Отчет по лабораторной работе №5

Дисциплина: архитектура компьютера

Адмиральская Александра Андреевна

Содержание

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

# 2 Задание

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

# 3 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы.

Например, в табл. 1 приведено краткое описание стандартных каталогов Unix.

Таблица 1: Описание некоторых каталогов файловой системы GNU Linux

| Имя каталога | Описание каталога |
| --- | --- |
| / | Корневая директория, содержащая всю файловую |
| /bin | Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям |
| /etc | Общесистемные конфигурационные файлы и файлы конфигурации установленных программ |
| /home | Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя |
| /media | Точки монтирования для сменных носителей |
| /root | Домашняя директория пользователя root |
| /tmp | Временные файлы |
| /usr | Вторичная иерархия для данных пользователя |

Более подробно про Unix см. в [1–4].

# 4 Выполнение лабораторной работы

Для начала создаем каталог для программам лабораторной работы № 6 и переходим в него (рис. 1).

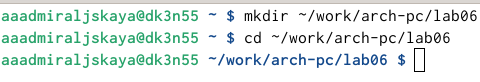


Рис. 1: Создание каталога lab06

Перейдя в каталог, создаем файл lab6-1.asm (рис. 2).



Рис. 2: Создание файла

Затем копируем в текущий каталог файл in\_out.asm с помощью утилиты cp, т.к. он будет использоваться в других программах (рис. 3).

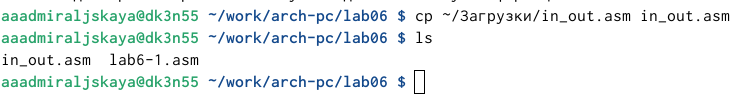


Рис. 3: Копирование файла

Открываем созданный файл lab6-1.asm, вставляем в него программу вывода значения регистра eax (рис. 4).

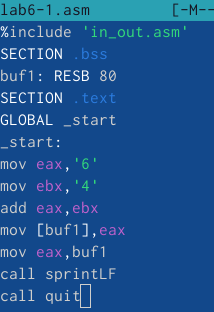


Рис. 4: Программа в файле

Создаем исполняемый файл и запускаем его. Программа выводит символ j (рис. 5).

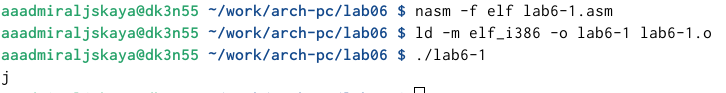


Рис. 5: Создание и запуск исполняемого файла

Изменяем в тексте программы символы “6” и “4” на цифры 6 и 4 (рис. 6).

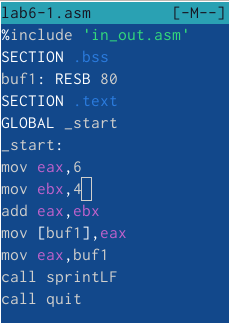


Рис. 6: Изменение текста программы

Создаем новый исполняемый файл программы и запускаем его. Теперь программа выводит символ с кодом 10, это символ перевода строки, этот символ не отображается при выводе на экран (рис. 7).

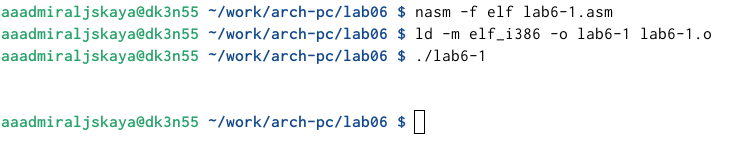


Рис. 7: Создание и запуск исполняемого файла

Затем создаем файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 (рис. 8).

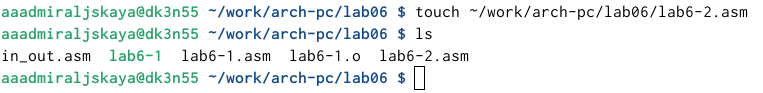


Рис. 8: Создание файла

Открываем созданный файл и вводим в него текст для вывода значения регистра eax (рис. 9).

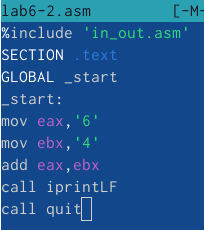


Рис. 9: Ввод текста программы в файл lab6-2.asm

Создаем и запускаем исполняемый файл lab6-2. Программа выводит число 106 (рис. 10).

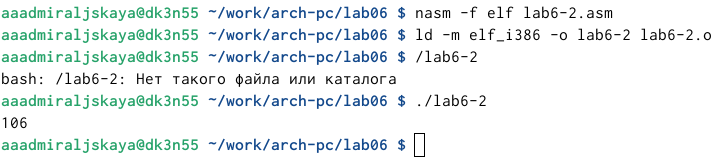


Рис. 10: Создание и запуск исполняемого файла

Так же, как и в lab6-1.asm, заменяем в тексте программы в файле lab6-2.asm символы “6” и “4” на числа 6 и 4 (рис. 11).

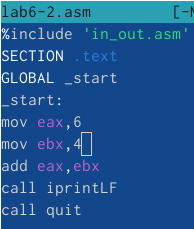


Рис. 11: Изменение текста программы

Создаем новый исполняемый файл программы и запускаем его. Теперь программа складывает не соответствующие символам коды в системе ASCII, а сами числа, поэтому вывод 10 (рис. 12).

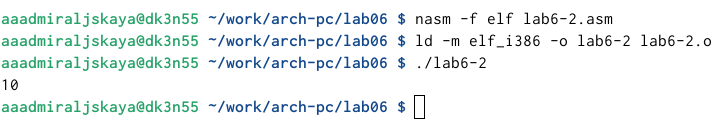


Рис. 12: Запуск исполняемого файла

Далее открываем файл lab6-2.asm и меняем функцию iprintLF на iprint (рис. 13).

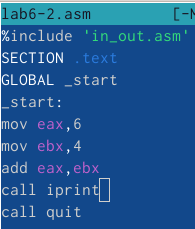


Рис. 13: Изменение текста программы

Создаем и запускаем исполняемый файл. Вывод не изменился, потому что символ переноса строки не отображался, когда программа исполнялась с функцией iprintLF, а iprint не добавляет к выводу символ переноса строки, в отличие от iprintLF (рис. 14).

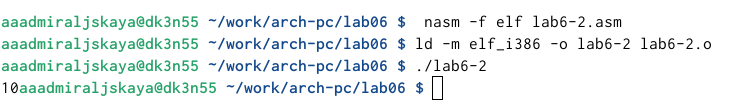


Рис. 14: Запуск измененного файла

Следующим шагом создаем файл lab6-3.asm в каталоге ~/work/arch-pc/lab06 (рис. 15).

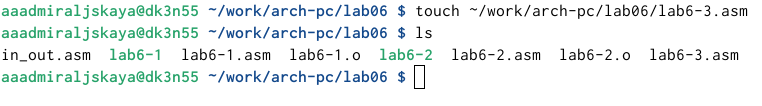


Рис. 15: Создание файла lab6-3.asm

Открываем созданный файл и вводим текст программы для вычисления значения выражения f(x) = (5 \* 2 + 3)/3 (рис. 16).

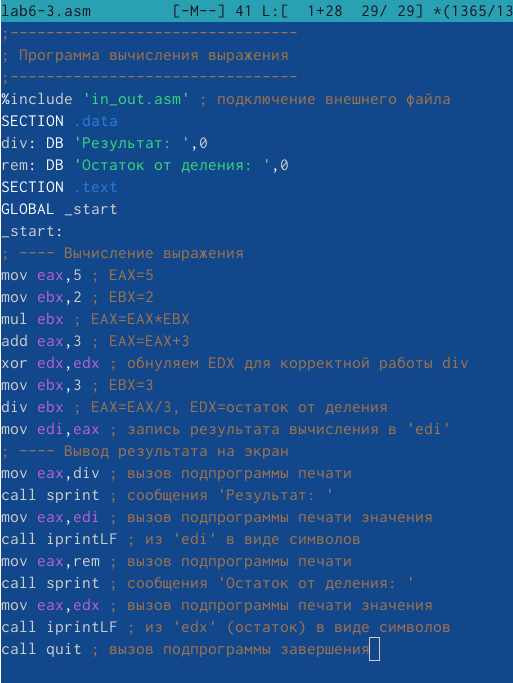


Рис. 16: Ввод текста программы в файл lab6-3.asm

Создаем исполняемый файл и запускаем его (рис. 17).

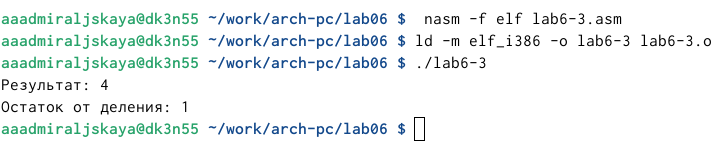


Рис. 17: Создание и запуск исполняемого файла

Затем в этом же файле изменяем текст программы для вычисления выражения 𝑓(𝑥) = (4 ∗ 6 + 2)/5 (рис. 18).

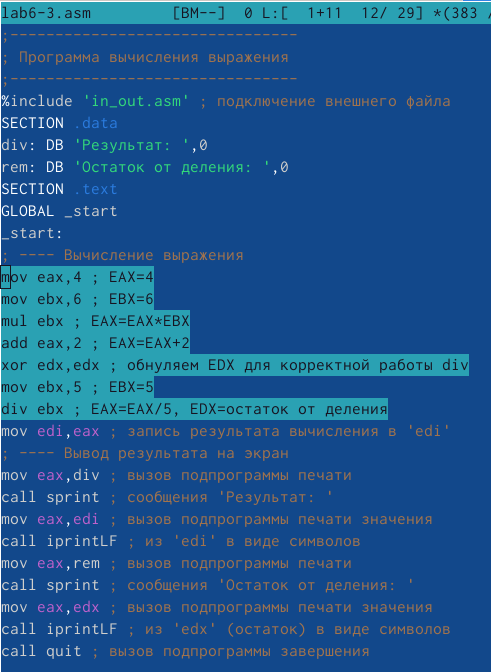


Рис. 18: Изменение текста программы

Создаем и запускаем исполняемый файл (рис. 19).

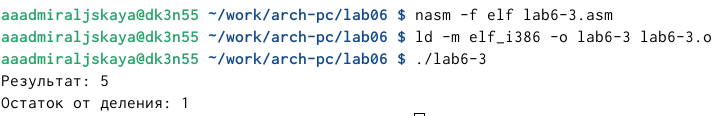


Рис. 19: Запуск исполняемого файла

Далее создаем файл variant.asm в каталоге ~/work/arch-pc/lab06 (рис. 20).

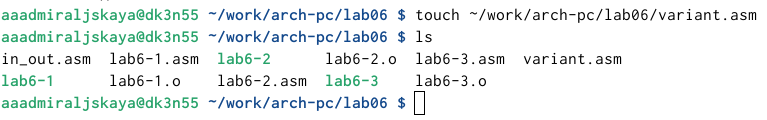


Рис. 20: Создание файла variant.asm

Вводим в этот файл текст программы для вычисления варианта задания по номеру студенческого билета (рис. 21).

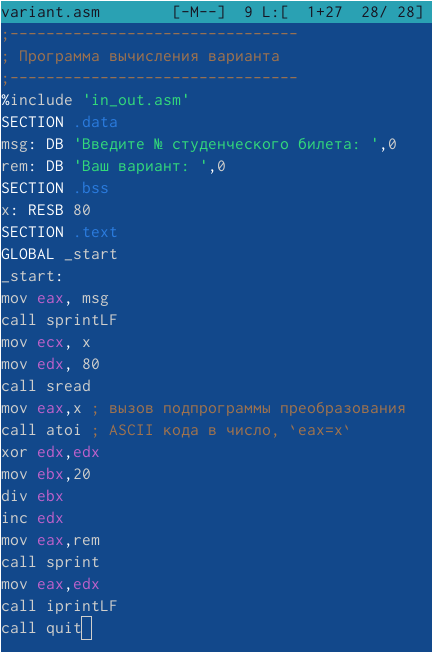


Рис. 21: Ввод текста программы в файл variant.asm

Создаем и запускаем исполняемый файл. Программа просит ввести номер студенческого билета (рис. 22).

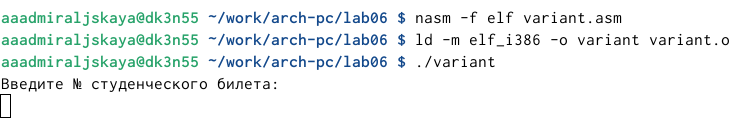


Рис. 22: Запуск исполняемого файла

Вводим номер своего студенческого билета с клавиатуры, программа вывела, что мой вариант - 7 (рис. 23).

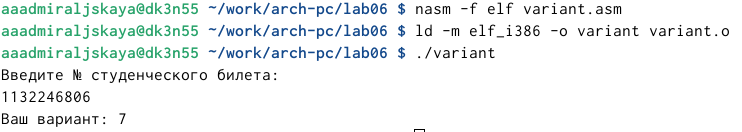


Рис. 23: Работа программы

Ответы на вопросы:

1)За вывод сообщения “Ваш вариант” отвечают строки кода:

mov eax,rem call sprint

2)Инструкция mov ecx, x используется, чтобы положить адрес вводимой строки x в регистр ecx mov edx, 80 - запись в регистр edx длины вводимой строки, call sread - вызов подпрограммы из внешнего файла, обеспечивающей ввод сообщения с клавиатуры

3)Call atoi используется для вызова подпрограммы из внешнего файла, которая преобразует ascii-код символа в целое число и записывает результат в регистр eax

4)За вычисления варианта отвечают строки:

xor edx,edx ; обнуление edx для корректной работы div mov ebx,20 ; ebx = 20 div ebx ; eax = eax/20, edx - остаток от деления inc edx ; edx = edx + 1

5)При выполнении инструкции div ebx остаток от деления записывается в регистр edx

6)Инструкция inc edx увеличивает значение регистра edx на 1

7)За вывод на экран результатов вычислений отвечают строки:

mov eax,edx call iprintLF

Приступим к выполнению заданий для самостоятельной работы. Создаем файл lab6-4.asm с помощью утилиты touch (рис. 24).

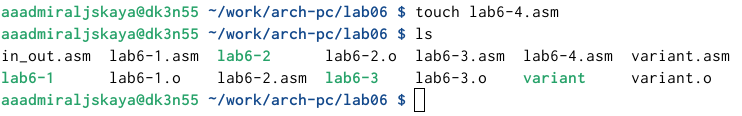


Рис. 24: Создание файла lab6-4.asm

Открываем созданный файл клавишей F4. Вводим в него текст программы для вычисления значения выражения 5(𝑥 − 1)^2. Оно было под вариантом 7 (рис. 25).

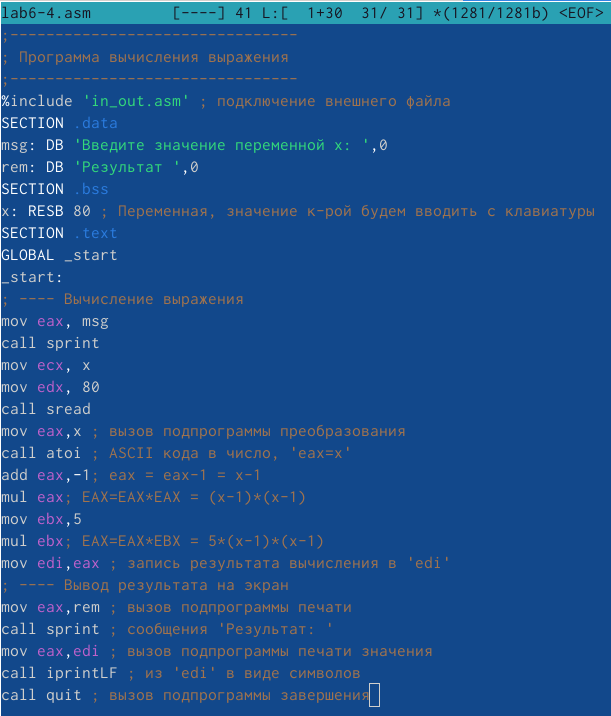


Рис. 25: Ввод текста программы в файл lab6-4.asm

Создаем и запускаем исполняемый файл. Проверяем его работу для значения 3 - на выводе 20 и для значения 5 - на выводе 80. Все верно (рис. 26).

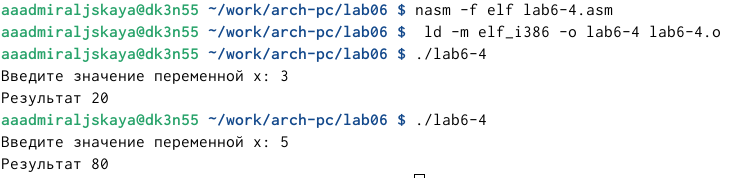


Рис. 26: Создание и запуск файла. Проверка его работы

Текст программы для вычисления значения выражения 5(𝑥 − 1)^2:

;——————————– ; Программа вычисления выражения ;——————————– %include ‘in\_out.asm’ ; подключение внешнего файла SECTION .data msg: DB ‘Введите значение переменной x:’,0 rem: DB ‘Результат’,0 SECTION .bss x: RESB 80 ; Переменная, значение к-рой будем вводить с клавиатуры SECTION .text GLOBAL \_start \_start: ; —- Вычисление выражения mov eax, msg call sprint mov ecx, x mov edx, 80 call sread mov eax,x ; вызов подпрограммы преобразования call atoi ; ASCII кода в число, ‘eax=x’ add eax,-1; eax = eax-1 = x-1 mul eax; EAX=EAX*EAX = (x-1)*(x-1) mov ebx,5 mul ebx; EAX=EAX*EBX = 5*(x-1)\*(x-1) mov edi,eax ; запись результата вычисления в ‘edi’ ; —- Вывод результата на экран mov eax,rem ; вызов подпрограммы печати call sprint ; сообщения ‘Результат:’ mov eax,edi ; вызов подпрограммы печати значения call iprintLF ; из ‘edi’ в виде символов call quit ; вызов подпрограммы завершения

# 5 Выводы

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

# Список литературы

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.

2. Robbins A. Bash Pocket Reference. O’Reilly Media, 2016. 156 с.

3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.

4. Newham C. [Learning the bash Shell: Unix Shell Programming](http://www.amazon.com/Learning-bash-Shell-Programming-Nutshell/dp/0596009658). O’Reilly Media, 2005. 354 с.