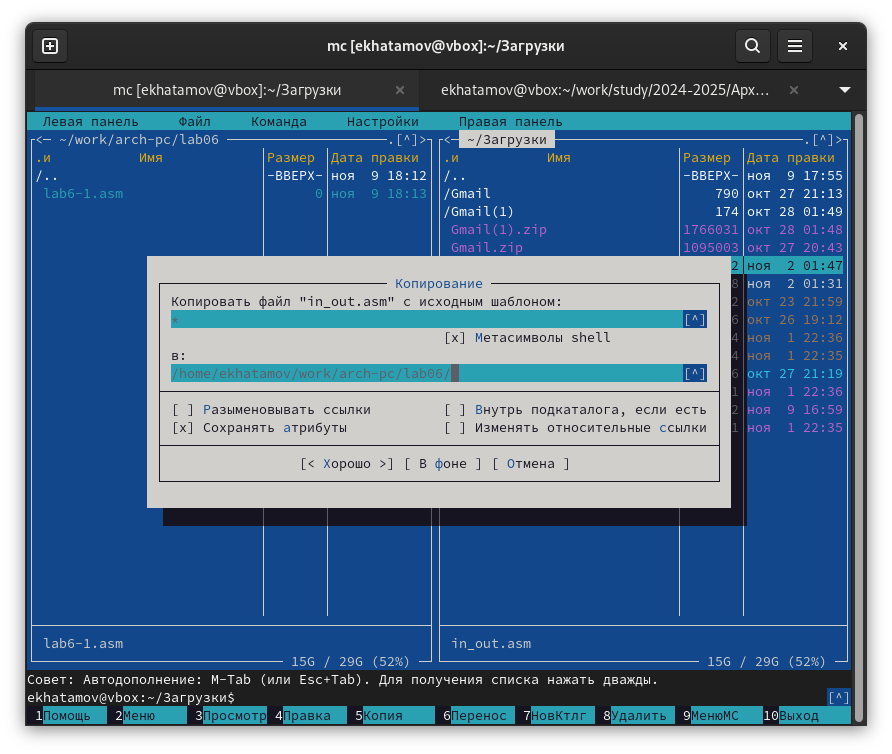


Рис. 2: Копирования файла на нужный каталог

Потом я открыл созданный файл lab6-1.asm и внес изменения в тексте файла.

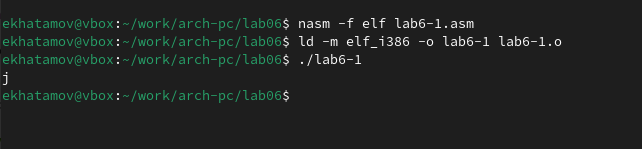


Рис. 3: Изменения в тексте файла

потом в тексте я изменил eax,‘6’ на eax,6 а ebx,‘4’ на ebx,4 .

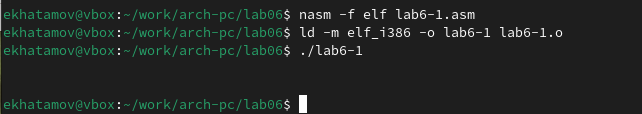


Рис. 4: Изменения ‘6’ и ‘4’ на 6 и 4

Все равно не вышло число 10 а вместо него вышло пустота. я зашел и посмотрел таблицу АSCII и там я увидел что символ числа 10 это пустота

Потом я создал новый файл в том же каталоге в катором был прошлый файл

Рис. 5: Создания нового файла lab6-2.asm

Рис. 5: Создания нового файла lab6-2.asm

После того как я создал файл я зашел на него и изменил текст файла

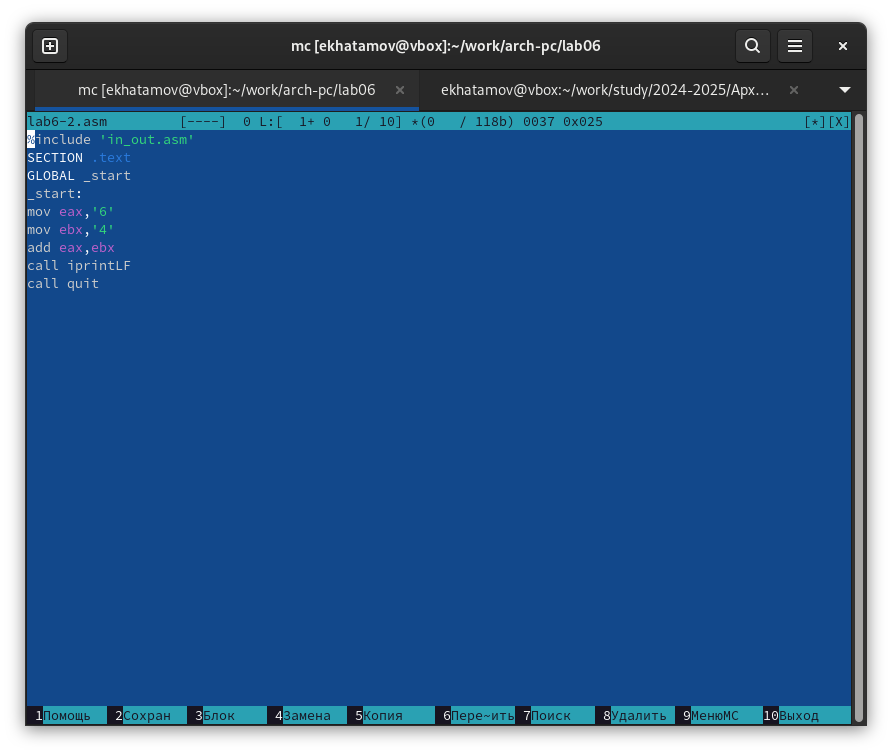


Рис. 6: Изменения текста файла lab6-2.asm

Создал исполняемый файл и запустил его

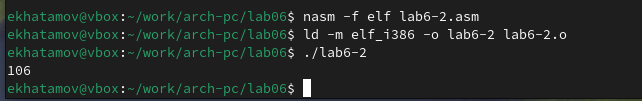


Рис. 7: Создание и запуск исполняемого файла

Потом заменил где eax ‘6’ и eax ‘4’ на ebx,6 и ebx,4

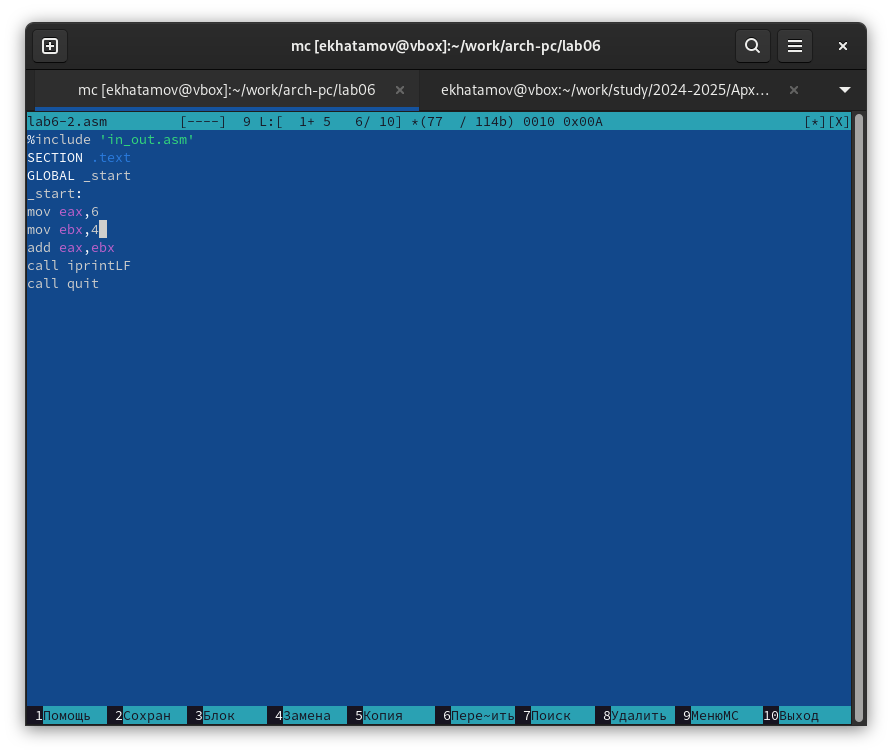


Рис. 8: Изменения eax и ebx

После изменения я создал исполняемый файл и запустил его

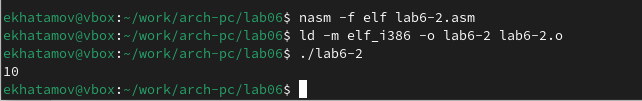


Рис. 9: Создание и запуск исполняемого файла(измененного eax)

После этих изменений я получил цифру 10. Потом я изменил iprintLF на iprint

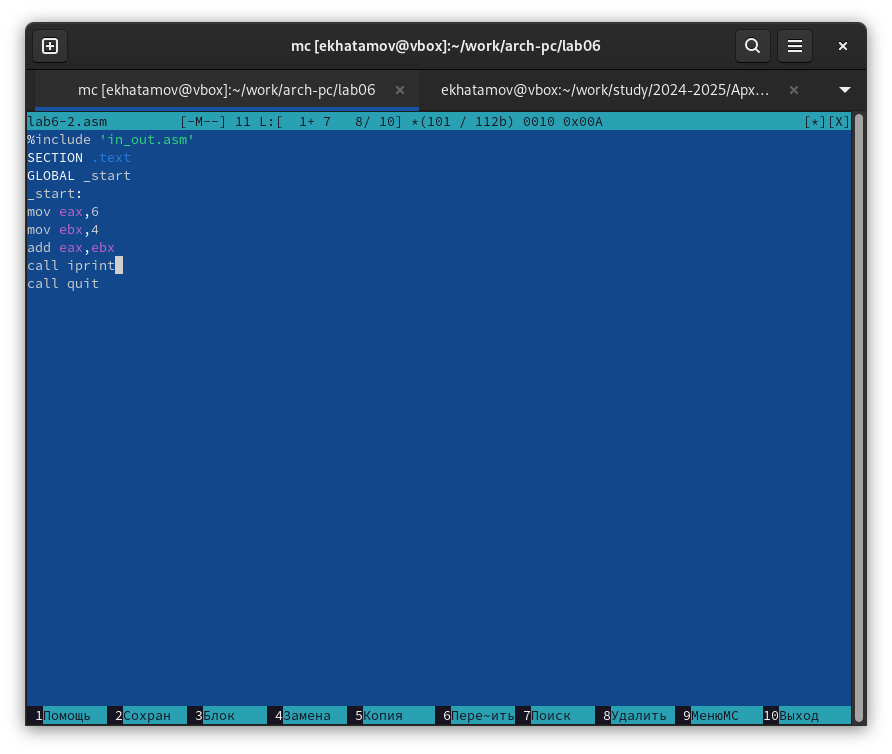


Рис. 10: Изменения iprintLF на iprint

После изменения я занова создал исполняемый файл и запустил его

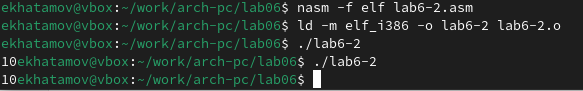


Рис. 11: Создание и запуск исполняемого файла(измененного iprintLF)

Таким образом, разница между iprintLF и iprint в NASM заключается в том, что iprintLF — это функция для печати целых чисел с переводом на новую строку, а iprint — для простой печати целых чисел без перевода на новую строку

## 4.2 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.

Для начала я создал файл lab6-3.asm с помощью touch потом внес изменения в текст файла

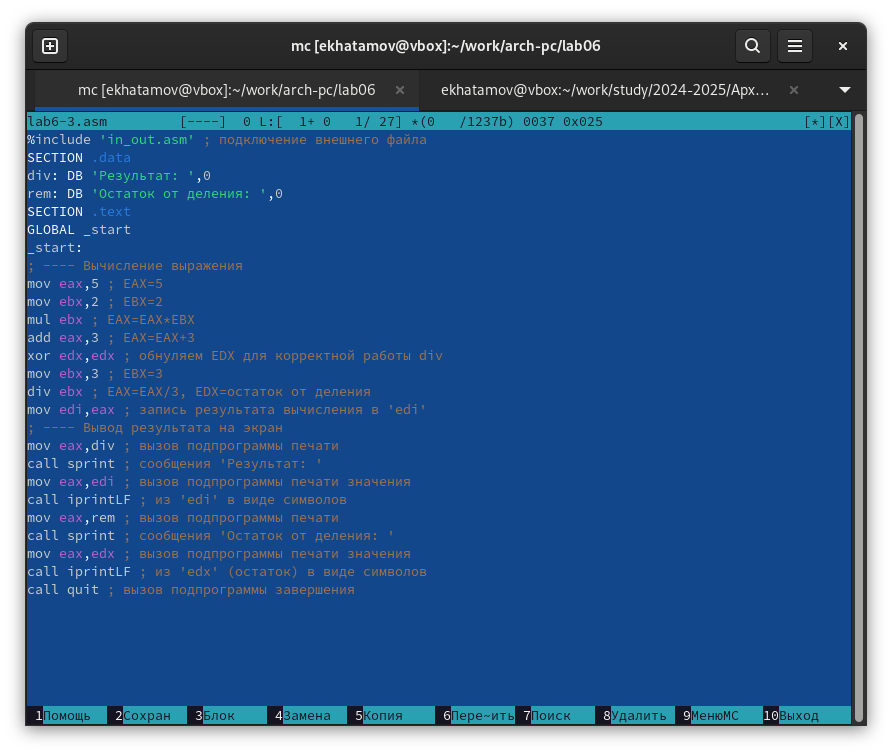


Рис. 12: Cоздание файла и изменения текста lab6-3.asm

Потом создал исполняемый файл lab6-3.asm и запуслтил его

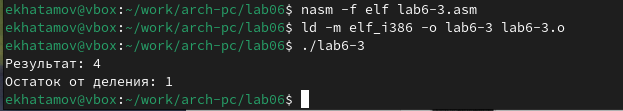


Рис. 13: Создание исполняемого файла lab6-3.asm и запуск файла

Потом изменил текст программы для вычисления выражения 𝑓(𝑥) = (4 ∗ 6 + 2)/5

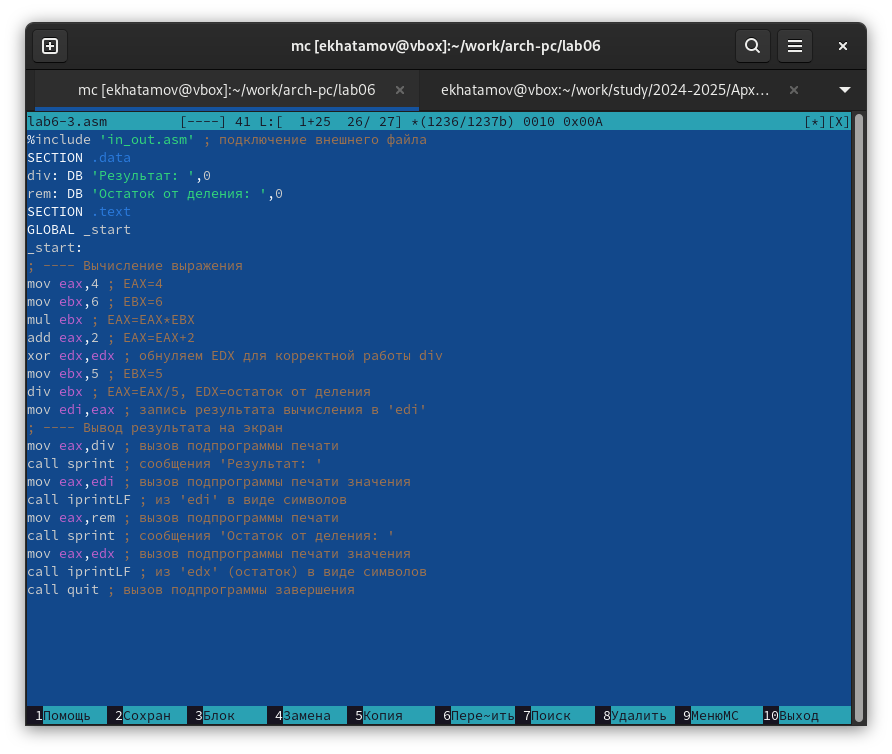


Рис. 14: Изменения текста lab6-3.asm

Потом создал исполняемый файл lab6-3.asm и запуслтил его

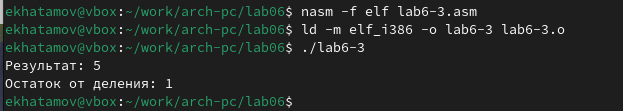


Рис. 15: Создание исполняемого файла lab6-3.asm и запуск файла (с изменением)

### 4.2.2 В качестве другого примера рассмотрим программу вычисления варианта задания по номеру студенческого билета, работающую по следующему алгоритму:

Для начала я создал файл variant.asm с помощью команды touch и внес в него изменения

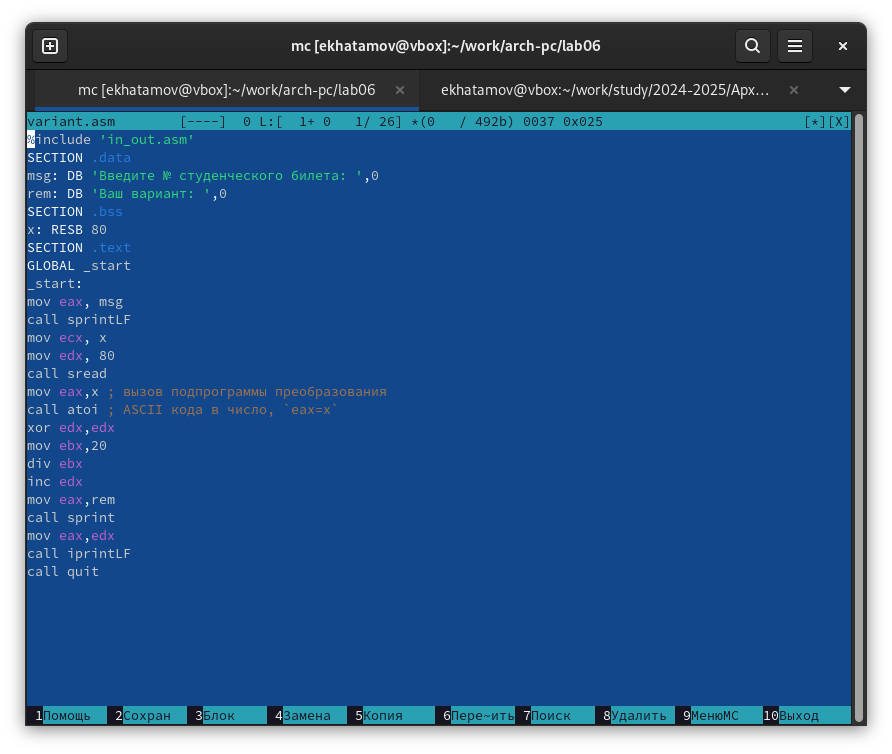


Рис. 16: Cоздание файла и изменения текста variant.asm

Потом создал исполняемый файл запустил его

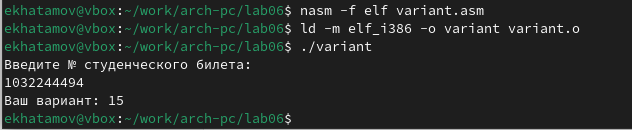


Рис. 17: Создание исполняемого файла lab6-3.asm и запуск файла

## 4.3 Ответы на вопросы по программе

1. Какие строки листинга 6.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’? Oтвет:За вывод сообщения “Ваш вариант” отвечают строки кода: mov eax,rem call sprint
2. Для чего используется следующие инструкции? mov ecx, x mov edx, 80 call sread Oтвет: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
3. Для чего используется инструкция “call atoi”? Oтвет:Инструкция «call atoi» используется для преобразования строки в целое число.
4. Какие строки листинга 6.4 отвечают за вычисления варианта? Oтвет: 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. В какой регистр записывается остаток от деления при выполнении инструкции “div ebx”? Oтвет:При выполнении инструкции div ebx остаток от деления записывается в регистр edx
6. Для чего используется инструкция “inc edx”? Oтвет:Инструкция inc edx увеличивает значение регистра edx на 1
7. Какие строки листинга 6.4 отвечают за вывод на экран результата вычислений? Oтвет:За вывод на экран результатов вычислений отвечают строки: mov eax,edx call iprintLF

# 5 Выполнение заданий для самостоятельной работы

Для начала я создал файл lab6-4.asm c помощью touch. И открыл файл для редактирования, ввел туда текст программы для вычисления (11+x)\*2-6

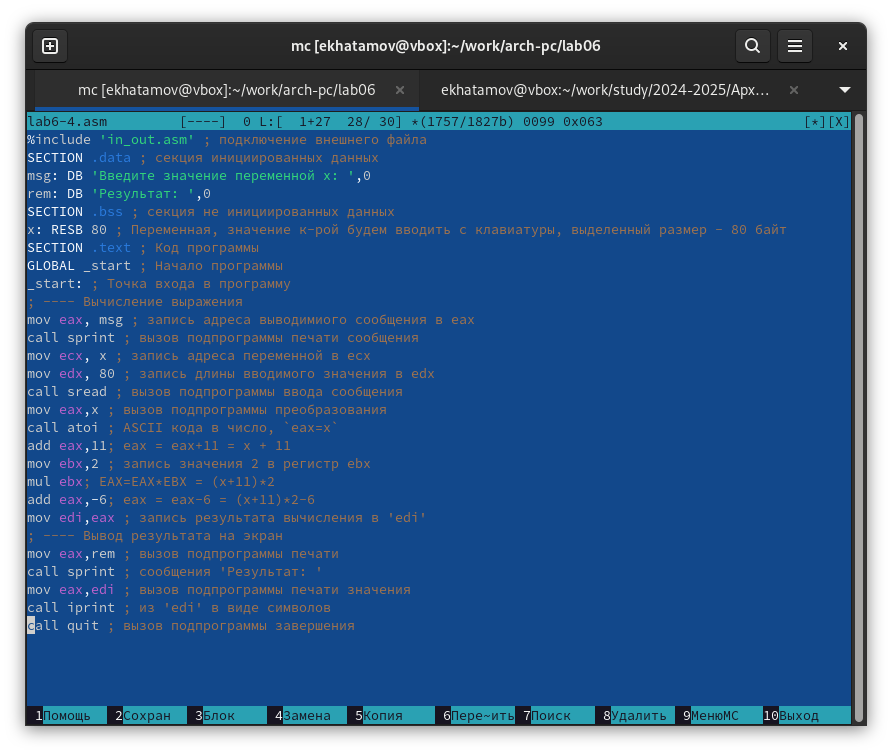


Рис. 18: Cоздание файла и изменения текста lab6-4.asm

Потом создал исполняемый файл и запустил его. И ввел туда цифру 1

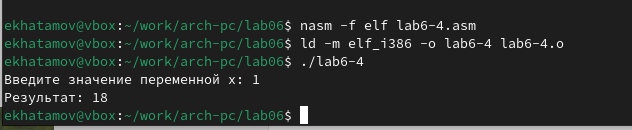


Рис. 19: Создание исполняемого файла lab6-4.asm и запуск файла

Еще раз запустил файл но в этот раз ввел цифру 9 и по алгоритму отработала верно и дал верный ответ

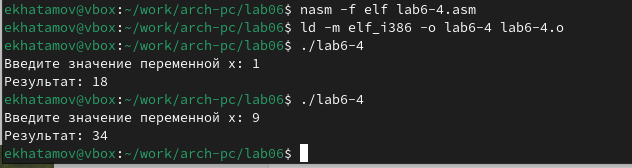


Рис. 20: Запуск исполняемго файла lab6-4.asm

# 6 Выводы

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

# Список литературы

1. Лабораторная работа №7
2. Таблица ASCII