Отчёт по лабораторной работе

Простейший вариант

Тимур Ринатович Каримов

Содержание

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM

# 2 Задание

1. Выполнение лабораторной работы
2. Ответы на вопросы
3. Выполнение 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

Создаем каталог для программ лабораторной №6. Переходим в него и создаем файл lab6-1.asm (рис. 1)



Рис. 1: Создание каталога и файла

Ввожу в созданный файл текст программы из листинга (рис. 2)

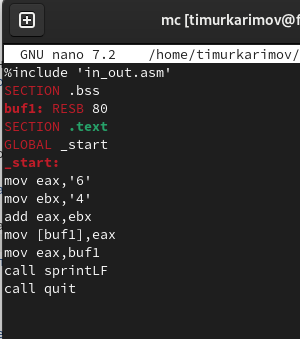


Рис. 2: Написание программы

Создаем исполняемый файл и запускаем его(рис. 3)

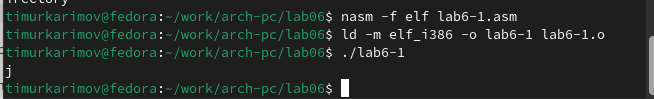


Рис. 3: Выполнение программы

Изменяем текст программы, вместо символов записываем в eax, ebx числа (рис. 4)

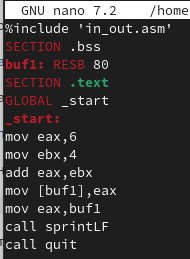


Рис. 4: Изменение программы

Создаем исполняемый файл и запускаем его (рис. 5)

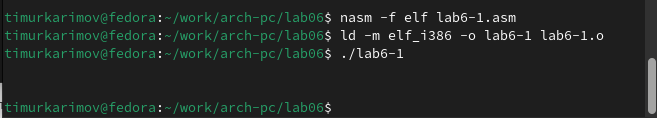


Рис. 5: Выполнение программы

На экране две пустые строки. Это связано с тем, что символ с кодом 10

- это символ перевода строки.

Создаем файл lab6-2.asm в каталоге для программ лабораторной №6. Ввожу в него текст программы из листинга 6.2 (рис. 6)

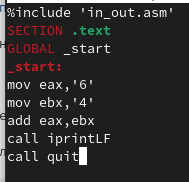


Рис. 6: Создание файла и написание программы

Создаем исполняемый файл и запускаем его (рис. 7)

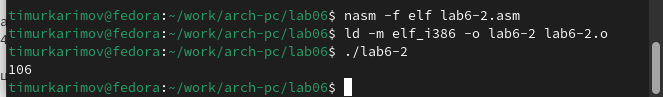


Рис. 7: Выполнение программы

Аналогично предыдущей программе заменяем символы на числа (рис. 8)

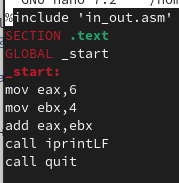


Рис. 8: Изменение программы

Создаем исполняемый файл и запускаем его (рис. 9)

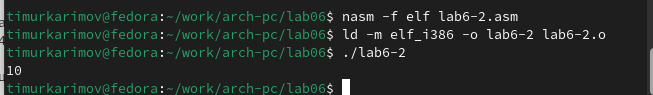


Рис. 9: Выполнение программы

Теперь программа складывает не коды, соответствующие символам, а сами числа. Поэтому выводит число 10 - сумму чисел 4 и 6.

Заменяем функцию iprintLF на iprint (рис. 10)

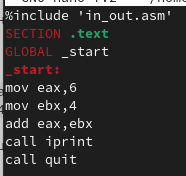


Рис. 10: Изменение программы

Создаем исполняемый файл и запускаем его(рис. 11)

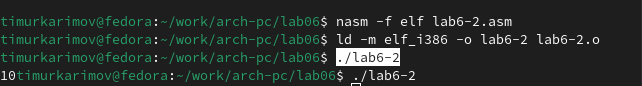


Рис. 11: Выполнение программы

Вывод функции iprintLF от вывода функции iprint отличается тем, что в последнем случае после вывода не добавляется переход на новую строку.

1. **Выполнение арифметических операций в Nasm**
2. Создаем файл lab6-3.asm. Вводим в него текст программы для вычисления значения указанного выражения (рис. 12)

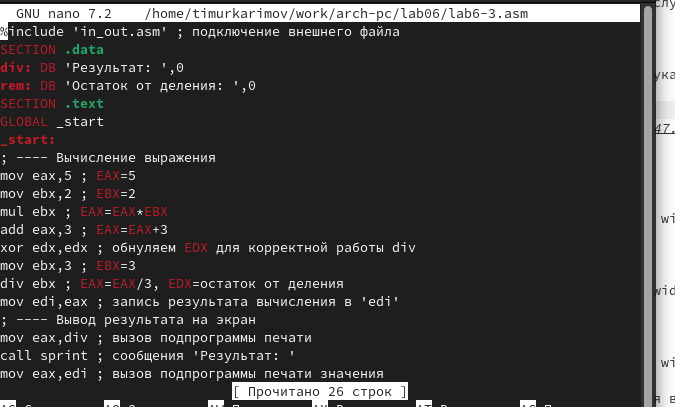


Рис. 12: Создание файла и написание программы

1. Создаем исполняемый файл и запускаем его (рис. 13)

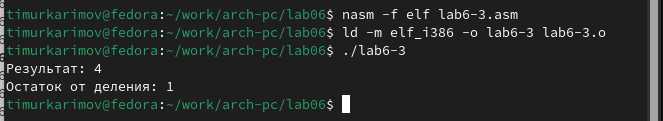


Рис. 13: Выполнение программы

Изменяем текст программы для вычисления нового выражения (рис. 14)

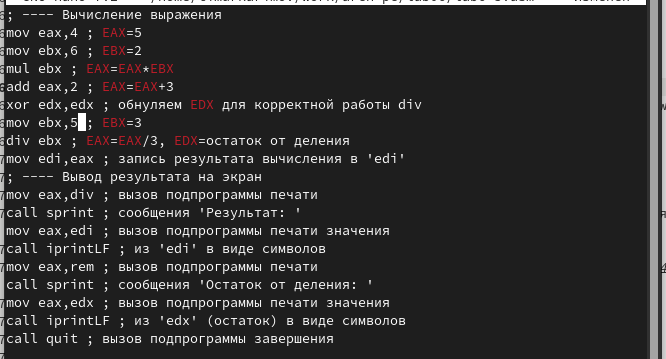


Рис. 14: Изменение программы

Создаем исполняемый файл и запускаем его (рис. 15)

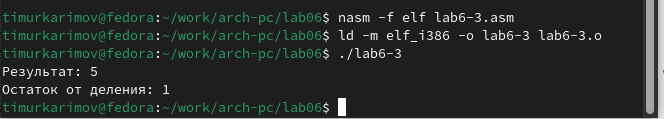


Рис. 15: Выполнение программы

С помощью утилиты touch создаем файл variant.asm. Вводим в него текст программы для вычисления варианта (рис. 16)

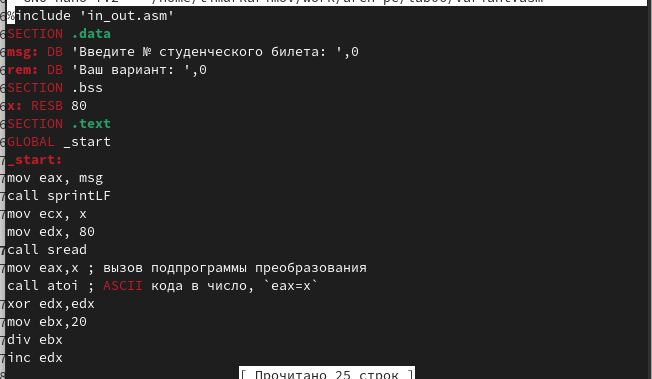


Рис. 16: Создание файла и написание программы

Создаем исполняемый файл и запускаем его (рис. 17)

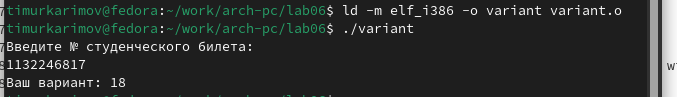


Рис. 17: Выполнение программы

Ответы на вопросы по программе: 1. За вывод сообщения “Ваш вариант” отвечают

строки кода:

mov eax,rem

call sprint

1. Инструкция mov ecx, x выполняется для того чтобы положить адрес вводимой строки x в регистр ecx. Инструкция mov edx, 80 выполняется для записи длины вводимой строки в регистр edx. Инструкция call sread выполняется для вызова подпрограммы из внешнего файла, обеспечивающей ввод сообщения с клавиатуры.
2. Инструкция call atoi используется для вызова подпрограммы из внешнего файла, которая преобразует ascii-код символа в целое число и записывает результат в регистр eax.
3. За вычисление варианта отвечают строки: *xor edx, edx*

*mov ebx, 20*

*div ebx*

*inc edx*

1. При выполнении инструкции div ebx остаток от деления записывается в регистр edx.
2. Инструкция inc edx увеличивает значение регистра edx на 1.
3. За вывод на экран результата вычислений отвечаю следующие строки: *mov eax, edx* *call iprintLF* #Выполнение самостоятельной работы

Создаем файл lab6-4.asm и записываем в него программу для вычисления выражения 18 варианта. (рис. 18)

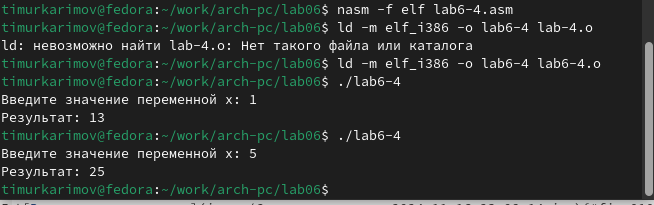


Рис. 18: Создание файла и написание программы

Создаем исполняемый файл и проверяем его работу для указанных значений (рис. 19)

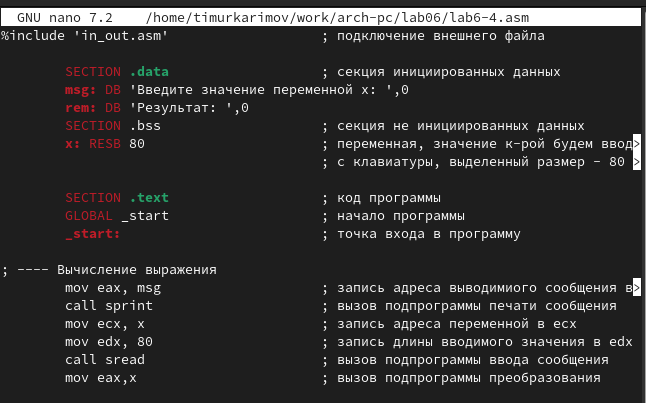


Рис. 19: Выполнение программы

# 5 Выводы

При выполении лабораторной работы помогла освоению арифметических инструкций 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.