Шаблон отчёта по лабораторной работе 06

Простейший вариант

Абдуллахи шугофа

Содержание

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

# 2 Порядок выполнения лабораторной работы

No 1: Символьные и численные данные в NASM

1.1: Создайте каталог для программам лабораторной работы No 6, перейдите в него и создайте файл lab6-1.asm:

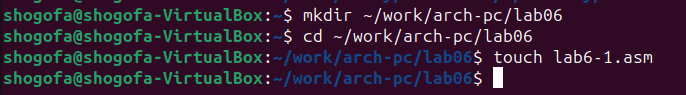


Figure 1: Создала файл



Figure 2:

1.2: Листинг 6.1. Программа вывода значения регистра eax

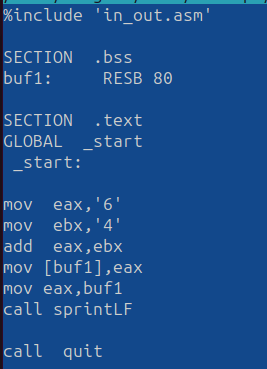


Figure 3: вывода значения регистра eax

Создайте исполняемый файл и запустите его.

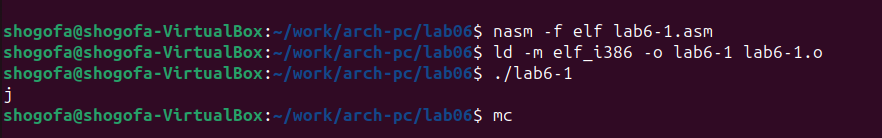


Figure 4: Создайла исполняемый файл и запустила его.

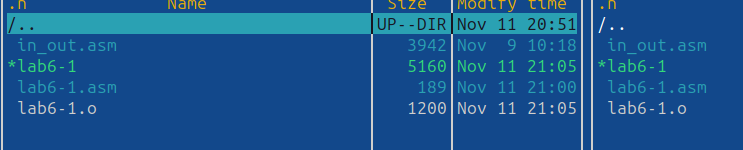


Figure 5:

1.3: Далее изменим текст программы и вместо символов, запишем в регистры числа. Ис- правьте текст программы (Листинг 6.1) следующим образом: замените строки

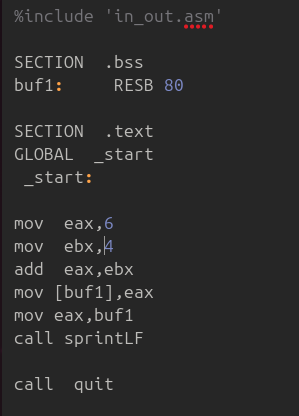


Figure 6: изменила текст программы

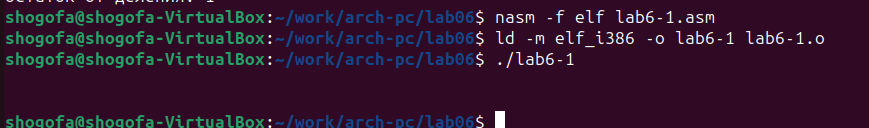


Figure 7: резултат изменила на символ

1.4: Как отмечалось выше, для работы с числами в файле in\_out.asm реализованы подпро- граммы для преобразования ASCII символов в числа и обратно. Преобразуем текст программы из Листинга 6.1 с использованием этих функций.

Создайте файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и введите в него текст про- граммы из листинга 6.2.

Figure 8: Создайла файл lab6-2.asm

Figure 8: Создайла файл lab6-2.asm

1. Листинг 6.2. Программа вывода значения регистра eax

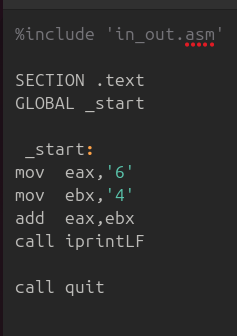


Figure 9: вывода значения регистра eax

Создайте исполняемый файл и запустите его

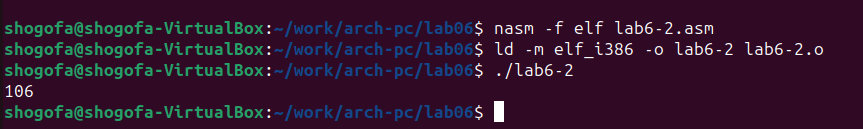


Figure 10: Создайла исполняемый файл и запустила его

2.1: Аналогично предыдущему примеру изменим символы на числа. Замените строки

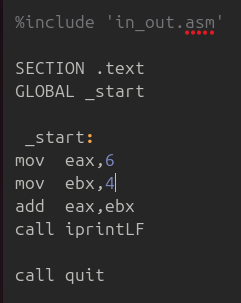


Figure 11: изменила символы на числа. Замените строки

2.2: Создайте исполняемый файл и запустите его. Какой результат будет получен при исполне- нии программы?

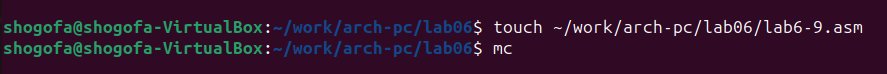


Figure 12: Создайла исполняемый файл

Замените функцию iprintLF на iprint. Создайте исполняемый файл и запустите его. Чем отличается вывод функций iprintLF и iprint?

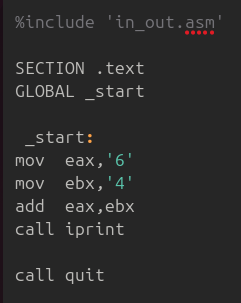


Figure 13: Заменила функцию iprintLF на iprint

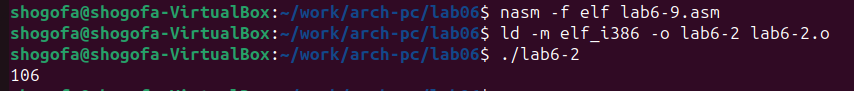


Figure 14: Создайла исполняемый файл и запустила его

No 2: Выполнение арифметических операций в NASM

2.1: В качестве примера выполнения арифметических операций в NASM приведем про- грамму вычисления арифметического выражения 𝑓(𝑥) = (5 ∗ 2 + 3)/3.

Создайте файл lab6-3.asm в каталоге ~/work/arch-pc/lab06: 2.6 Задание 6:

Figure 15: Создайте файл

Figure 15: Создайте файл

Внимательно изучите текст программы из листинга 6.3 и введите в lab6-3.asm.

1. Листинг 6.3. Программа вычисления выражения 𝑓(𝑥) = (5 ∗ 2 + 3)/3

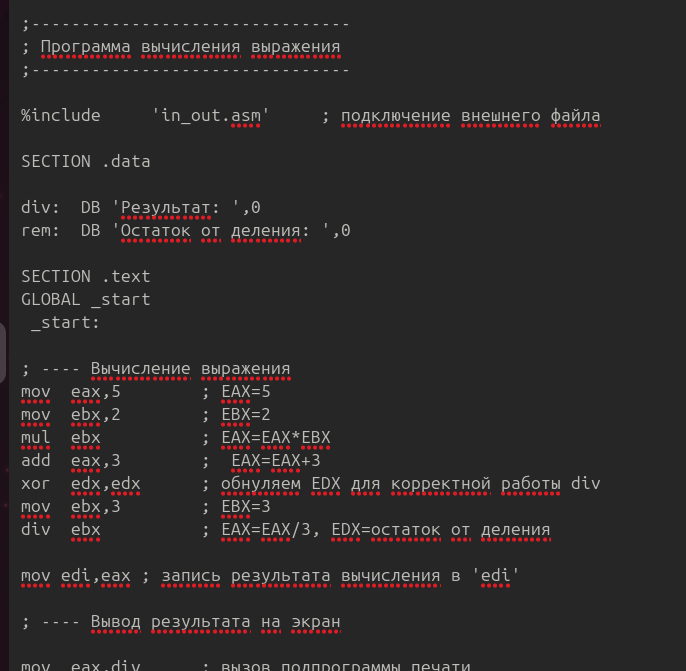


Figure 16: Программа вычисления выражени

Создайте исполняемый файл и запустите его. Результат работы программы должен быть следующим:

user@dk4n31:~$ ./lab6-3 Результат: 4 Остаток от деления: 1 user@dk4n31:~$

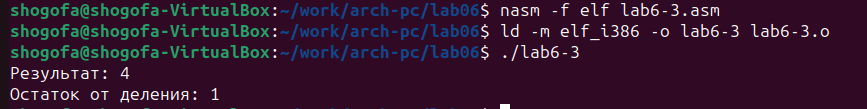


Figure 17: Создайла исполняемый файл и запустила его

Измените текст программы для вычисления выражения 𝑓(𝑥) = (4 ∗ 6 + 2)/5. Создайте исполняемый файл и проверьте его работу.

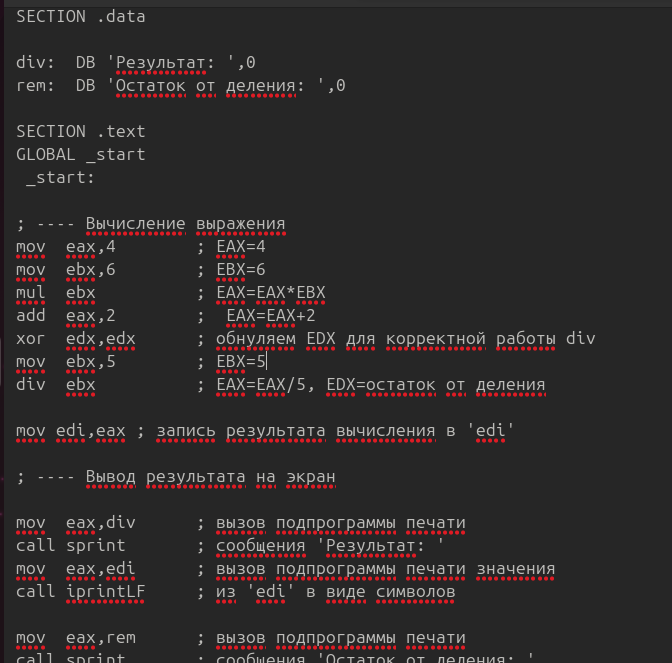


Figure 18: Изменила текст программы

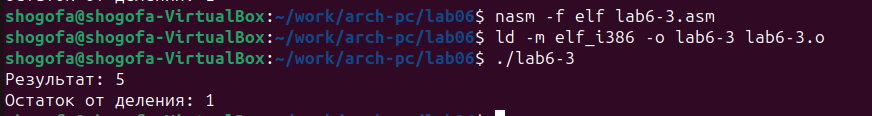


Figure 19: Создайла исполняемый файл и проверьла его работу

3.1: Создайте файл variant.asm в каталоге ~/work/arch-pc/lab06:

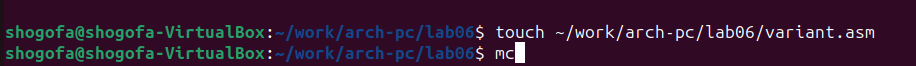


Figure 20: Создайла файл variant.asm

Figure 21: 

Figure 21:

Внимательно изучите текст программы из листинга 6.4 и введите в файл variant.asm.

1. Листинг 6.4. Программа вычисления вычисления варианта задания по номеру студенческого билета

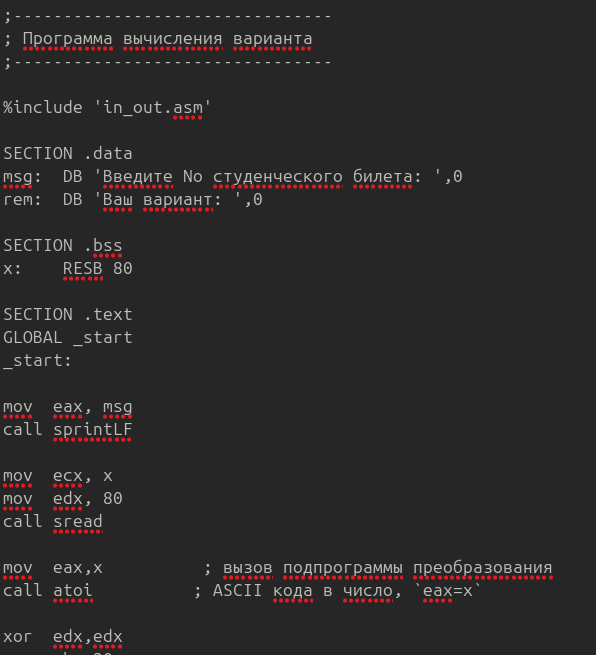


Figure 22: введила в файл текст

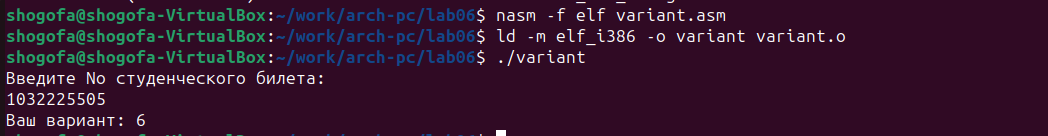


Figure 23: Проверьла результат работы программы

Создайте исполняемый файл и запустите его. Проверьте результат работы программы вычислив номер варианта аналитически.

Включите в отчет по выполнению лабораторной работы ответы на следующие вопросы:

1. Какие строки листинга 6.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’?

Строка mov eax, rem и строка call sprint отвечают за вывод на экран сообщения “мой вариант:”.

1. Для чего используется следующие инструкции? mov ecx, x mov edx, 80 call sread

Инструкции mov ecx, x и mov edx, 80 используются для подготовки аргументов перед вызовом подпрограммы sread. mov ecx, x загружает адрес переменной x в регистр ecx, который будет использован в качестве аргумента для функции sread. mov edx, 80 загружает значение 80 в регистр edx, указывая функции sread, сколько байт нужно прочитать.

1. Для чего используется инструкция “call atoi”?

Инструкция call atoi используется для вызова подпрограммы atoi, которая преобразует ASCII-код, хранящийся в регистре eax, в число. Результат преобразования сохраняется в регистре eax.

1. Какие строки листинга 6.4 отвечают за вычисления варианта?

Строка xor edx, edx и строка mov ebx, 20 отвечают за подготовку значений перед выполнением вычислений варианта. xor edx, edx устанавливает регистр edx в ноль, а mov ebx, 20 загружает значение 20 в регистр ebx, которое будет использовано для деления.

1. В какой регистр записывается остаток от деления при выполнении инструкции “div ebx”?

Остаток от деления при выполнении инструкции div ebx записывается в регистр edx.

1. Для чего используется инструкция “inc edx”?

Инструкция inc edx используется для увеличения значения в регистре edx на единицу. В данном случае, она увеличивает значение остатка от деления на 1.

1. Какие строки листинга 6.4 отвечают за вывод на экран результата вычислений?

Строка mov eax, edx и строка call iprintLF отвечают за вывод на экран результата вычислений. mov eax, edx загружает значение в регистр eax, чтобы передать его в функцию iprintLF, которая выводит значение на экран с новой строкой.

1. Задание для самостоятельной работы

Написать программу вычисления выражения 𝑦 = 𝑓(𝑥). Программа должна выводить выражение для вычисления, выводить запрос на ввод значения 𝑥, вычислять задан- ное выражение в зависимости от введенного 𝑥, выводить результат вычислений. Вид функции 𝑓(𝑥) выбрать из таблицы 6.3 вариантов заданий в соответствии с номером полученным при выполнении лабораторной работы. Создайте исполняемый файл и проверьте его работу для значений 𝑥1 и 𝑥2 из 6.3

Figure 24: мой вариант

Figure 24: мой вариант

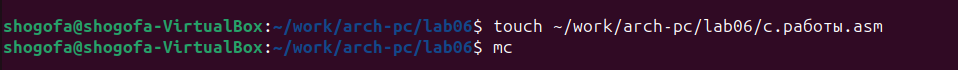


Figure 25: Создайла файл

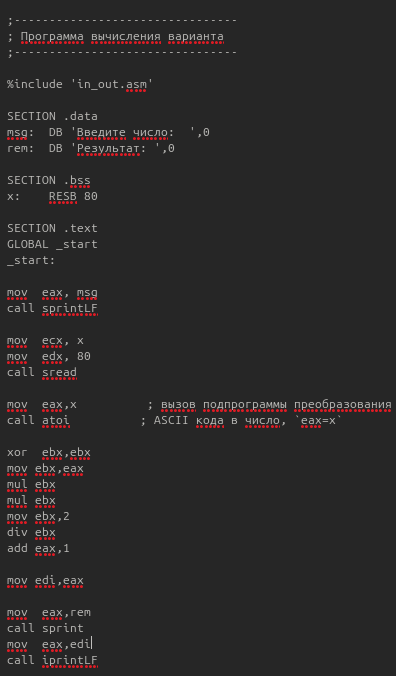


Figure 26:

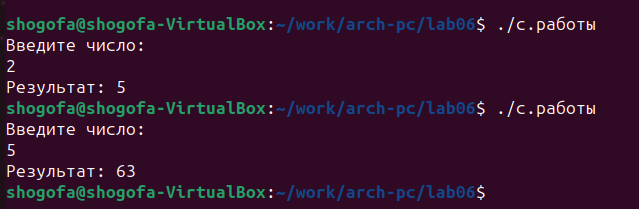


Figure 27: результат

# 3 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы.

Например, в табл. [1](#tbl:std-dir) приведено краткое описание стандартных каталогов Unix.

Table 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 об Unix см. в [1–6].

Выполнение лабораторной работы

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. [28](#fig:001)).



Figure 28: Название рисунка

# 4 Выводы

Здесь кратко описываются итоги проделанной работы.

# Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016. URL: <https://www.gnu.org/software/bash/manual/>.

2. 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.

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

4. Robbins A. [Bash Pocket Reference](https://www.ncbi.nlm.nih.gov/pubmed/25246403). O’Reilly Media, 2016. 156 с.

5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.

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