Отчёта по лабораторной работе

Дисциплина: Архитектура ЭВМ

Перегудов Александр Вадимович

Содержание

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

# 2 Задание

Здесь приводится описание задания в соответствии с рекомендациями методического пособия и выданным вариантом.

# 3 Теоретическое введение

# 4 Выполнение лабораторной работы

1. Создайте каталог для программ лабораторной работы № 6, перейшёл в него и создал файл lab6-1.asm. (рис. [1](#fig:001))

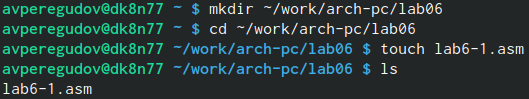


Figure 1: Создание каталога и файла

1. Использую gedit для редактирование файла lab6-1.asm (рис. [2](#fig:002)).

Figure 2: Открытие gedit с помощью консоли

Figure 2: Открытие gedit с помощью консоли

1. Ввел в файл lab6-1.asm текст программы из листинга 6.1 (рис. [3](#fig:003)).

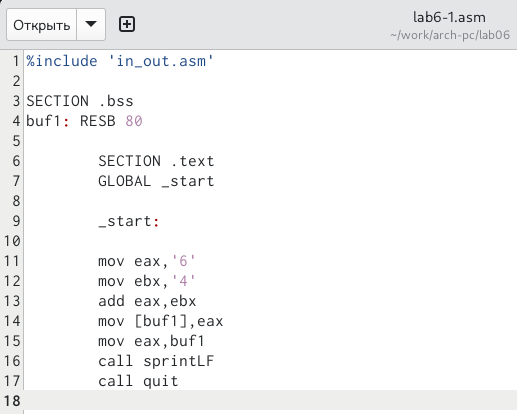


Figure 3: Текст программы

1. Создал исполняемый файл и запустил его. (рис. [4](#fig:004))

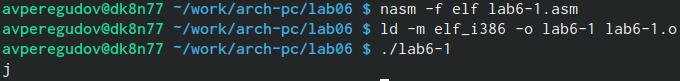
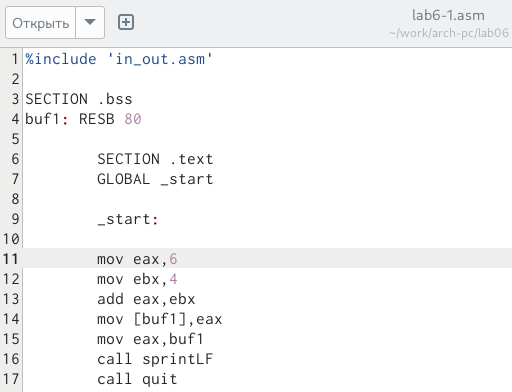


Figure 4: Результат работы программы

1. Изменил текст программы и вместо символов, записал в регистры числа. (рис. ??)

. {#fig:005 width=70%}

1. Создал исполняемый файл и запустил его. {[5](#fig:006) width=70%}

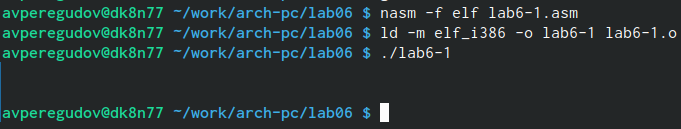


Figure 5: Результат работы программы

1. Создал файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 {[6](#fig:007) width=70%}.

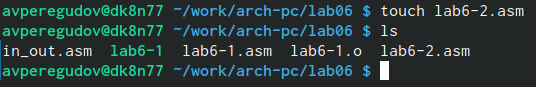


Figure 6: Создание файла с помощью команды touch

1. Ввел в файл lab6-2.asm текст программы из листинга 6.2 {?? width=70%}.

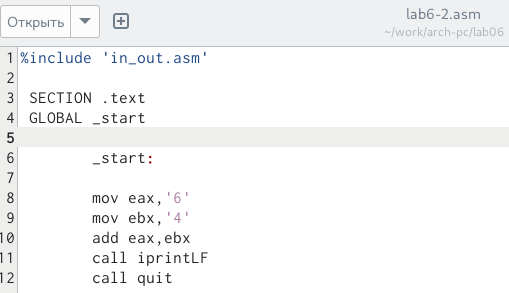


Figure 7: Текст программы

1. Создал исполняемый файл и запустил его.{рис. [8](#fig:09)}

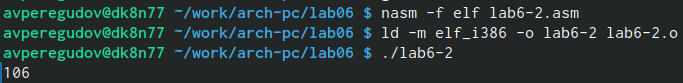


Figure 8: Результат работы программы

1. Изменил текст программы и вместо символов, записал в регистры числа.(рис. [9](#fig:010)).

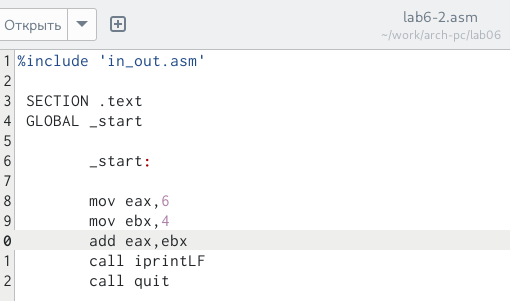


Figure 9: Изменёный текст программы

1. Создал исполняемый файл и запустил его.{[10](#fig:011) width=70%}

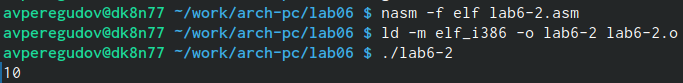


Figure 10: Результат работы программы

1. Заменил функцию iprintLF на iprint.{[11](#fig:012) width=70%}

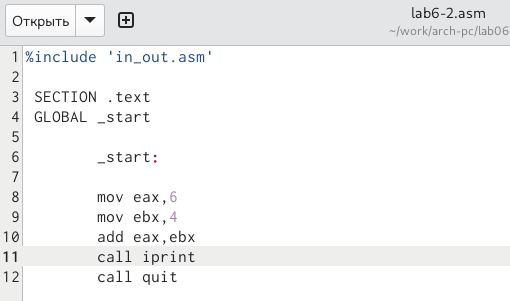


Figure 11: Изменёный текст программы

1. Создал исполняемый файл и запустил его.{[12](#fig:013) width=70%}

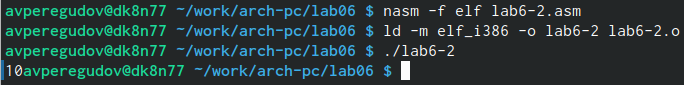


Figure 12: Результат работы программы

1. Создайте файл lab6-3.asm в каталоге ~/work/arch-pc/lab06.{[13](#fig:014) width=70%}

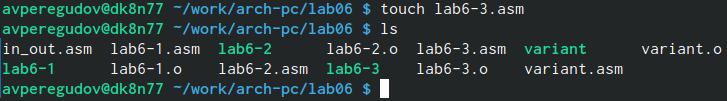


Figure 13: Создание файла с помощью команды touch

1. Ввел в файл lab6-3.asm текст программы из листинга 6.3.{[14](#fig:015) width=70%}

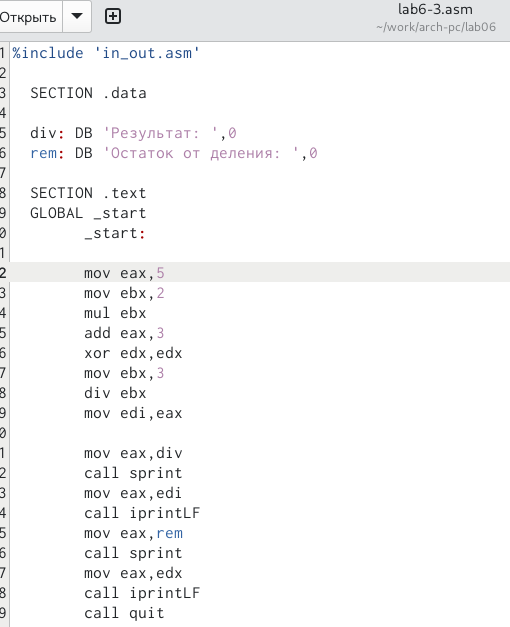


Figure 14: Текст программы

1. Создал исполняемый файл и запустил его.{[15](#fig:016) width=70%}

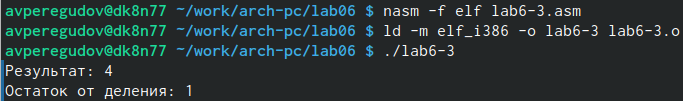


Figure 15: Результат работы программы

1. Изменил текст программы для вычисления результата другого выражения.{[16](#fig:0-1) width=70%}

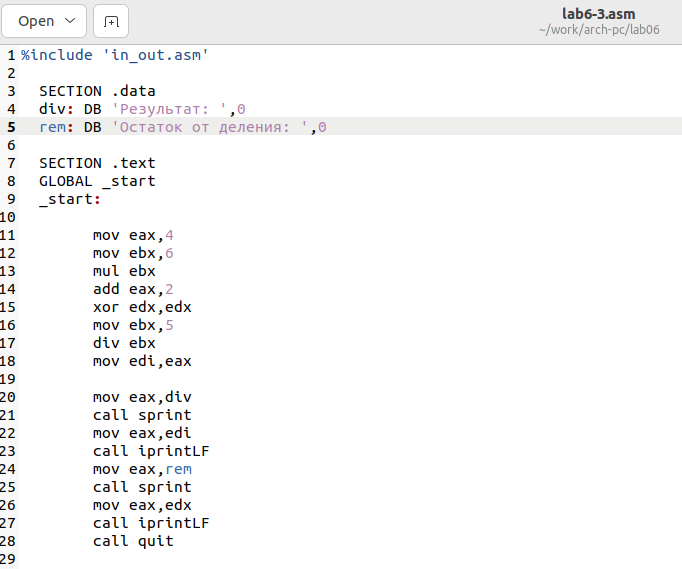


Figure 16: Изменёный текст программы

1. Создал исполняемый файл и запустил его.{[17](#fig:0-2) width=70%}

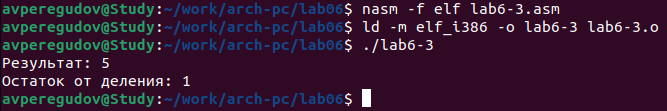


Figure 17: Результат работы программы

1. Создал файл variant.asm в каталоге ~/work/arch-pc/lab06.{[18](#fig:017) width=70%}

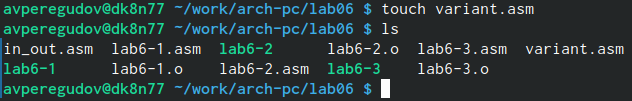


Figure 18: Создание файла с помощью команды touch

1. Ввел в файл variant.asm текст программы из листинга 6.4.{[19](#fig:018) width=70%}

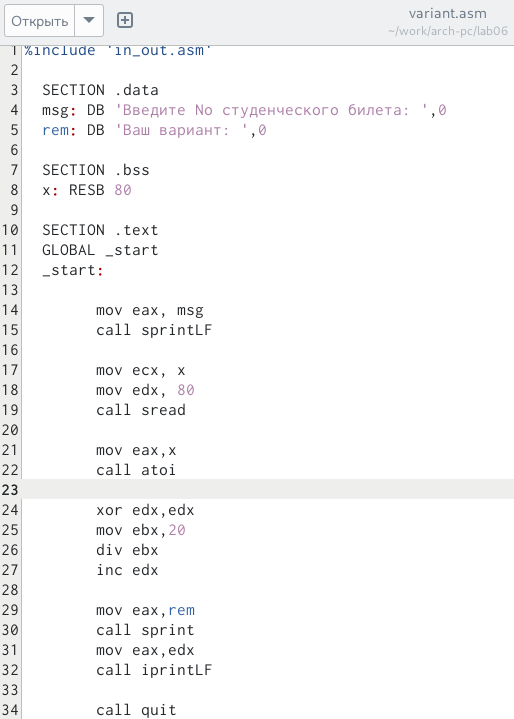


Figure 19: Проверка работоспособности

1. Создал исполняемый файл, запустил его и ввёл номер своего студенческого билета.{[20](#fig:019) width=70%}

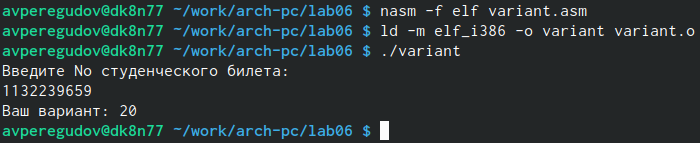


Figure 20: Результат работы программы

1. Создал файл lab6-4.asm в каталоге ~/work/arch-pc/lab06.{[21](#fig:1-1) width=70%}

Figure 21: Создание файла с помощью команды touch

Figure 21: Создание файла с помощью команды touch

1. Ввёл код для вычисления значения функции от x из таблицы 6.3 под номером 20.{[22](#fig:1-2) width=70%}

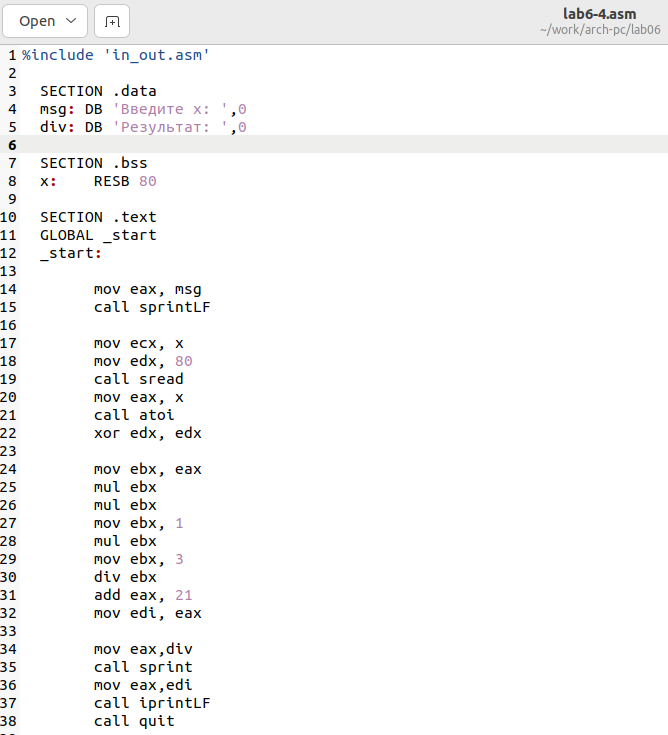


Figure 22: Текст программы

1. Создал исполняемый файл.{[23](#fig:1-3) width=70%}

Figure 23: Трансляция и компановка

Figure 23: Трансляция и компановка

1. Запустил исполняемый файл и ввёл первое значение x.{[24](#fig:1-4) width=70%}

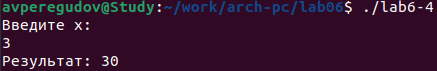


Figure 24: Результат работы программы

1. Запустил исполняемый файл и ввёл второе значение x.{[25](#fig:1-5) width=70%}

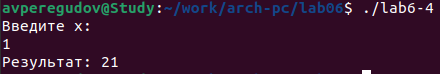


Figure 25: Результат работы программы

#Ответы на вопросы

#1.

mov eax, rem call sprint

#2. Для считывания сообщения.

#3. Для преобразования ascii-код символа в целое число.

#4.

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

#5. edx.

#6. Для прибавления к единицы к значению лежащему в edx.

#7.

mov eax,rem call sprint mov eax,edx call iprintLF

# 5 Выводы

В этой работе были освоены арифметические инструкции языка ассемблера NASM.

# Список литературы