---

## 0.1 Front matter

title: “Отчёт по лабораторной работе №6” subtitle: “Простейший вариант” author: “Янушкевич Михаил Денисович”

## 0.2 Generic otions

lang: ru-RU toc-title: “Содержание”

## 0.3 Bibliography

bibliography: bib/cite.bib csl: pandoc/csl/gost-r-7-0-5-2008-numeric.csl

## 0.4 Pdf output format

toc: true # Table of contents toc-depth: 2 lof: true # List of figures lot: true # List of tables fontsize: 12pt linestretch: 1.5 papersize: a4 documentclass: scrreprt ## I18n polyglossia polyglossia-lang: name: russian options: - spelling=modern - babelshorthands=true polyglossia-otherlangs: name: english ## I18n babel babel-lang: russian babel-otherlangs: english ## Fonts mainfont: PT Serif romanfont: PT Serif sansfont: PT Sans monofont: PT Mono mainfontoptions: Ligatures=TeX romanfontoptions: Ligatures=TeX sansfontoptions: Ligatures=TeX,Scale=MatchLowercase monofontoptions: Scale=MatchLowercase,Scale=0.9 ## Biblatex biblatex: true biblio-style: “gost-numeric” biblatexoptions: - parentracker=true - backend=biber - hyperref=auto - language=auto - autolang=other\* - citestyle=gost-numeric ## Pandoc-crossref LaTeX customization figureTitle: “Рис.” tableTitle: “Таблица” listingTitle: “Листинг” lofTitle: “Список иллюстраций” lotTitle: “Список таблиц” lolTitle: “Листинги” ## Misc options indent: true header-includes: -

# keep figures where there are in the text

## # keep figures where there are in the text

# 1 Содержание

1. Цель работы
2. Выполнение лабораторной работы
3. Задание для самостоятельной работы
4. Вывод

# 2 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

# 3 Выполнение лабораторной работы

1. Создать каталог для программ лабораторной работы №6, в нём создать файл lab6-1.asm.(рис. [12](#fig:001)).

Figure 1: 1

Figure 1: 1

С помощью команды mkdir создаём каталог lab06, далее в нём, с помощью команды touch создаём файл lab6-1.asm.

1. В файл lab6-1.asm ввести текст программы из листинга 6.1.(рис. [2](#fig:002)).

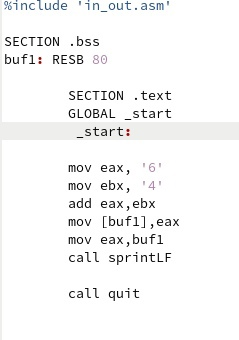


Figure 2: 2

Открываем файл lab6-1.asm в редакторе gedit, вводим необходимый текст из листинга 6.1.

1. Создать исполняемый файл и запустить его.((рис. [3](#fig:003)).

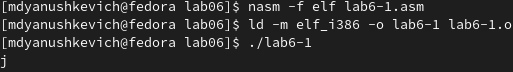


Figure 3: 3

В командной строке вводим необходимые команды, чтобы создать из текстового файла исполняемый, далее запускаем его. Результатом выполения программы становится символ “j”.

1. Изменить текст программы, заменяя символы регистрами чисел.(рис. [4](#fig:004)).(рис. [5](#fig:005)).

Figure 4: 4

Figure 4: 4

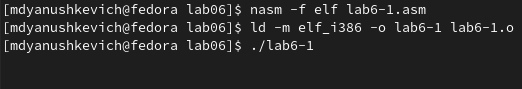


Figure 5: 5

Текст из листинга 6.1 изменяем, заменяя необходимые строки. Далее создаём исполняемый файл,запускаем файл. Результат работы программы не выводится на экран. Опираясь на таблицу ASCII, можно понять, что код 10 соответсвует символу “пробел”, который и является результатом выполнения программы.

1. Создать файл lab6-2.asm.и ввести в него текст программы из листинга 6.2.(рис. [6](#fig:006)).(рис. [7](#fig:007)).

Figure 6: 6

Figure 6: 6

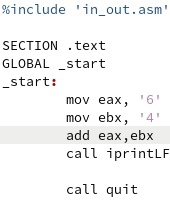


Figure 7: 7

В каталоге lab06 создаём файл lab6-2.asm и вводим в него текст программы из листинга 6.2.

1. Создать исполняемый файл и запустить его.(рис. [8](#fig:008)).

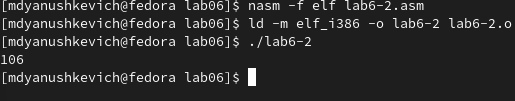


Figure 8: 8

В командной строке вводим необходимые команды, чтобы создать исполняемый файл. Далее запускаем его. Результатом работы программы будет число 106.

1. Заменить необходимые строки в тексте листинга 6.2. Создать исполняемый файл и запустить его.(рис. [9](#fig:009)).(рис. [10](#fig:010)).

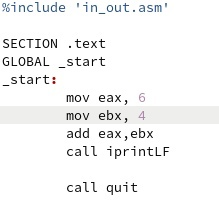


Figure 9: 9

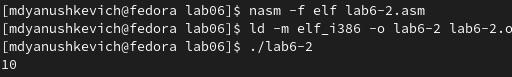


Figure 10: 10

В тексте файла lab6-2.asm заменяем необходимые строки. Далее создаём исполняемый файл, запускаем его. Как можно увидеть, результатом работы программы будет число 10.

1. В тексте файла lab6-2.asm заменить функцию iprintLF на iprint.(рис. ??).(рис. ??).

Figure 11: 11

Figure 11: 11

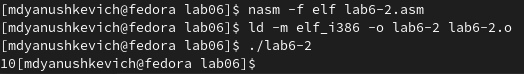


Figure 12: 12

В файле lab6-2.asm заменяем функцию iprintLF на iprint. Создаём исполняемый файл и запускаем программу. Результатом выполнения программы будет число 10. В отличие от вывода функции iprintLF, iprint выводит результат в одну строчку.

1. Создать файл lab6-3.asm в каталоге lab06. В файл ввести текст программы из листинга 6.3. Создать исполняемый файл и запустить его.(рис. [13](#fig:013)).(рис. [14](#fig:014)).(рис. [15](#fig:015)).

Figure 13: 13

Figure 13: 13

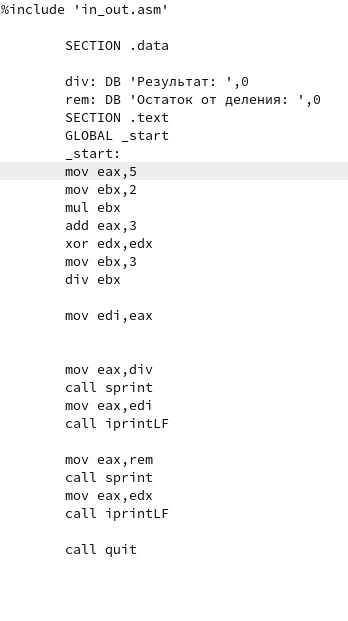


Figure 14: 14

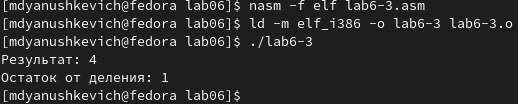


Figure 15: 15

С помощью команды touch создаём файл lab6-3.asm. Открываем его и вводим текст из листинга 6.3. Создаём исполняемый файл и запускаем программу. Результатом работы программы будет число 4, а остаток будет равняться 1.

1. Изменить текст программы из файла lab6-3.asm для вычисления выражения 𝑓(𝑥) = (4 ∗ 6 + 2)/5. (рис. [16](#fig:016)).(рис. [17](#fig:017)).

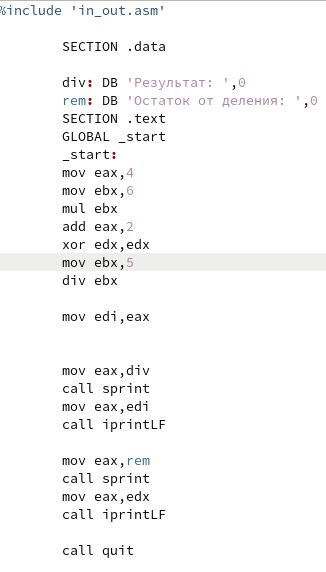


Figure 16: 16

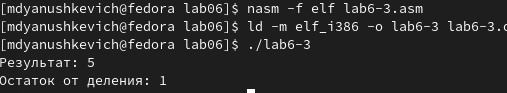


Figure 17: 17

В файле lab6-3.asm вносим необходимые изменения, чтобы вычислить выражение 𝑓(𝑥) = (4 ∗ 6 + 2)/5. Далее создаём исполняемый файл, проверяем его работу. Результатом исполнения программы будет число 5, а в остатке будет 1.

1. Создать файл variant.asm в каталоге lab06. В этот файл ввести текст программы из листинга 6.4. Создать исполняемый файл программы и запустить его.(рис. [18](#fig:018)).(рис. [19](#fig:019)).(рис. [20](#fig:020)).

Figure 18: 18

Figure 18: 18

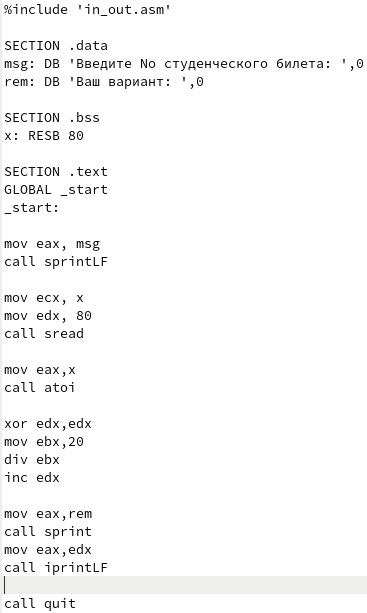


Figure 19: 19

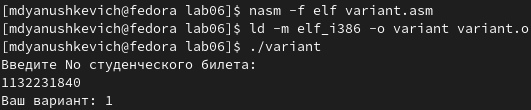


Figure 20: 20

С помощью команды touch создаём файл variant.asm в каталоге lab06. Открываем этот файл и вводим в него текст из листинга 6.4. Эта программа вычисляет вариант задания по номеру студенческого билета. Далее запускам программу, вводим номер ст.билета(мой номер-1132231840). В результате выводится число 1, что означает, что мне достался вариант №1.

# 4 Ответы на вопросы

1. Строки, отвечающие за вывод на экран сообщения “Ваш вариант:” mov eax,rem, call sprint.
2. Эти конструкции используются для ввода сообщения с клавиатуры. В eax записывается адрес переменной, в которую сообщение будет записано, в edx указывается длина вводимой строки.
3. Call atoi используется для преобразования ASCII чисел в целые числа.
4. mov eax,x - Строки, отвечающие за вычисление варианта. call atoi xor edx,edx mov ebx,20 div ebx inc edx
5. Остаток записывается в регистр edx.
6. Эта функция считывает значение eax, прибавляет к нему 1, а потом перезаписывает обратно в eax.
7. mov eax,edx - Строки, отвечающие за вывод результата вычислений. call iprintLF

# 5 Задание для самостоятельной работы

1. Создать файл в формате asm. В нём написать программу вычисления выражения y=f(x). Вид функции соответствует номеру варианта из предыдущего задания лабораторной работы.(рис. [21](#fig:021)).(рис. [22](#fig:022)).

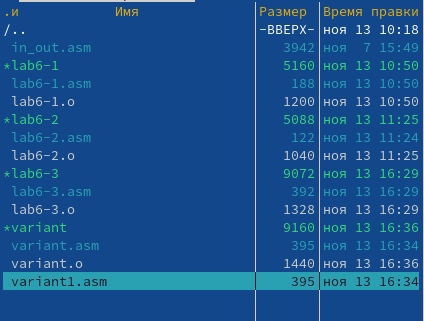


Figure 21: 21

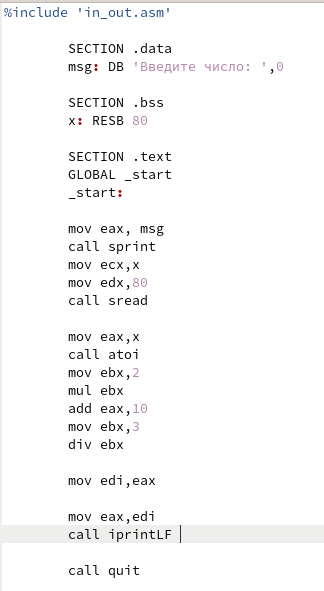


Figure 22: 22

Создаём файл variant1.asm, с помощью midnight commander убеждаемся в его создании. Далее открываем его и на основе полученных ранее знаниях пишем программу, которая будет вычислять выражение y=(10+2x)/3, где х1=1, х2=10.

1. Запустить написанную программу и проверить её работу.(рис. [23](#fig:023)).

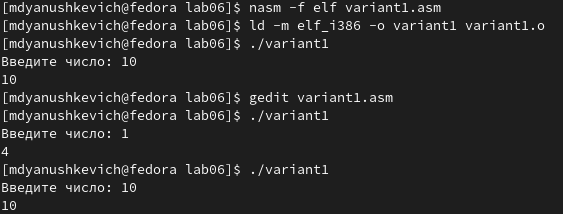


Figure 23: 23

После написания программы создаём объектный файл, запускаем программу. Для этого поочерёдно вводим в строку значения х. Ответами являются числа 4 и 10, из чего можно сделать вывод, что программа написана верно.

# 6 Выводы

Благодаря этой лабораторной работе я освоил арифметические инструкции языка ассемблера NASM, что позволило мне написать полноценную программу по вычислению выражения типа y=f(x). Также я закрепил полученные ранее навыки по работе с командной строкой ОС Linux и языком ассемблера NASM.