Отчёт по лабораторной работе №6

Дисциплина: Архитектура компьютера

Абакумов Тимофей Александрович

Содержание

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

# 2 Задание

**Порядок выполнения лабораторной работы**

1. Символьные и численные данные в NASM.
2. Выполнение арифметических операций в NASM.
3. Выполнение заданий для самостоятельной работы.

# 3 Выполнение лабораторной работы

## 3.1 Символьные и численные данные в NASM

1. Для начала создадим каталог для программам лабораторной работы № 6, перейдём в него и создадим файл lab6-1.asm с помощью утилиты touch (рис. 1).

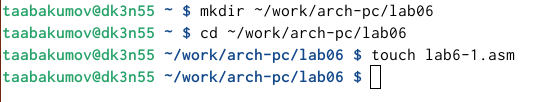


Рис. 1: Создание нового файла

1. Копирую в текущий каталог файл in\_out.asm с помощью утилиты cp, т.к. он будет использоваться в других программах (рис. 2).

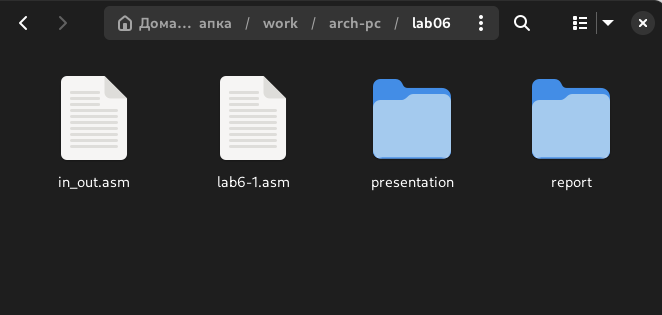


Рис. 2: Копирование файла

1. После этого откроем созданный файл lab6-1.asm, вставим в него программу вывода значения регистра eax (рис. 3).

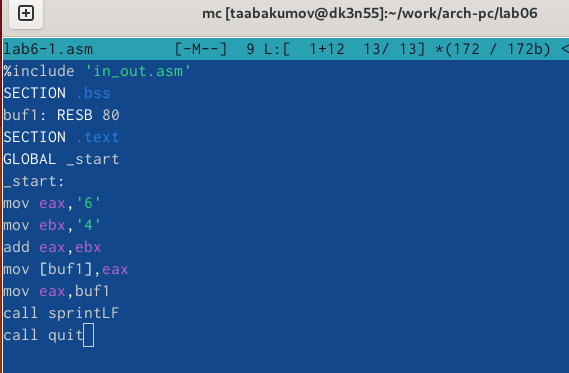


Рис. 3: Вставка программы

Код программы из пункта 3:

%include ‘in\_out.asm’

SECTION .bss

buf1: RESB 80

SECTION .text

GLOBAL \_start

\_start:

mov eax,‘6’

mov ebx,‘4’

add eax,ebx

mov [buf1],eax

mov eax,buf1

call sprintLF

call quit

1. Создадим исполняемый файл программы и запустим его. Вывод программы: символ j, потому что программа вывела символ, соответствующий по системе ASCII сумме двоичных кодов символов 4 и 6 (рис. 4).

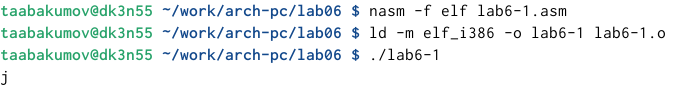


Рис. 4: Запуск программы

1. Далее изменим в тексте программы символы “6” и “4” на цифры 6 и 4 (рис. 5).

Рис. 5: Изменение программы

Рис. 5: Изменение программы

1. Создадим новый исполняемый файл программы и запустим его. Теперь вывелся символ с кодом 10, это символ перевода строки, этот символ не отображается при выводе на экран (рис. 6).

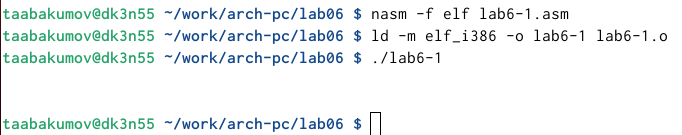


Рис. 6: Запуск программы

1. Создадим новый файл lab6-2.asm с помощью утилиты touch (рис. 7).

Рис. 7: Создание нового файла

Рис. 7: Создание нового файла

1. Введём в файл текст другой программы для вывода значения регистра eax (рис. 8).

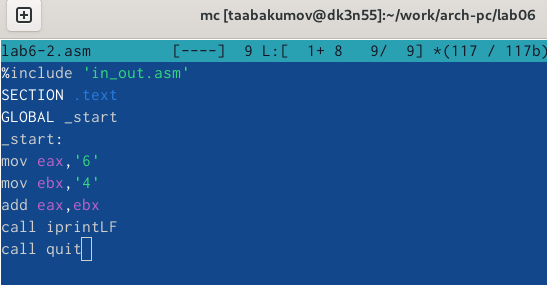


Рис. 8: Создание нового файла

Код программы из пункта 8:

%include ‘in\_out.asm’

SECTION .text

GLOBAL \_start

\_start:

mov eax,‘6’

mov ebx,‘4’

add eax,ebx

call iprintLF

call quit

1. Создадим и запустим исполняемый файл lab6-2 (рис. 10). Теперь вывод число 106, потому что программа позволяет вывести именно число, а не символ, хотя все еще происходит именно сложение кодов символов “6” и “4” (рис. 9).

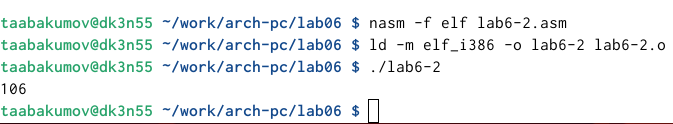


Рис. 9: Запуск программы

1. Заменим в тексте программы в файле lab6-2.asm символы “6” и “4” на числа 6 и 4. Заменим в тексте программы функцию iprintLF на iprint (рис. 13). (рис. 10).

Рис. 10: Изменение программы

Рис. 10: Изменение программы

1. Создадим и запустим новый исполняемый файл.Теперь программа складывает не соответствующие символам коды в системе ASCII, а сами числа, поэтому вывод 10 (рис. 11).

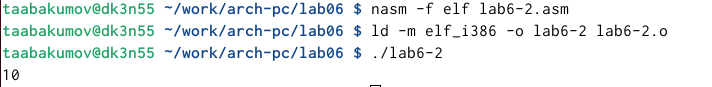


Рис. 11: Запуск программы

## 3.2 Выполнение арифметических операций в NASM

1. Создадим файл lab6-3.asm с помощью утилиты touch (рис. 12).

Рис. 12: Создание нового файла

Рис. 12: Создание нового файла

1. Введём в созданный файл текст программы для вычисления значения выражения f(x) = (5 \* 2 + 3)/3 (рис. 13).

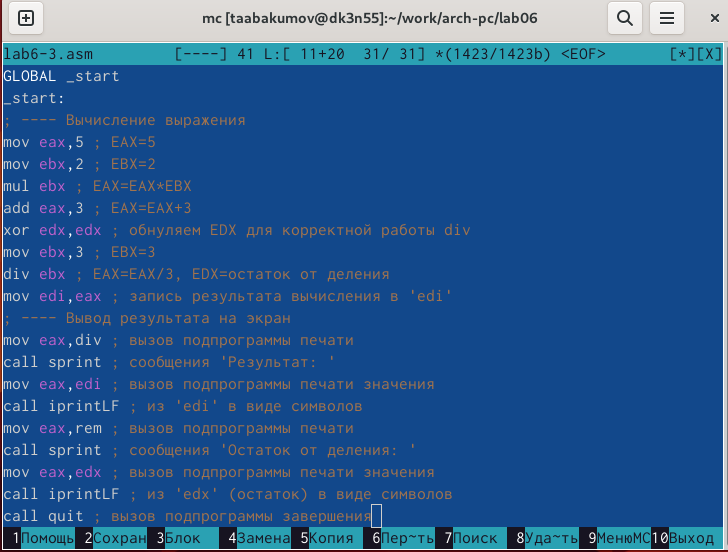


Рис. 13: Изменение программы

Код программы из пункта 13:

%include ‘in\_out.asm’ ; подключение внешнего файла

SECTION .data

div: DB ‘Результат:’,0

rem: DB ‘Остаток от деления:’,0

SECTION .text

GLOBAL \_start

\_start:

mov eax,5 ; EAX=5

mov ebx,2 ; EBX=2

mul ebx ; EAX=EAX\*EBX

add eax,3 ; EAX=EAX+3

xor edx,edx ; обнуляем EDX для корректной работы div

mov ebx,3 ; EBX=3

div ebx ; EAX=EAX/3, EDX=остаток от деления

mov edi,eax ; запись результата вычисления в ‘edi’

mov eax,div ; вызов подпрограммы печати

call sprint ; сообщения ‘Результат:’

mov eax,edi ; вызов подпрограммы печати значения

call iprintLF ; из ‘edi’ в виде символов

mov eax,rem ; вызов подпрограммы печати

call sprint ; сообщения ‘Остаток от деления:’

mov eax,edx ; вызов подпрограммы печати значения

call iprintLF ; из ‘edx’ (остаток) в виде символов

call quit ; вызов подпрограммы завершения

1. Создаю исполняемый файл и запускаю его (рис. 14).

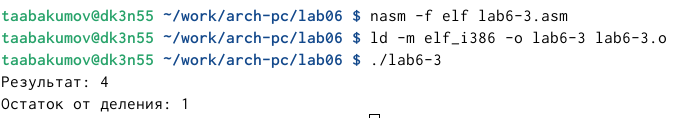


Рис. 14: Запуск программы

1. Изменим программу так, чтобы она вычисляла значение выражения f(x) = (4 \* 6 + 2)/5 (рис. 15).

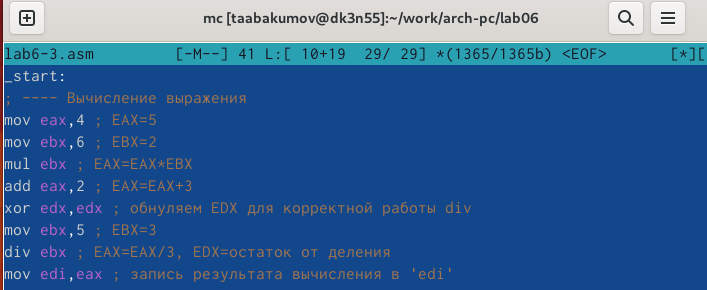


Рис. 15: Изменение программы

1. Создадим и запустм новый исполняемый файл. Я посчитал для проверки правильности работы программы значение выражения самостоятельно, программа отработала верно (рис. 16).

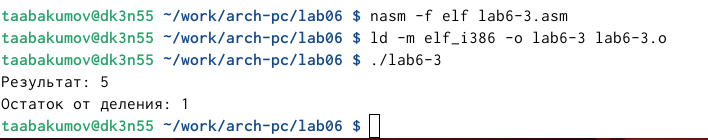


Рис. 16: Запуск программы

1. Создадим файл variant.asm с помощью утилиты touch (рис. 17).

Рис. 17: Создание файла

Рис. 17: Создание файла

1. Введём в файл текст программы для вычисления варианта задания по номеру студенческого билета (рис. 18).

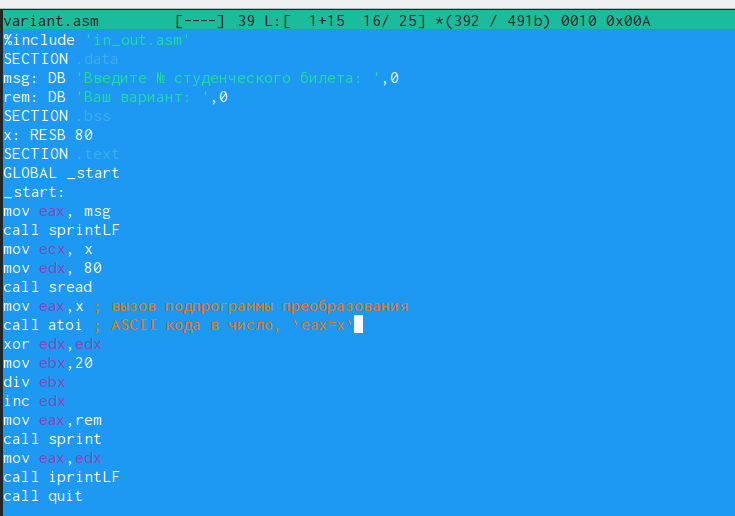


Рис. 18: Изменение программы

Код программы из пункта 18:

%include ‘in\_out.asm’

SECTION .data

msg: DB ‘Введите № студенческого билета:’,0

rem: DB ‘Ваш вариант:’,0

SECTION .bss

x: RESB 80

SECTION .text

GLOBAL \_start

\_start:

mov eax, msg

call sprintLF

mov ecx, x

mov edx, 80

call sread

mov eax,x ; вызов подпрограммы преобразования

call atoi ; ASCII кода в число, eax=x

xor edx,edx

mov ebx,20

div ebx

inc edx

mov eax,rem

call sprint

mov eax,edx

call iprintLF

call quit

1. Создаю и запускаю исполняемый файл. Ввожу номер своего студ. билета с клавиатуры, программа вывела, что мой вариант - 9 (рис. 19).

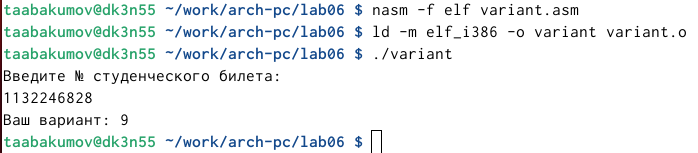


Рис. 19: Запуск программы

## 3.3 Выполнение заданий для самостоятельной работы

1. Создадим файл lab6-4.asm с помощью утилиты touch (рис. 20).

Рис. 20: Создание файла

Рис. 20: Создание файла

1. Откроем созданный файл для редактирования, введём в него текст программы для вычисления значения выражения 10+(31х-5). Это выражение было под вариантом 9 (рис. 21).

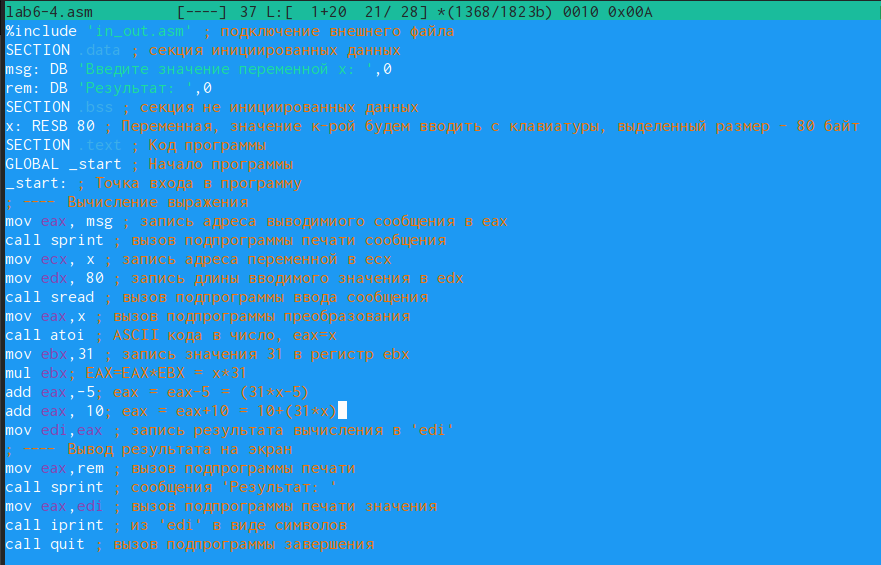


Рис. 21: Изменение программы

1. Создаю и запускаю исполняемый файл. При вводе значения 1, вывод - 36. Проведём еще один запуск исполняемого файла для проверки работы программы с другим значением на входе. При вводе значения 3, вывод - 98. Программа отработала верно (рис. 22).

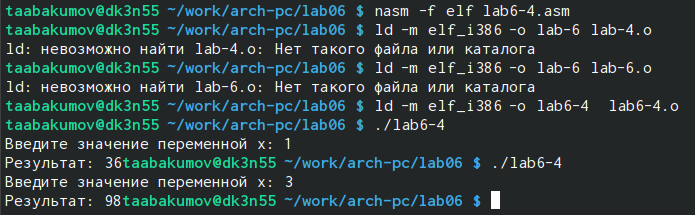


Рис. 22: Запуск программы

# 4 Выводы

При выполнении данной лабораторной работы я освоил арифметические инструкции языка ассемблера NASM.