Отчет по лабораторной работе №6

Арифметические операции в NASM

Татьяна Александровна Буллер

Содержание

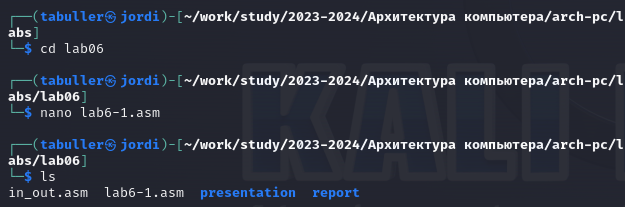
# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

# 2 Выполнение лабораторной работы

## 2.1 Задание 1:

Создайте каталог для программам лабораторной работы № 6, перейдите в него и создайте файл lab6-1.asm.

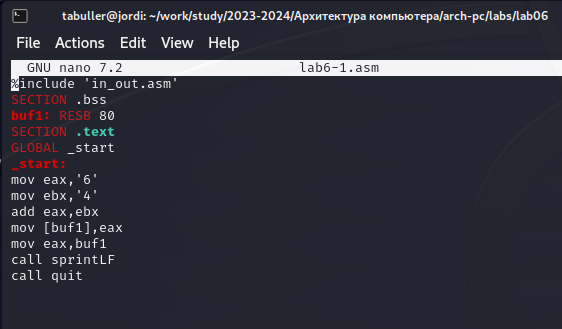


Переход в каталог курса и введение команды на создание файла

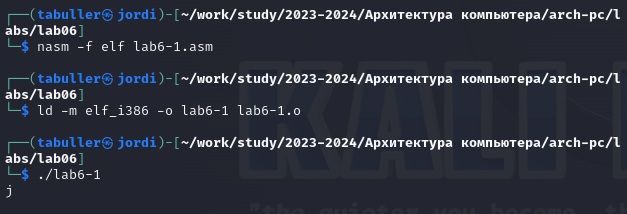
Для дальнейшего успешного выполнения программы из листинга необходимо, чтобы файл in\_out.asm находился в одном каталоге с рабочими файлами. Он был скопирован заранее, правильность копирования проверена с помощью команды ls.

## 2.2 Задание 2:

Введите в файл lab6-1.asm текст программы из листинга 6.1, создайте исполняемый файл и запустите его.



Копирование текста программы из листинга

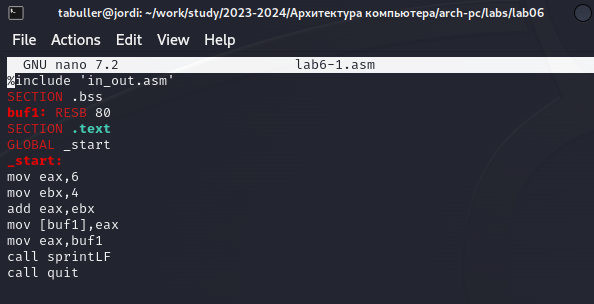


Создание и запуск исполняемого файла

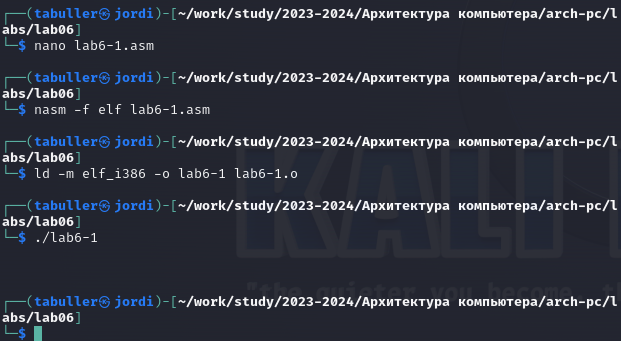
Чего и следовало ожидать, вместо числа 10 в результат выводится символ ‘j’: в регистр eax была записана сумма кодов символов (106), что соответствует коду символа ‘j’.

## 2.3 Задание 3:

Исправьте текст программы, записав в регистры числа. Создайте исполняемый файл и запустите его. Пользуясь таблицей ASCII определите какому символу соответствует код 10. Отображается ли этот символ при выводе на экран?



Исправленный текст программы

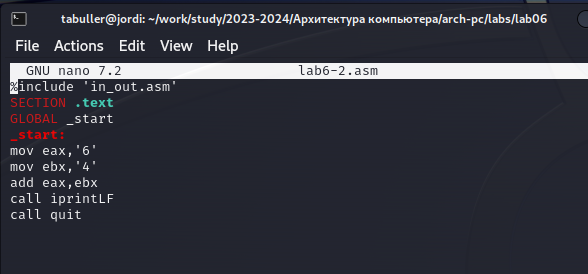


Создание и запуск исполняемого файла

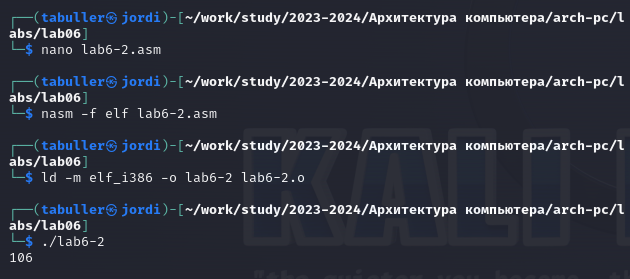
Коду 10 соответствует символ VT (vertical tab). Это прозрачный символ - что-то такое, собственно, и вывелось на экран.

## 2.4 Задание 4:

Преобразуем текст программы из Листинга 6.1 с использованием функций для преобразования ASCII символов в числа и обратно из файла in\_out.asm. Создайте файл lab6-2.asm и введите в него текст программы из листинга 6.2. Создайте исполняемый файл и запустите его.



Преобразованный текст программы

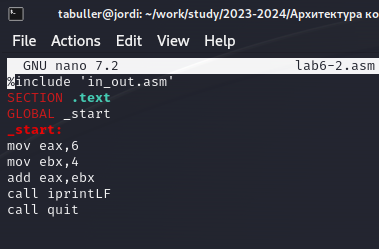


Создание и запуск исполняемого файла

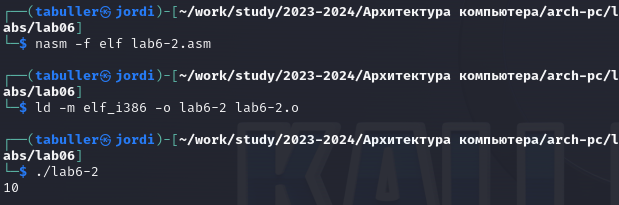
В результат вывелось число 106: складываются коды символов ‘6’ и ‘4’, но функция iprintLF выводит число, а не символ, кодом которого является это число.

## 2.5 Задание 5:

Аналогично предыдущему примеру изменим символы на числа. Создайте исполняемый файл и запустите его. Какой результат будет получен при исполнении программы?



Отредактированный текст программы

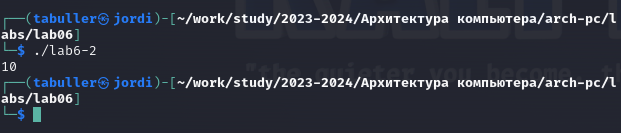


Создание и запуск исполняемого файла

В результат вывелось число 10: сложились сами числа, а не их коды; программа отработала так, как было запланированно изначально.

### 2.5.1 Задание 5.1:

Замените функцию iprintLF на iprint. Создайте исполняемый файл и запустите его. Чем отличается вывод функций iprintLF и iprint?



Вывод измененной программы

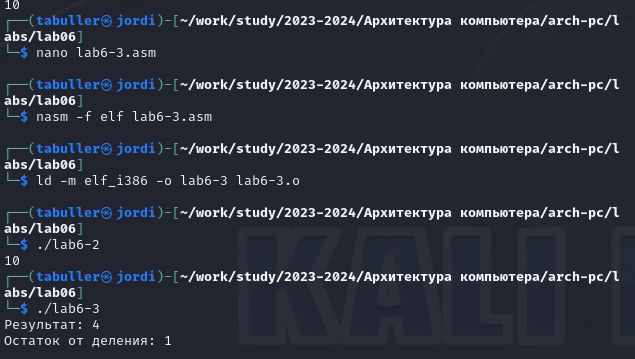
Программа скомпилировалась без значительных изменений, но замена sprintLF на sprint привела к тому, что исчез символ переноса строки при выводе сообщения на экран.

## 2.6 Задание 6:

Создайте файл lab6-3.asm и введите в него текст программы из листинга 6.3. Создайте исполняемый файл и запустите его.



Текст программы

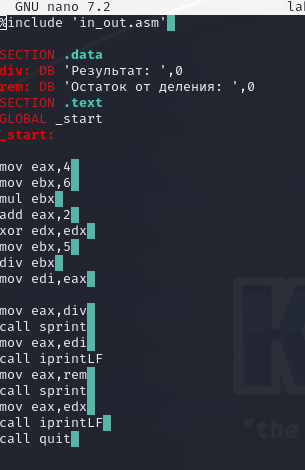


Создание и запуск исполняемого файла

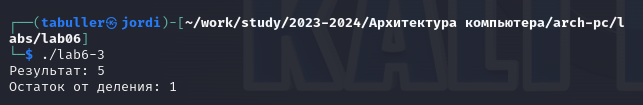
Результат программы соответствует заданной функции: 13/3 при целочисленном делении даст результат 4 с отстатком 1.

### 2.6.1 Задание 6.1:

Измените текст программы для вычисления выражения f(x) = (4 \* 6 + 2)/5. Создайте исполняемый файл и проверьте его работу.



Измененный екст программы

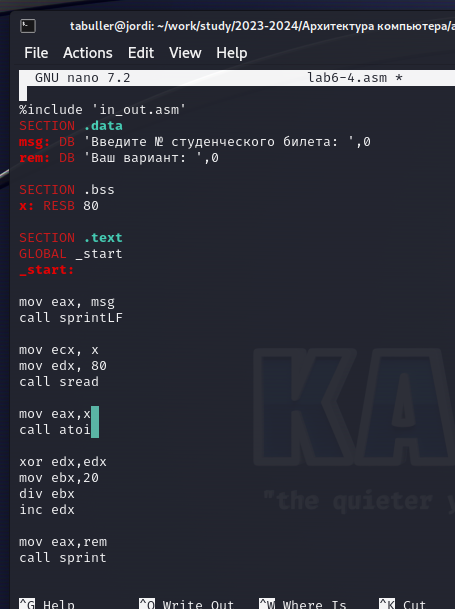


Создание и запуск исполняемого файла

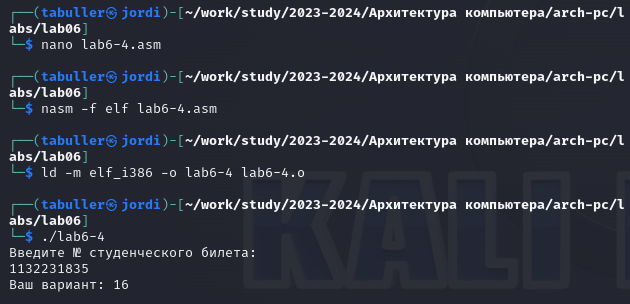
Для изменения функции необходимо только заменить значения переменных. Результат остается верным: 26/5 при целочисленном делении дает 5 с остатком 1.

## 2.7 Задание 7:

Создайте файл variant.asm и введите в него текст программы из листинга 6.4. Проверьте результат работы программы вычислив номер варианта аналитически.

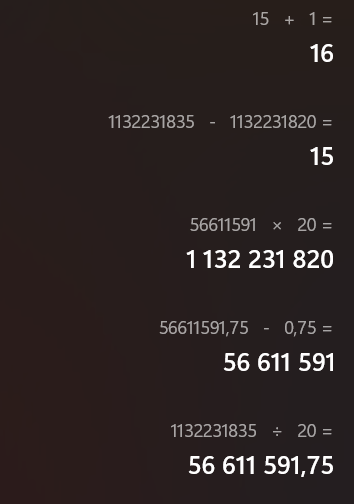


Текст программы для определения варианта



Создание и запуск исполняемого файла

Программа отрабатывает без ошибок, выводя в результат номер варианта 16. Проверим результат аналитически:



Проверка калькулятором

Результаты совпадают: можно утверждать, что программа отработала верно при заданном номере студенческого билета.

## 2.8 Задание 8:

Включите в отчет по выполнению лабораторной работы ответы на следующие вопросы:

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

mov eax,rem call sprint

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

mov ecx, x mov edx, 80 call sread

Эти инструкции выполняют считывание ввода пользователя, в дальнейшем полученное значение передается в переменную eax.

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

Для преобразования ASCII символов в числа.

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

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

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

edx

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

Увеличение на 1 значения регистра edx.

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

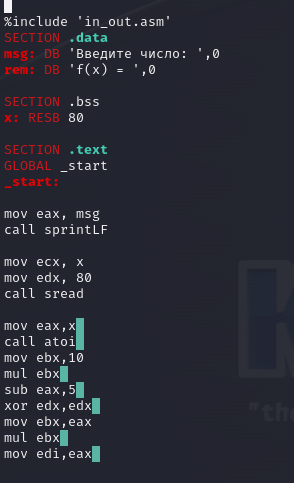
mov eax,edx call iprintLF

# 3 Задание для самостоятельной работы

## 3.1 Задание 8:

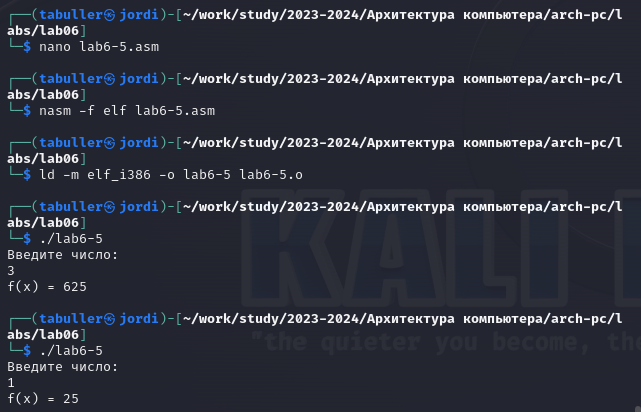
Написать программу вычисления выражения y = f(x). Программа должна выводить выражение для вычисления, выводить запрос на ввод значения x, вычислять заданное выражение в зависимости от введенного x, выводить результат вычислений. Вид функции f(x) выбрать из таблицы 6.3 вариантов заданий в соответствии с номером полученным при выполнении лабораторной работы. Создайте исполняемый файл и проверьте его работу для значений x1 и x2 из 6.3.

Функция в варианте 16: (10x − 5)^2. Разберем алгоритм: сперва программа должна получить ввод от пользователя (x), затем умножить полученное число на 10, вычесть из результата 5 и умножить новый результат сам на себя. Функции умножения и считывания ввода пользователя уже известны, функция для вычитания - sub, ee синтаксис аналогичесн синтаксису команды add.



Код новой программы

Первый блок команд после команды “старт” вызывает сообщение ‘Введите число’. Следующий блок считывает введенное число. Далее переменной eax присваивается значение x, ebx - 10, после чего выполняется умножение ebx на eax. Из результата в переменной eax вычитается 5, после чего полученное число умножается само на себя.



Компиляция программы

Для предложенных чисел 3 и 1 программа выдает результаты 625 и 25 соответственно. Это сходится с результатами, которые можно получить при самостоятельном решении: программа отработала верно.

# 4 Вывод

При выполнении лабораторной работы были освоены арифметические инструкции языка ассемблера NASM.