Отчёт по лабораторной работе №7.Арифметические операции в NASM.

Королёв Иван Андреевич

Содержание

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

# 2 Теоретическое введение

1. Адресация в NASM

Большинство инструкций на языке ассемблера требуют обработки операндов. Адрес операнда предоставляет место, где хранятся данные, подлежащие обра- ботке.Этомогутбытьданныехранящиесяврегистреиливячейкепамяти.Далее рассмотрены все существующие способы задания адреса хранения операндов – способы адресации. Существуеттри 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 непосред- ственно в команде, Например: mov ax,2.
* Адресация памяти–операндзадаетадресвпамяти.Вкомандеуказывает- ся 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 выде- лена 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.

# 3 Выполнение лабораторной работы

# 4 Символьные и численные данные в NASM

1. Создаю каталог для программам лабораторной работы № 7, перейду в него и создам файл lab7-1.asm:
2. Рассмотрим примеры программ вывода символьных и численных значе- ний. Программы будут выводить значения записанные в регистр eax.Листинг 7.11,2

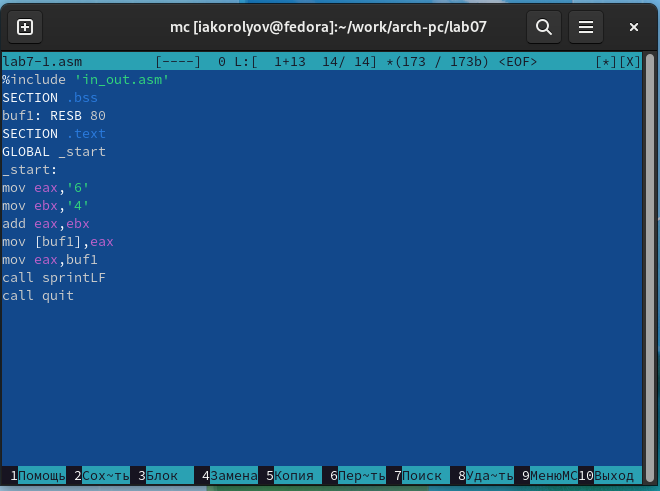


Рис. 1: lab7-1

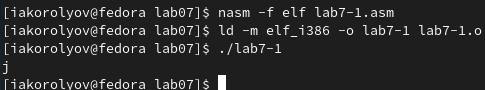


Рис. 2: lab7-1

1. Далее изменим текст программы и вместо символов, запишем в реги- стры числа. Исправьте текст программы (Листинг 1) следующим образом. Не отображается3,4

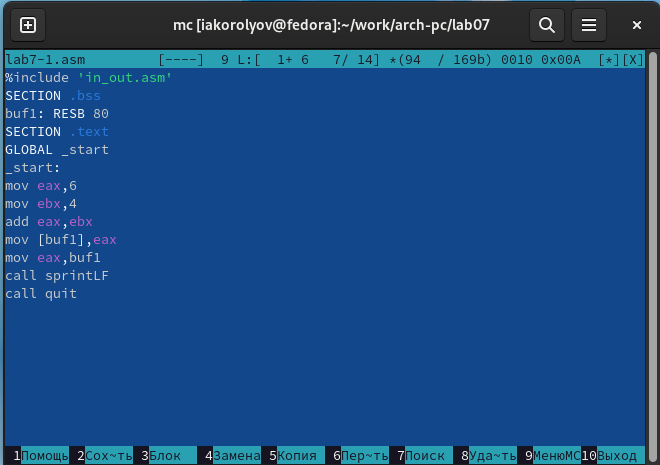


Рис. 3: lab7-1

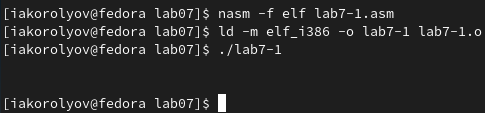


Рис. 4: lab7-1

1. Создаю файл lab7-2.asm в каталоге ~/work/arch-pc/lab07 и введу в него текст программы из листинга 7.2.5,6

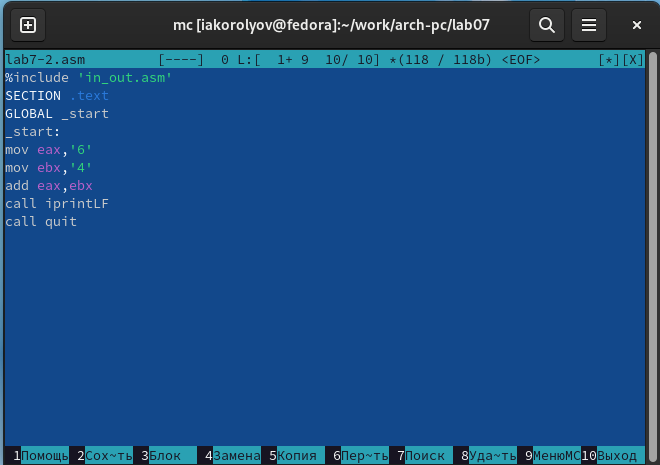


Рис. 5: lab7-2

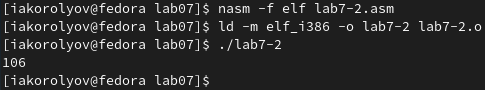


Рис. 6: lab7-2

1. Аналогично предыдущему примеру изменим символы на числа. Замените строки. Не переходит на новую строку.7,8

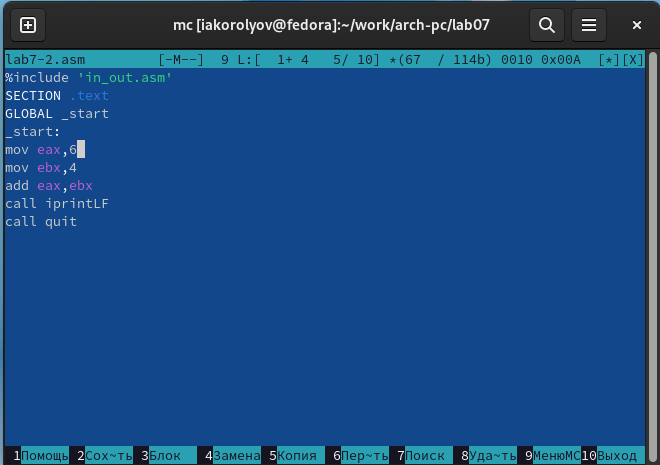


Рис. 7: lab7-2

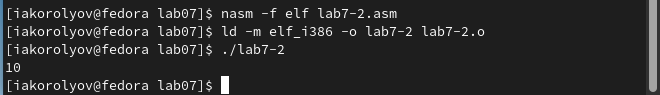
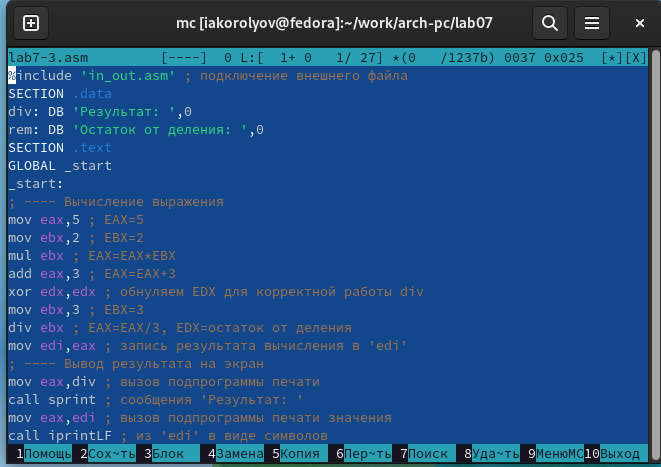


Рис. 8: lab7-2

# 5 Выполнение арифметических операций в NASM

1. В качестве примера выполнения арифметических операций в NASM при- ведем программу вычисления арифметического выражения. **¿fig:009?**,9,10,11 

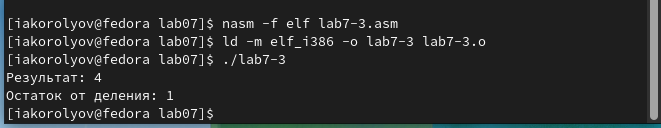


Рис. 9: lab7-3

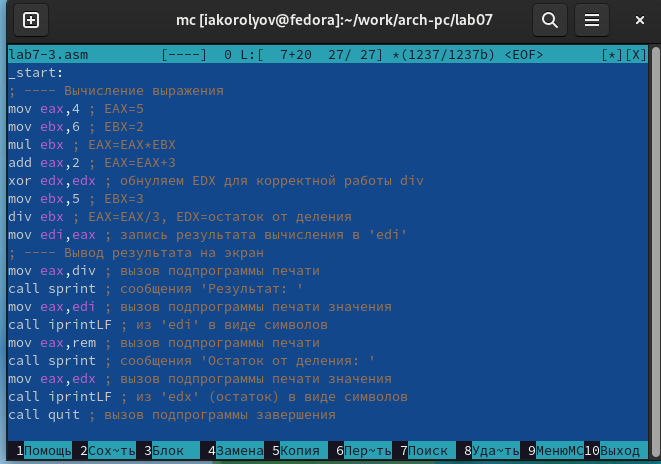


Рис. 10: lab7-3

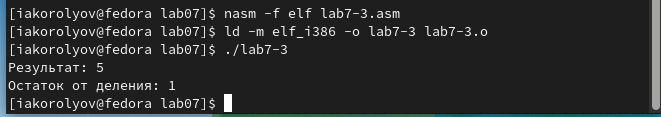


Рис. 11: lab7-3

1. В качестве другого примера рассмотрим программу вычисления варианта задания по номеру студенческого билета, работающую по следующему алгоритму12,13

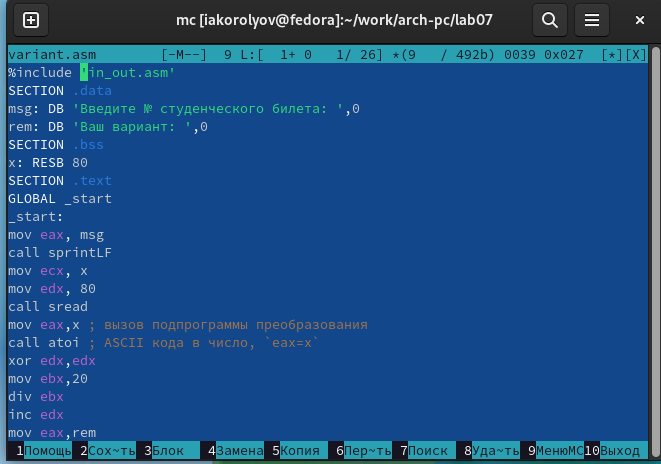


Рис. 12: variant

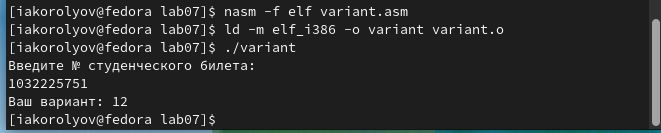


Рис. 13: variant

Ответы на вопросы:

1.mov eax,rem call sprint

2.Ввод переменной x

3.Преобразует строку string в целое значение типа int.

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

5.В mov edi,eax

6.inc уеличивает на 1 свой операнд

7.mov eax,edx call iprintLF

1. Самостоятельная работа:14,15

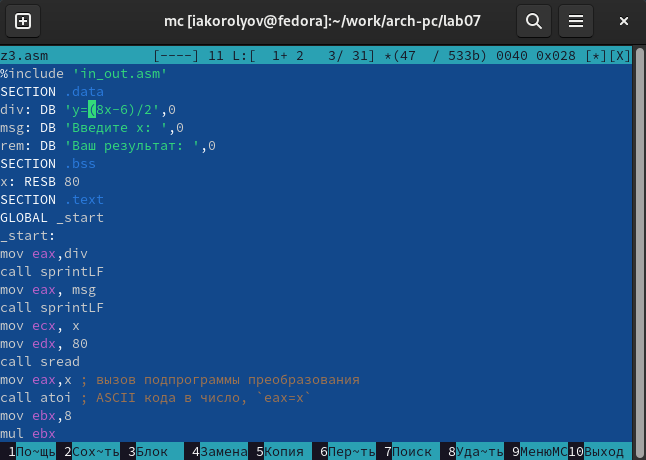


Рис. 14: z3

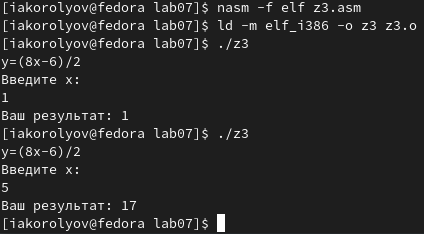


Рис. 15: z3

# 6 Выводы

Я освоил арифметические инструкций языка ассемблера NASM.