Отчёта по лабораторной работе №6

Арифметические операции в NASM.

Семенов Сергей Алексеевич

Содержание

# 1 Цель работы

Освоить арифметических инструкций языка ассемблера NASM и написать программы для вычисления арифметических выражений с неизвестной.

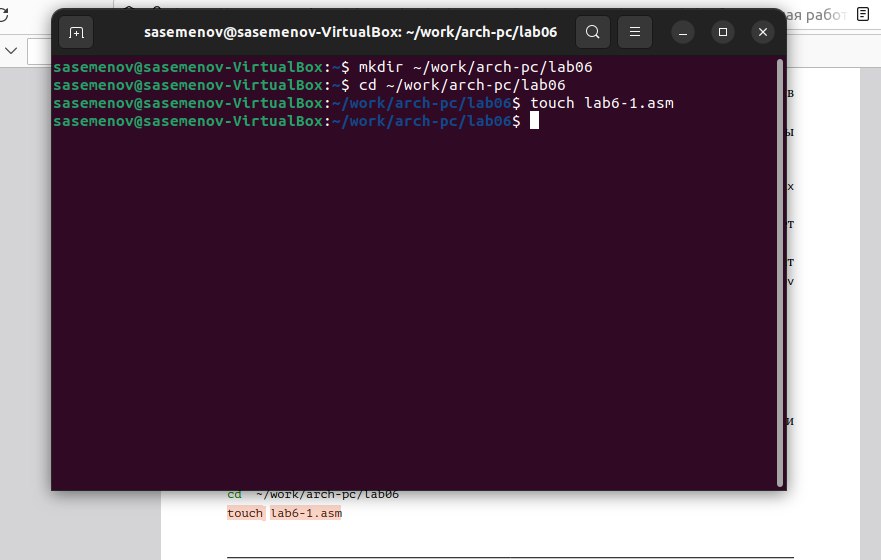
# 2 Задание

Написать программы для решения выражений.

# 3 Выполнение лабораторной работы

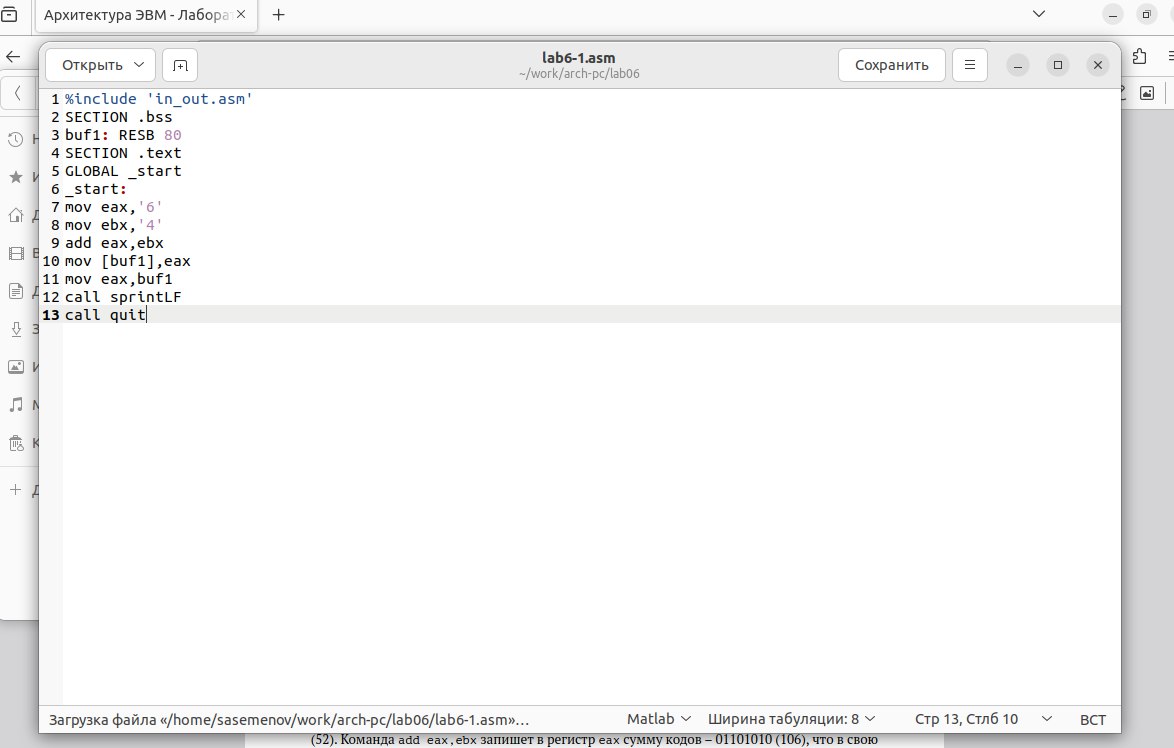
## 3.1 Cимвольные и численные данные в NASM

Создаем каталог для программ ЛБ6, и в нем создаем файл



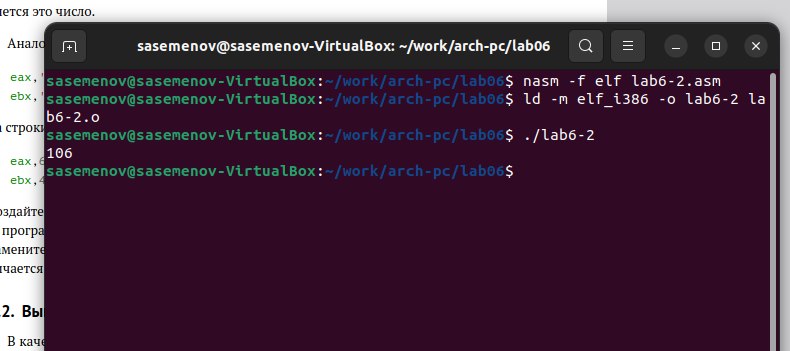
Создаем каталог с помощью команды mkdir и файл с помощью команды touch

Открываем файл в Midnight Commander и заполняем его в соответствии с листингом 6.1



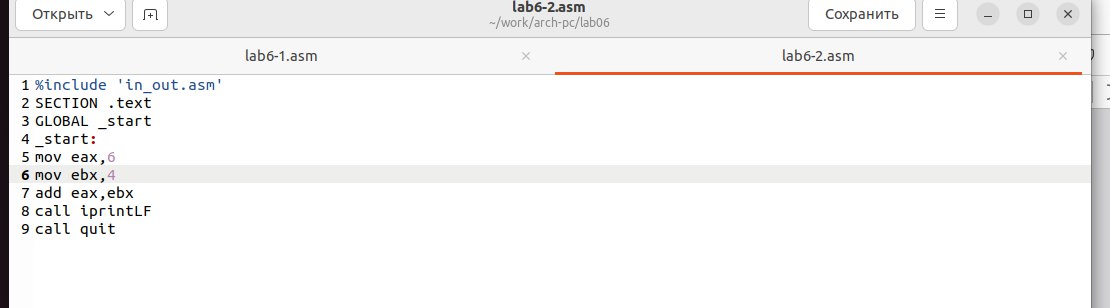
Заполняем файл

Создаем исполняемый файл и запускаем его.



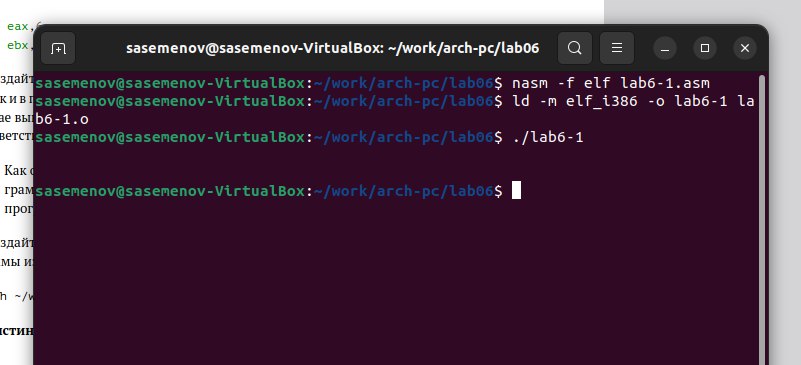
Запускаем файл и смотрим на его работу

Снова открываем файл для редактирования и убиравем кавычки с числовых значений



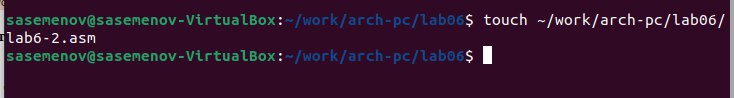
Изменяем файл

Создаем исполняемый файл и запускаем его



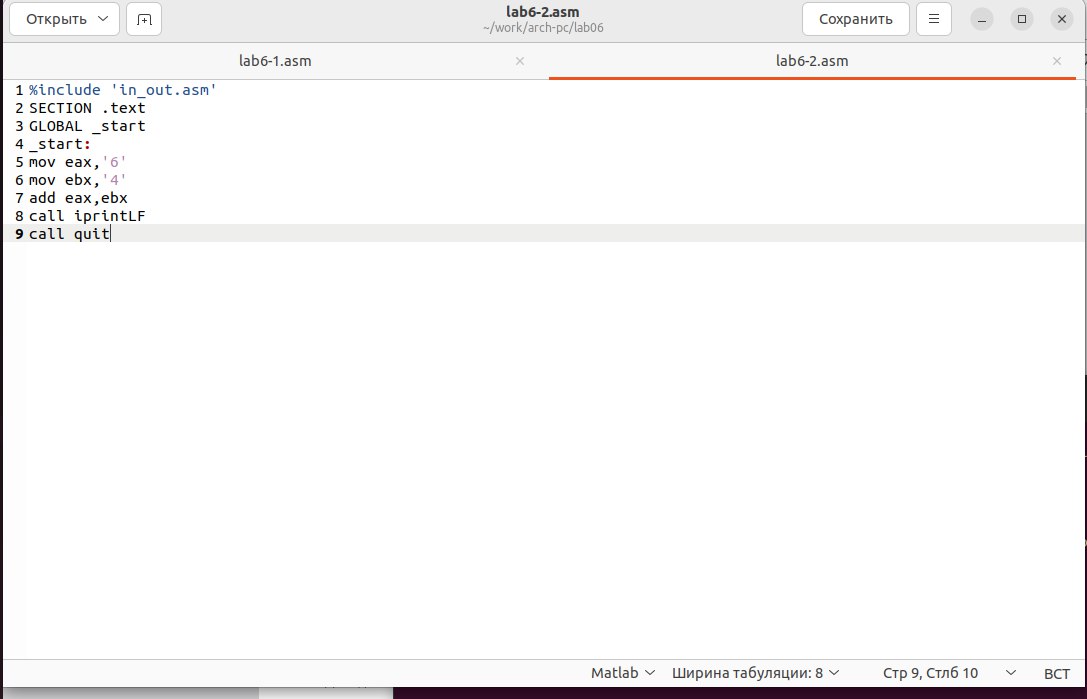
Запускаем файл и смотрим на его работу

Создаем новый файл в каталоге



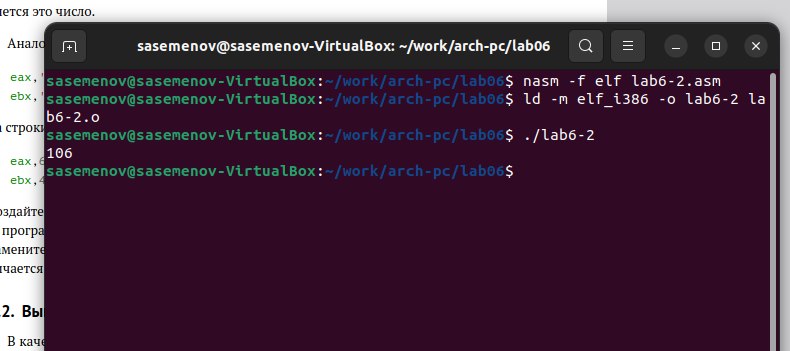
Создаем файл

Заполняем файл в соответствии с листингом 6.2



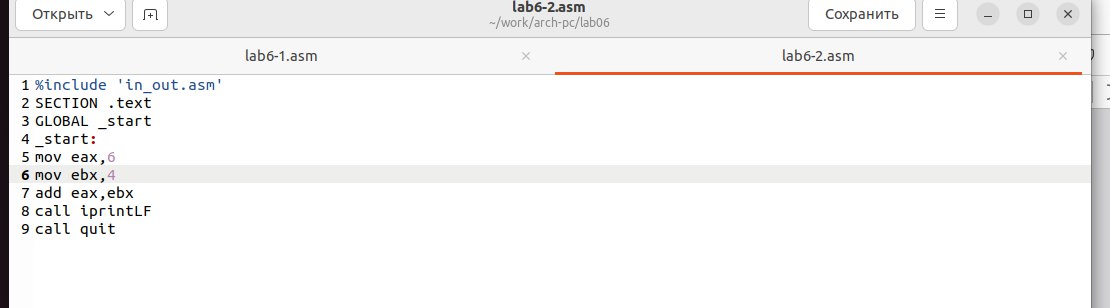
Заполняем файл

Создаем исполняемый файл и запускаем его



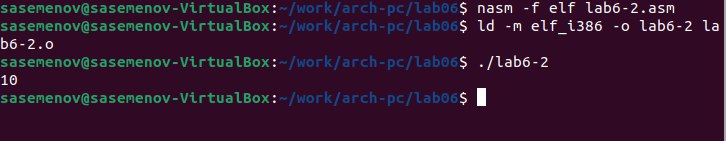
Смотрим на работу программы

Снова открываем файл для редактирования и убиравем кавычки с числовых значений



Изменяем файл

Создаем исполняемый файл и запускаем его

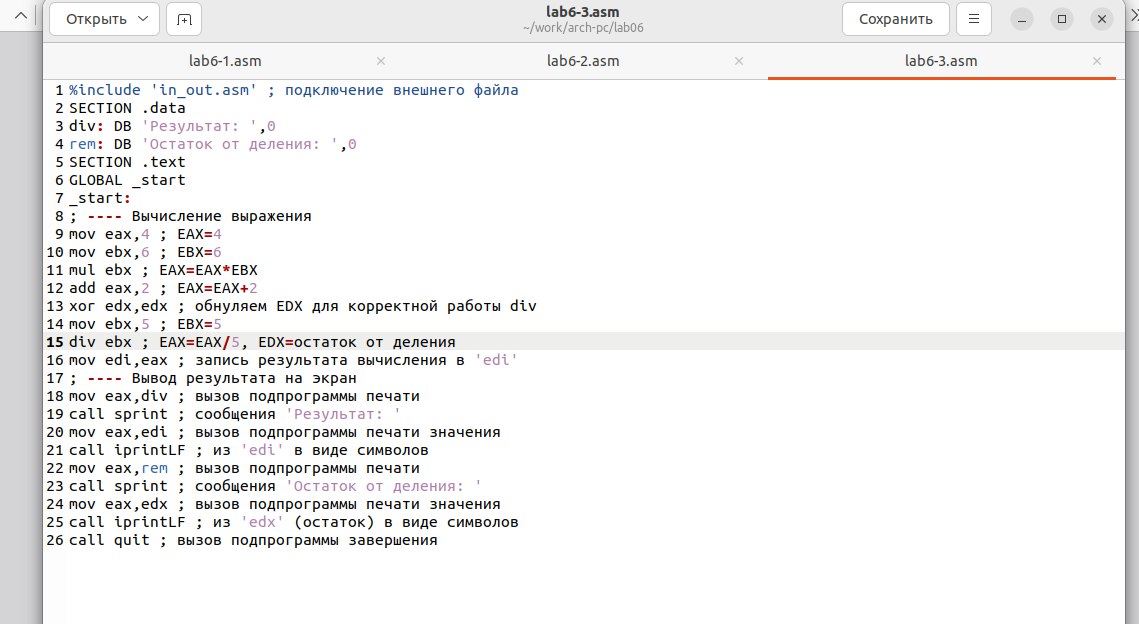


Смотрим на работу программы

Вывод функций iprintLF и iprint отличаются только тем, что LF переносит на новую строку.

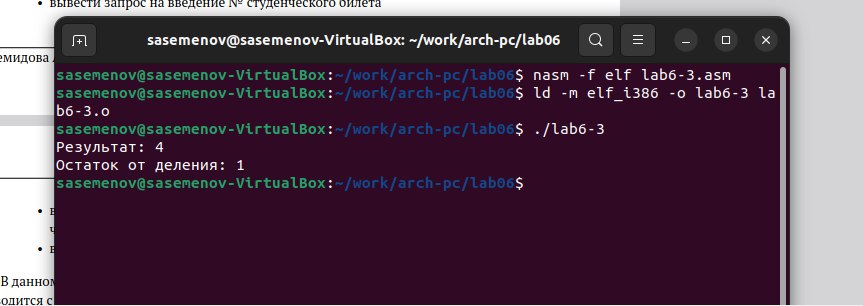
## 3.2 Выполнение арифметических операций в NASM

Открываем файл и редактируем в соответствии с листингом 6.3



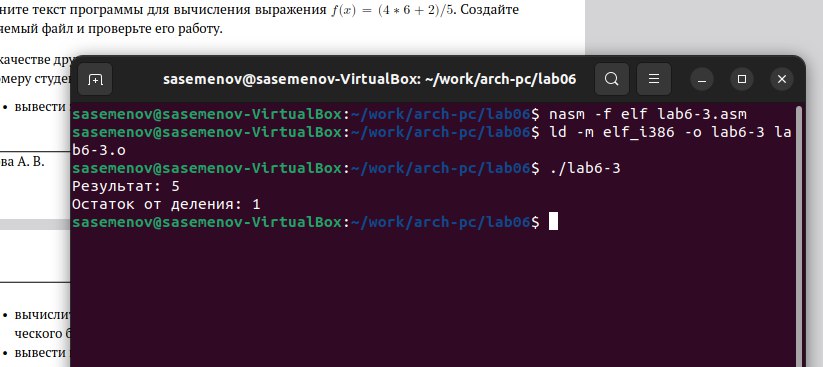
Заполняем файл

Создаем исполняемый файл и запускаем



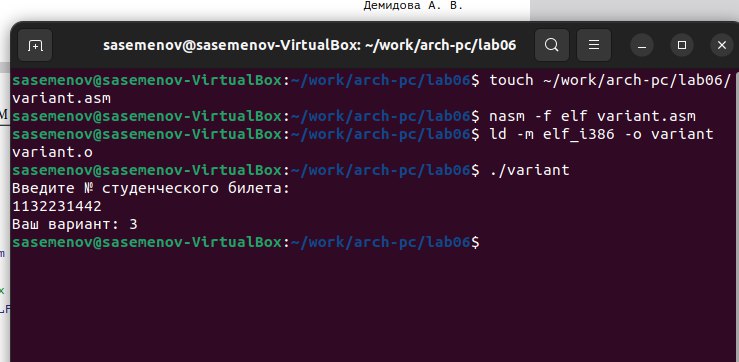
Смотрим на результат работы программы

Компилируем файл и запускаем программу



Смотрим на результат работы программы

Компилируем файл и запускаем его



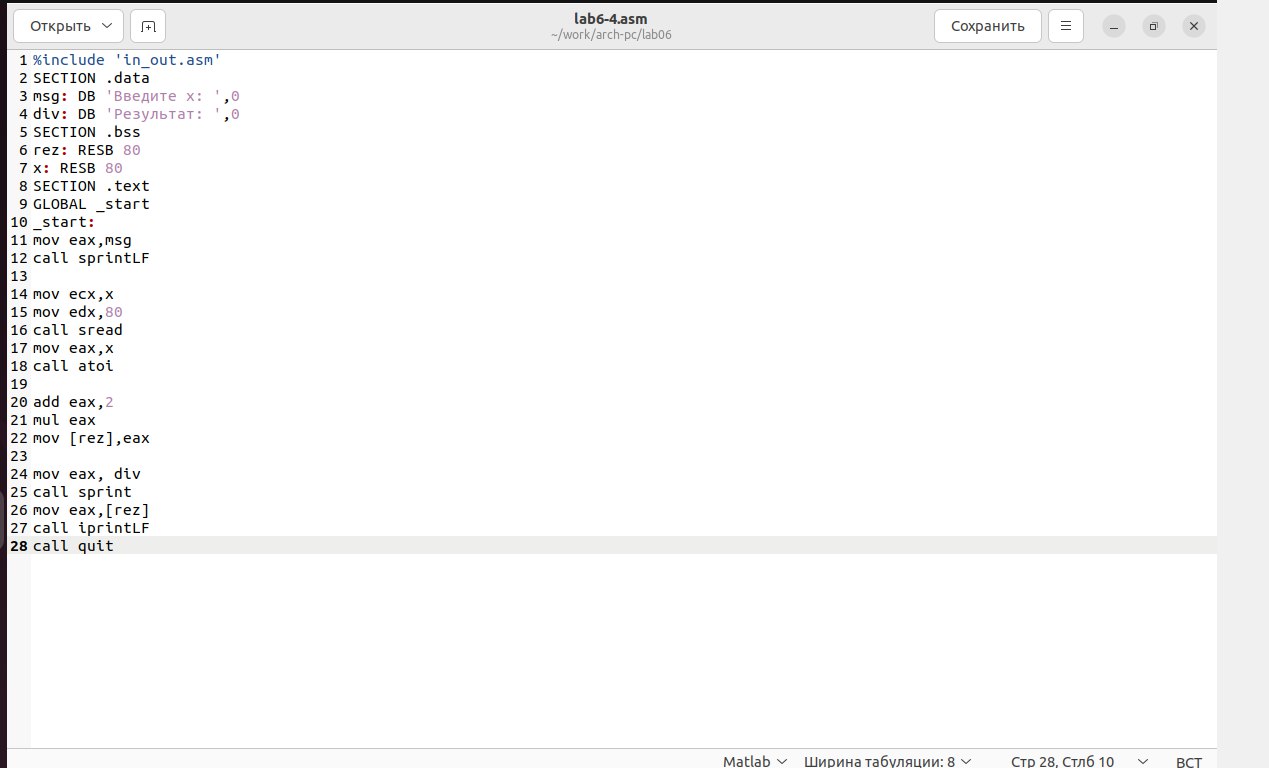
Проверяемс результат работы программы

## 3.3 Ответы на вопросы по программе

1. Строка “mov eax,rem” и строка “call sprint” отвечают за вывод на экран сообщения ‘Ваш вариант:’.
2. Эти инструкции используются для чтения строки с вводом данных от пользователя. Начальный адрес строки сохраняется в регистре ecx, а количество символов в строке (максимальное количество символов, которое может быть считано) сохраняется в регистре edx. Затем вызывается процедура sread, которая выполняет чтение строки.
3. Инструкция “call atoi” используется для преобразования строки в целое число. Она принимает адрес строки в регистре eax и возвращает полученное число в регистре eax.
4. Строка “xor edx,edx” обнуляет регистр edx перед выполнением деления. Строка “mov ebx,20” загружает значение 20 в регистр ebx. Строка “div ebx” выполняет деление регистра eax на значение регистра ebx с сохранением частного в регистре eax и остатка в регистре edx.
5. Остаток от деления записывается в регистр edx.
6. Инструкция “inc edx” используется для увеличения значения в регистре edx на 1. В данном случае, она увеличивает остаток от деления на 1.
7. Строка “mov eax,edx” передает значение остатка от деления в регистр eax. Строка “call iprintLF” вызывает процедуру iprintLF для вывода значения на экран вместе с переводом строки.

## 3.4 Задание для самостоятельной работы

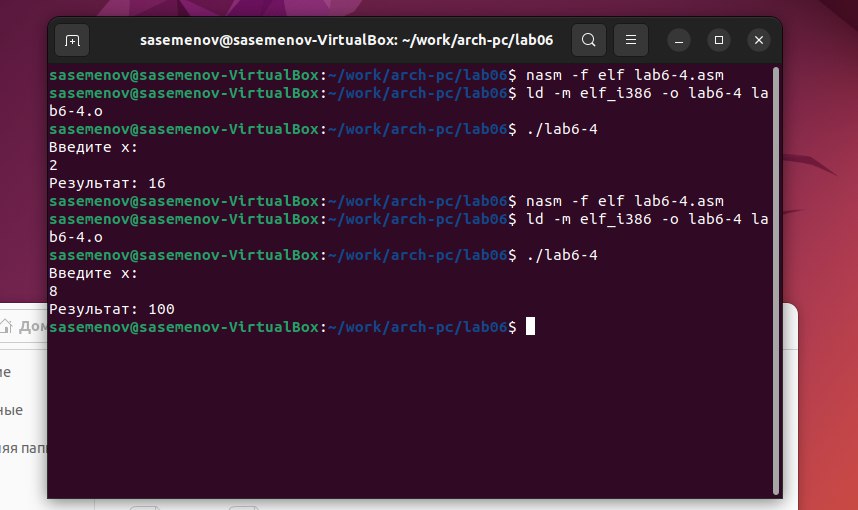
Создаем новый файл в каталоге



Создаем файл

Открываем его и заполняем, чтобы решалось выражение (2+x)^2

Компилируем программу и проверяем для x=2 и для x=8



Проверяем работу программы

# 4 Выводы

Мы приобрели навыки создания исполнительных файлов для решения выражений и освоили арифметические инструкции в NASM.