Отчёт по лабораторной работе №6

Ярослав Антонович Меркулов

Содержание

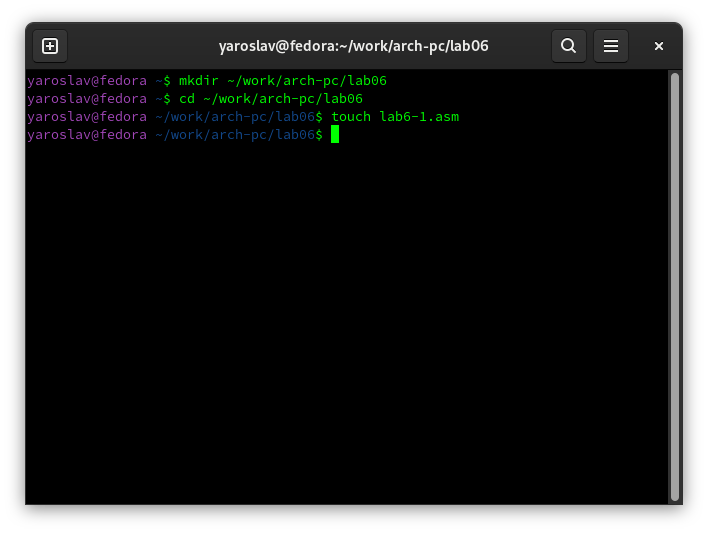
# 1 Цель работы

Освоить арифметические инструкции языка программирования NASM

# 2 Выполнение лабораторной работы

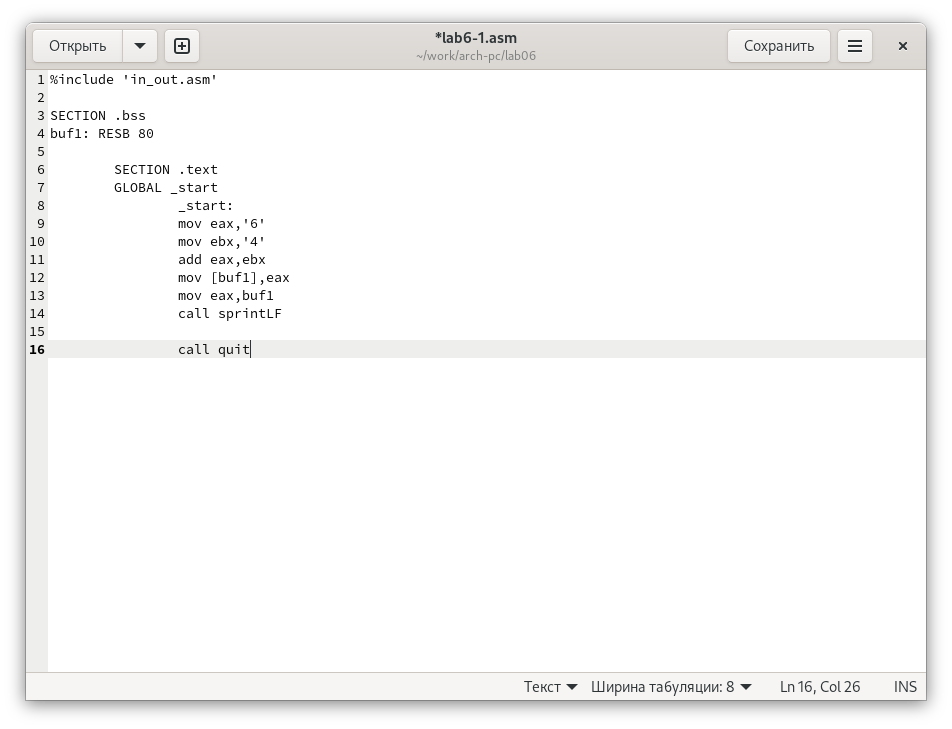
## 2.1 Символьные и численные данные в NASM

1. Создаём каталог для лабораторной работы, переходим в него и создаём файл lab6-1.asm.



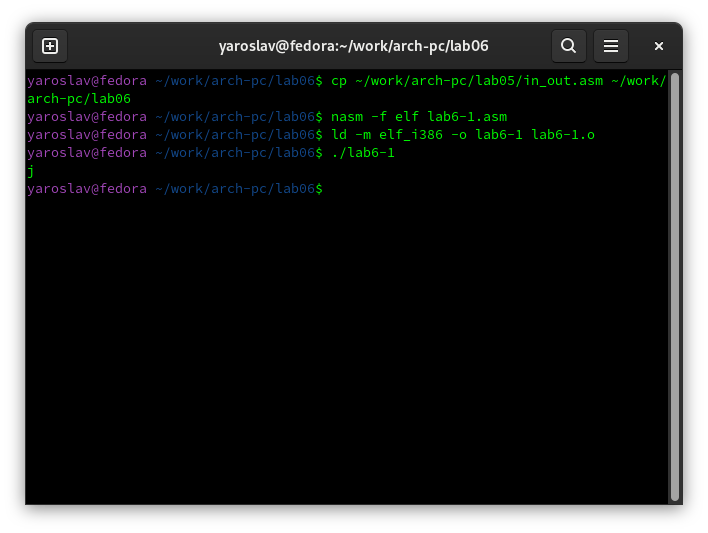
Создание каталога, переход в него, создание файла

1. Вводим текст программы из листинга 6.1.



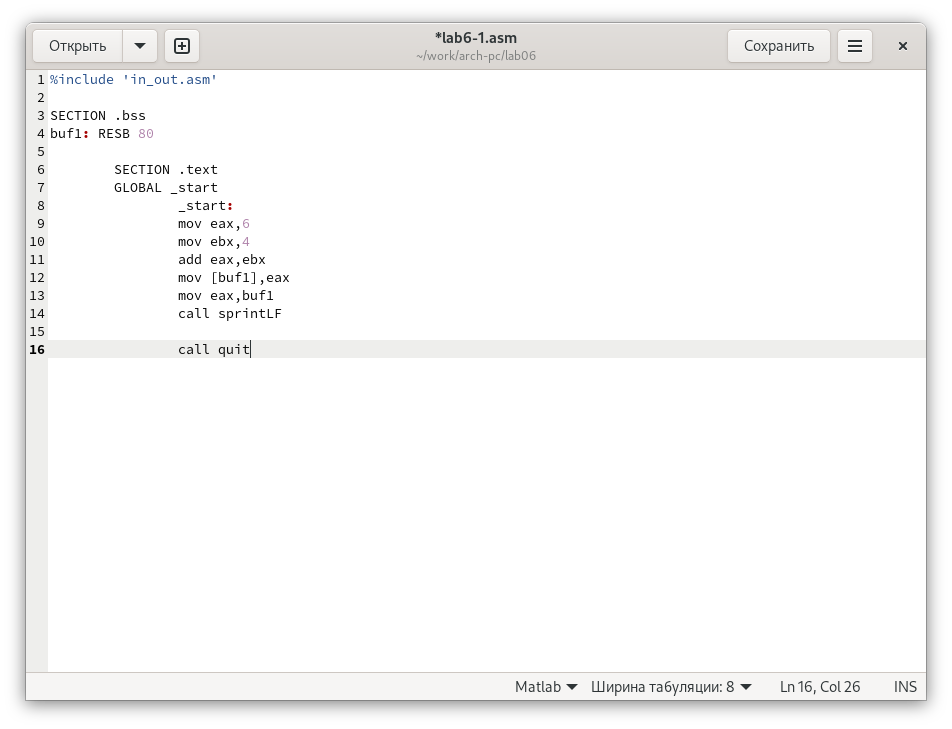
Введённый текст программы

1. Создаём исполняемый файл и запускаем. В результате выводится символ j.

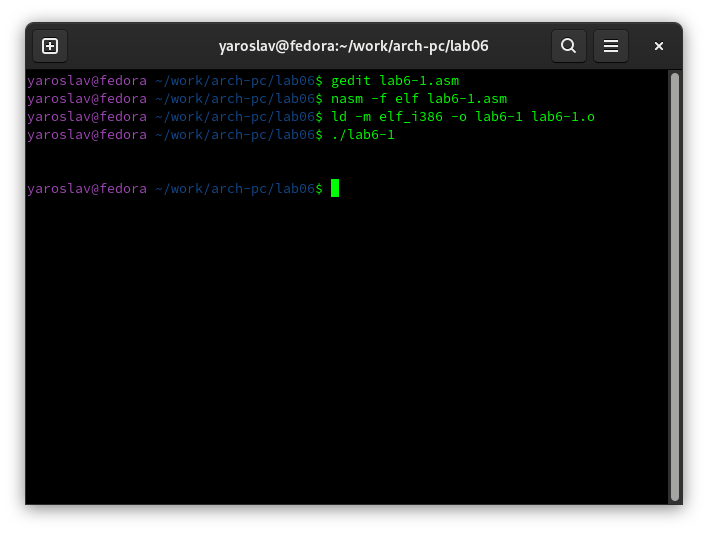


Работа программы

1. Изменяем текст программы (убираем одинарные кавычки), создаём исполняемый файл и запускаем. Вывелся символ переноса строки.

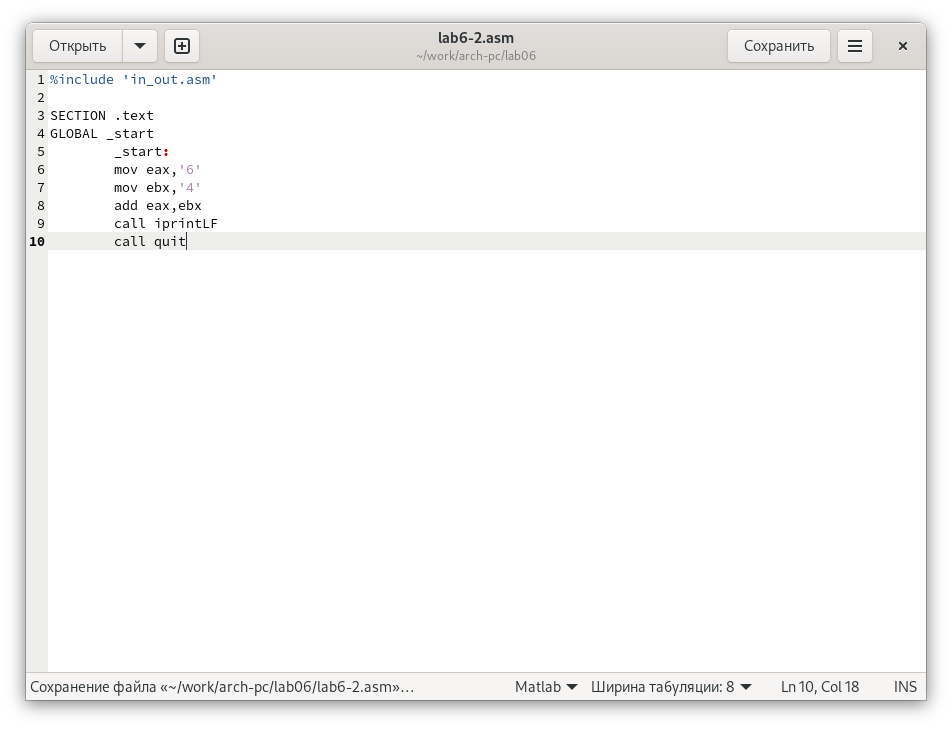


Изменённый файл



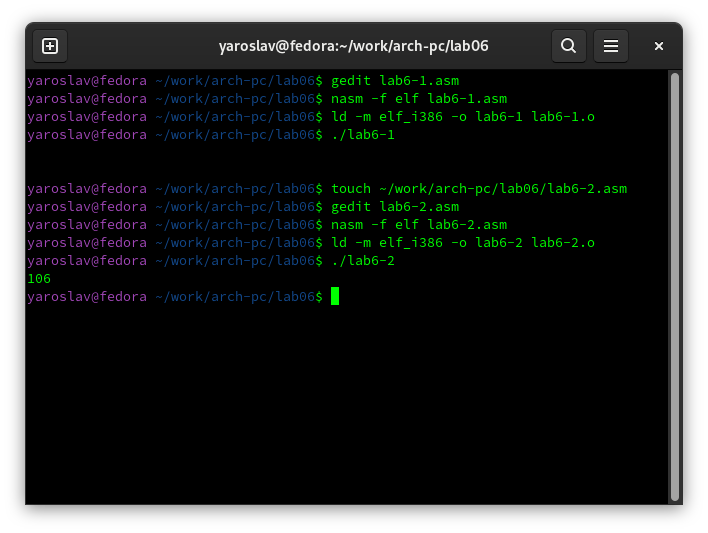
Работа новой программы

1. Создаём новый файл и переносим туда текст из листинга 6.2.



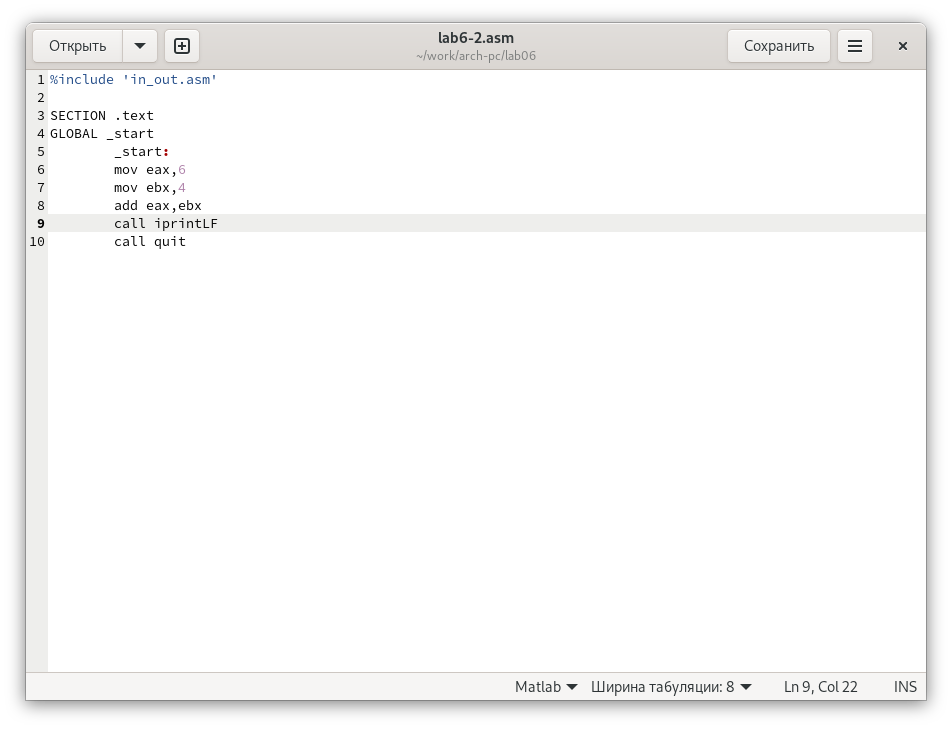
Файл lab6-2.asm

1. Создаём исполняемый файл и запускаем. Видим число 106.

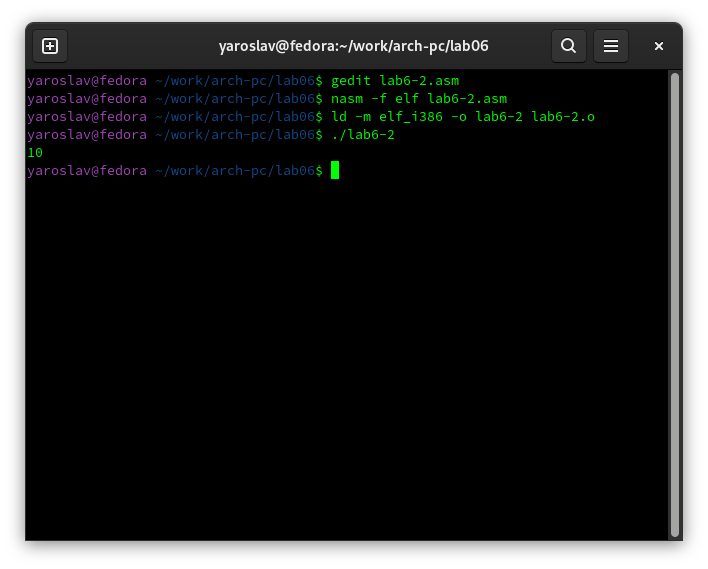


Работа программы

1. Удаляем одинарные кавычки (как в пункте 4).

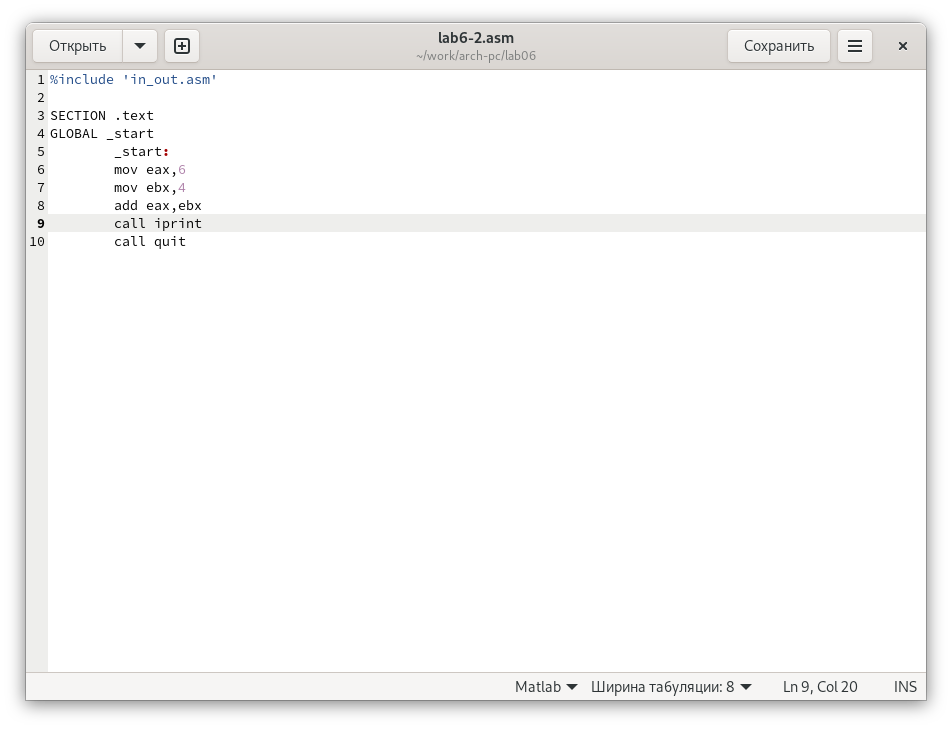


Изменённый lab6-2.asm

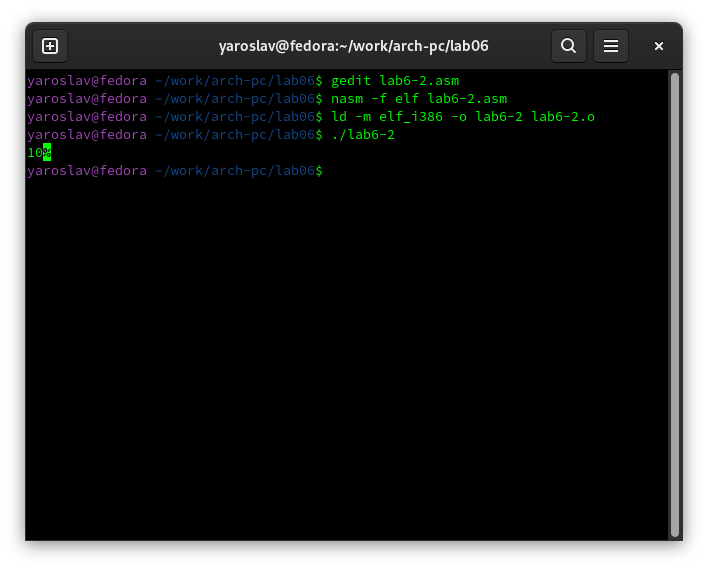


Работа программы

1. Меняем iprintLF на iprint. Вывод сработал без переноса на следующую строку.



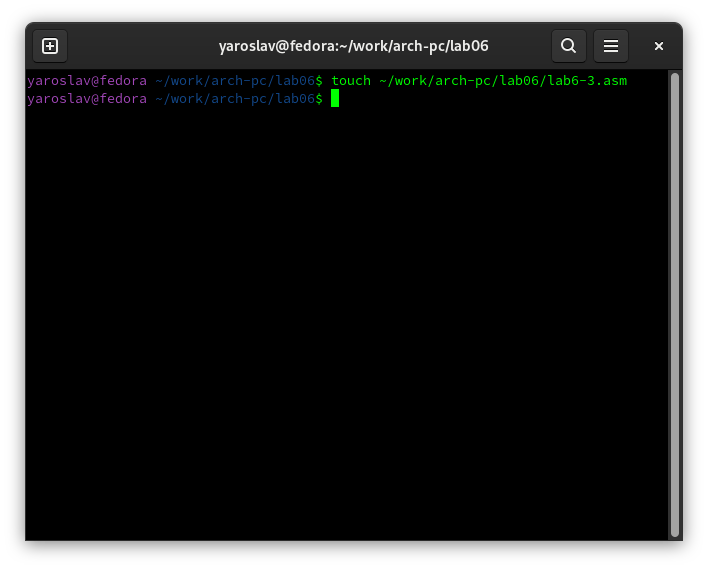
Изменённый lab6-2.asm



Работа программы

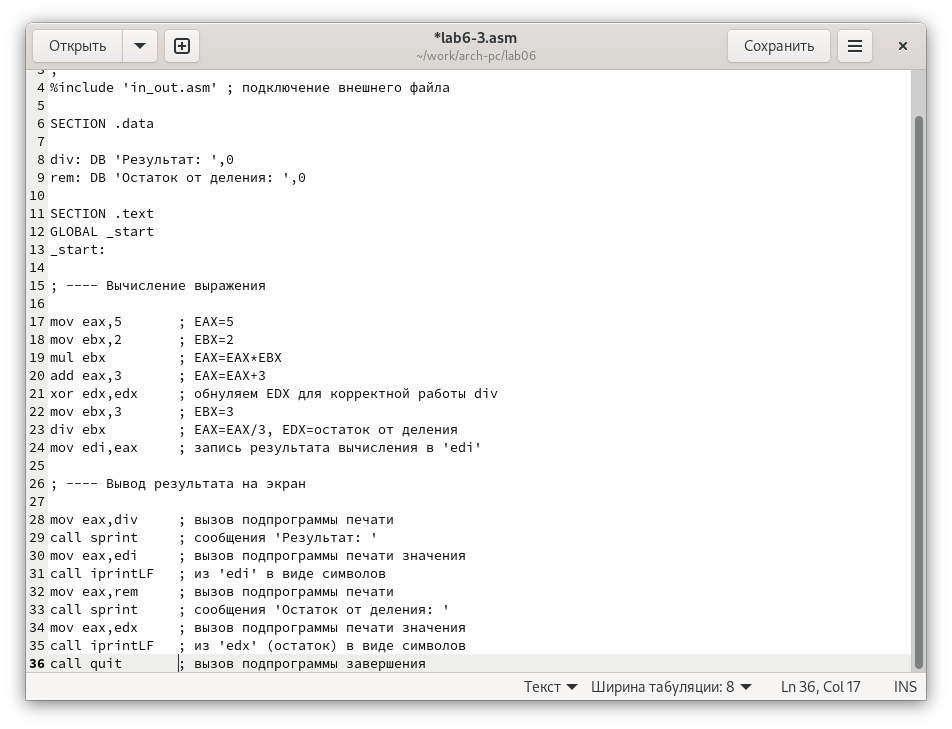
## 2.2 Выполнение арифметических операций в NASM

1. Создаём lab6-3.asm.



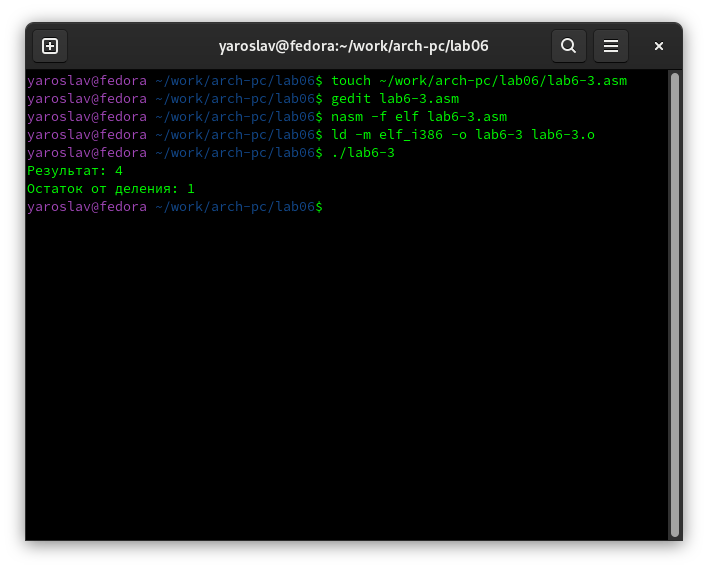
Создание файла

1. Копируем листинг в созданный файл.



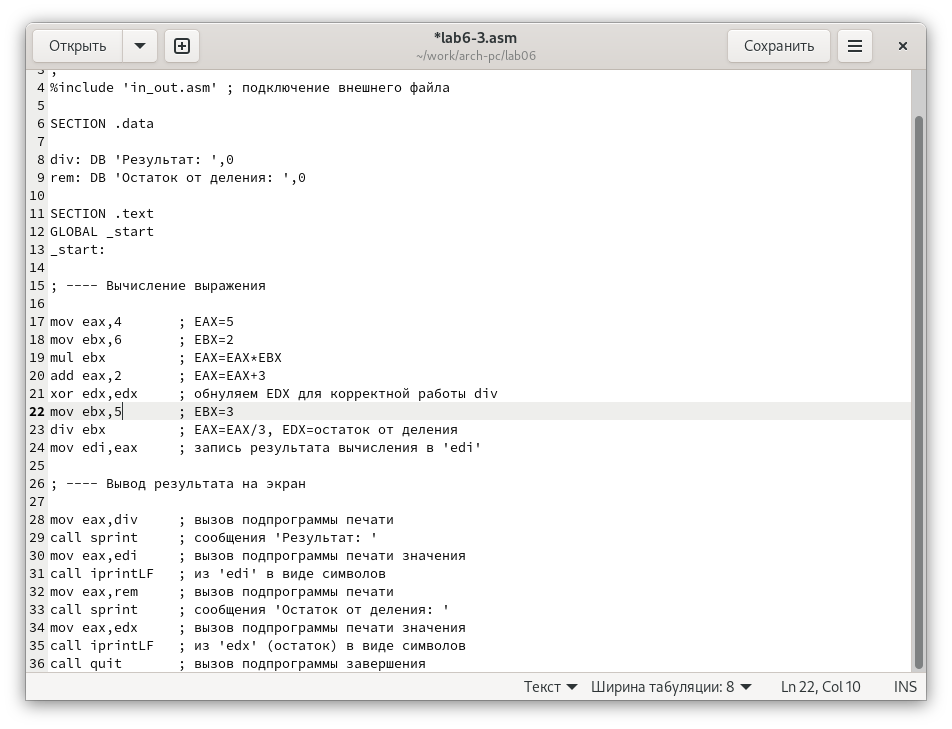
Готовый файл lab6-3.asm

1. Запускаем, получаем результат.

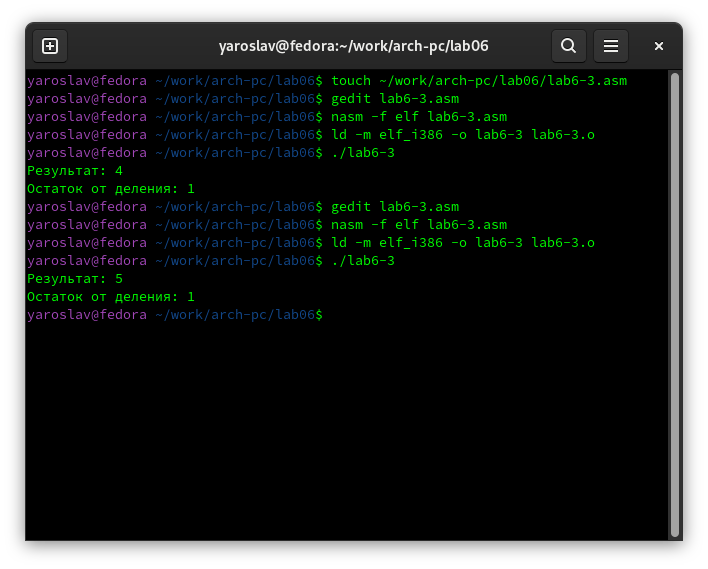


Работа программы

1. Меняем текст программы под новые условия. Запускаем.

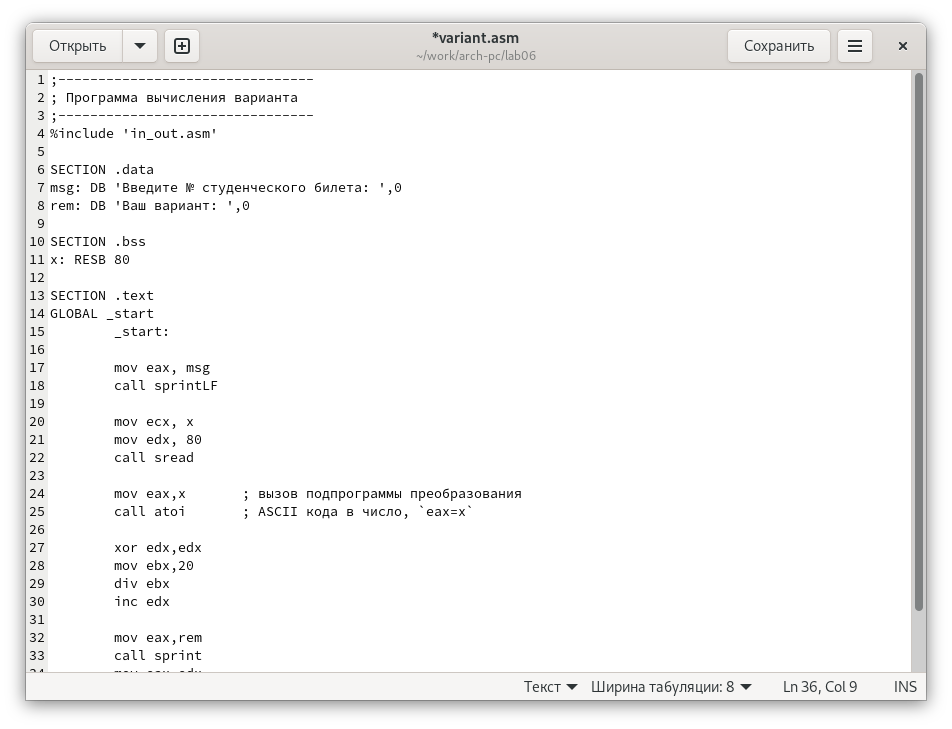


Изменённый файл lab6-3.asm

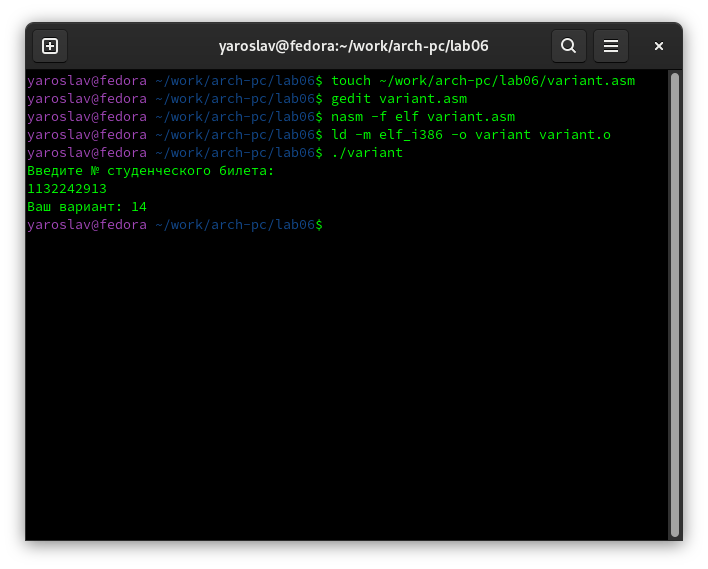


Работа программы

1. Создаём файл variant.asm и вставляем туда листинг 6.4. Запускаем и вводим номер студ.билета. Получаем номер варианта.



Файл variant.asm



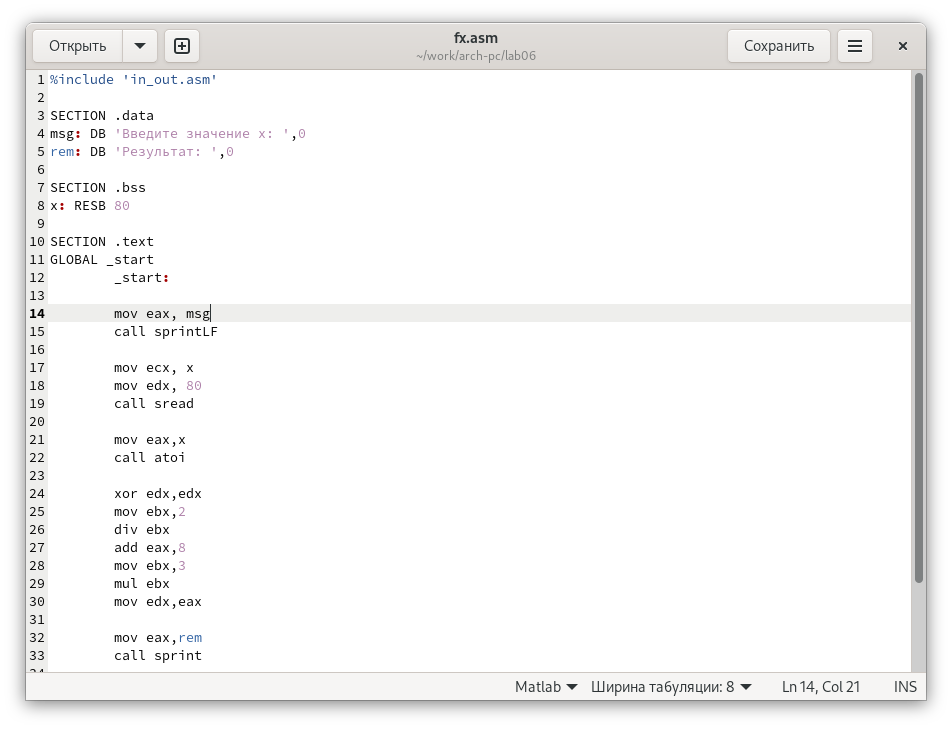
Работа программы

## 2.3 Ответы на вопросы

1. Строки mov eax,rem и call sprint отвечают за вывод на экран сообщения ‘Ваш вариант:’.
2. mov ecx, x ; запись в eax значения х mov edx, 80 ; запись в edx значения 80 call sread ; вызов функции чтения
3. Вызов функции atoi - функции преобразования ascii-кода символа в целое число.
4. Строки xor edx,edx mov ebx,20 div ebx inc edx отвечают за вычисления варианта.
5. Остаток от деления при выполнении инструкции “div ebx” записывается в edx.
6. inc edx увеличивает значение в edx на 1 и перезаписывает его в edx.
7. Строки mov eax,edx call iprintLF отвечают за вывод результата.

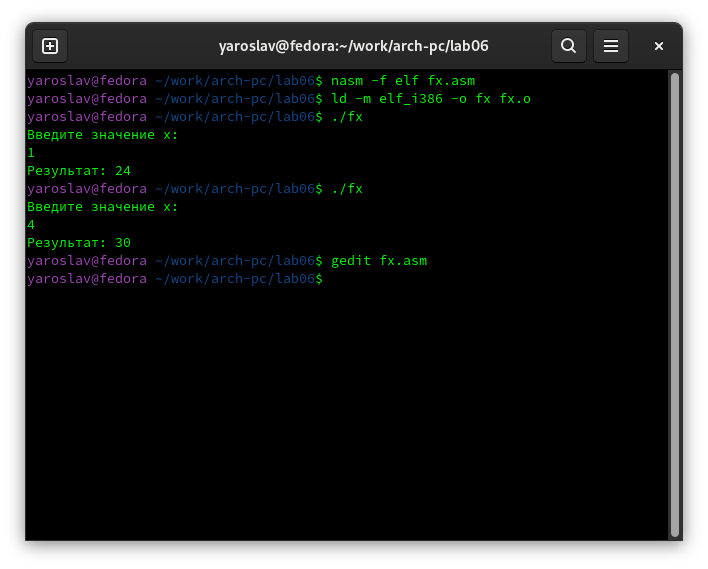
# 3 Выполнение самостоятельной работы

1. Создаём файл fx.asm и пишем в нём текст программы (14 вариант).



Готовый fx.asm

1. Проверяем работу на заданных числах.



Работа программы

# 4 Выводы

Были освоены арифметические операции в NASM