Лабораторная работа №6

Архитектура вычислительных систем

Савурская Полина Александровна

Содержание

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

# 2 Задание

Написать программу вычисления выражения y=f(x). Программа должна выводить выражение для вычисления, выводить запрос на ввод значения x, вычислять заданное 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

# 3 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы.

Например, в табл. 1 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].

# 4 Выполнение лабораторной работы

1. Создаем каталог для программ лабораторной работы No 6, переходим в него и создадим файл lab6-1.asm (рис. 1)



Рис. 1: Создание каталога для программ

1. Введем в файл текст программы из листнинга, с помощью команды mc (рис. 2)

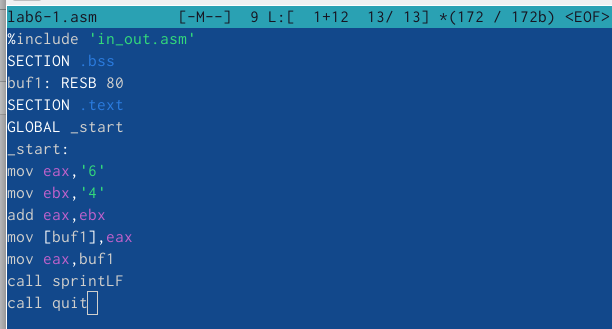


Рис. 2: Ввод текста программы

1. Создадим исполняемый файл и запустим его (рис. 3)

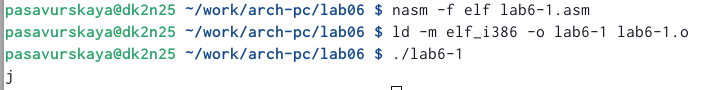


Рис. 3: Запуск исполняемого файла

1. Изменим текст программы (нужно убрать кавычки) (рис. 4)

Рис. 4: Изменение текста программы

Рис. 4: Изменение текста программы

1. Запускаем файл lab6-1. Как и в предыдущем случае, при исполнее программы мы не получили число 10. Символ под номером 10 в таблице ASCII выглядит так. Он не отобразился на экране. (рис. 5) (рис. 6)

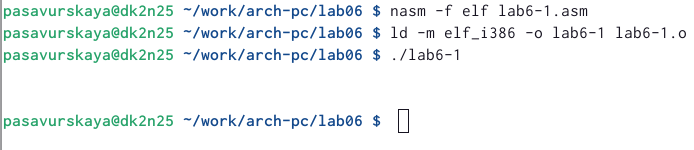


Рис. 5: Запуск файла lab6-1

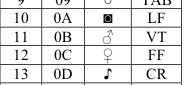


Рис. 6: Символ 10

1. Создаем файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и вводим в него текст программы из листинга 7.2. (рис. 7) (рис. 8)

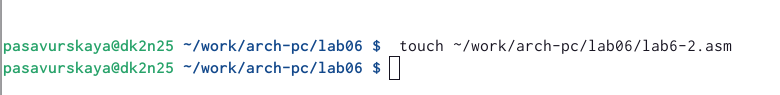


Рис. 7: Создание файла lab6-2

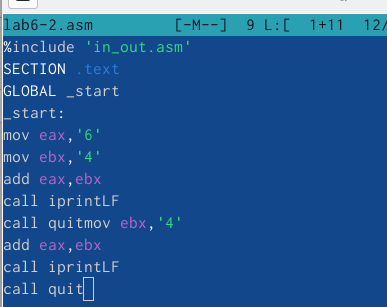


Рис. 8: Ввод текста в файл

1. Создадим исполняемый файл и запустим его (рис. 9)

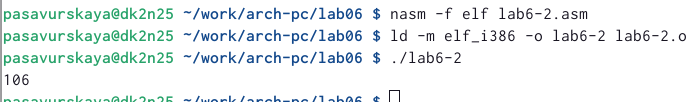


Рис. 9: Запуск исполняемого файла

1. Снова изменим текст программы (нужно убрать кавычки) (рис. 10)

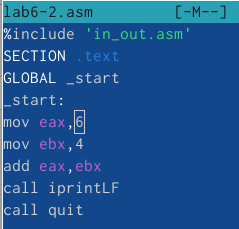


Рис. 10: Изменение текста программы

1. Создадим исполняемый файл и запустим его. В результате на экран выводится 10 (рис. 11)

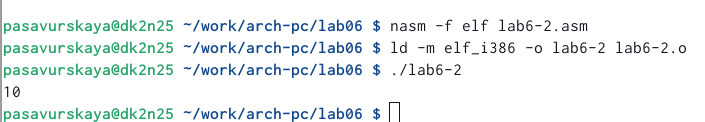


Рис. 11: Запуск исполняемого файла

1. Заменим функцию iprintLF на iprint. Создаем исполняемый файл и запустим его. Отличие в том, что LF отвечает за пропуск строки. Между строками нет разделения (рис. 12) (рис. 13)

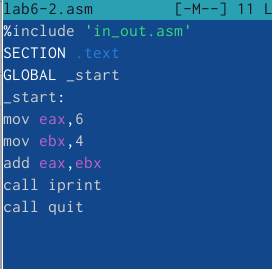


Рис. 12: Меняем текст

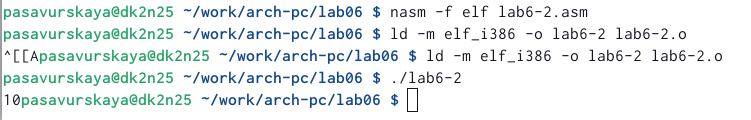


Рис. 13: Запускаем измененный файл

1. Создаем файл lab6-3.asm в каталоге ~/work/arch-pc/lab06 (рис. 14)



Рис. 14: Создание файла lab6-3

1. Вводим в файл lab6-3.asm текст из листинга 3 (рис. 15)

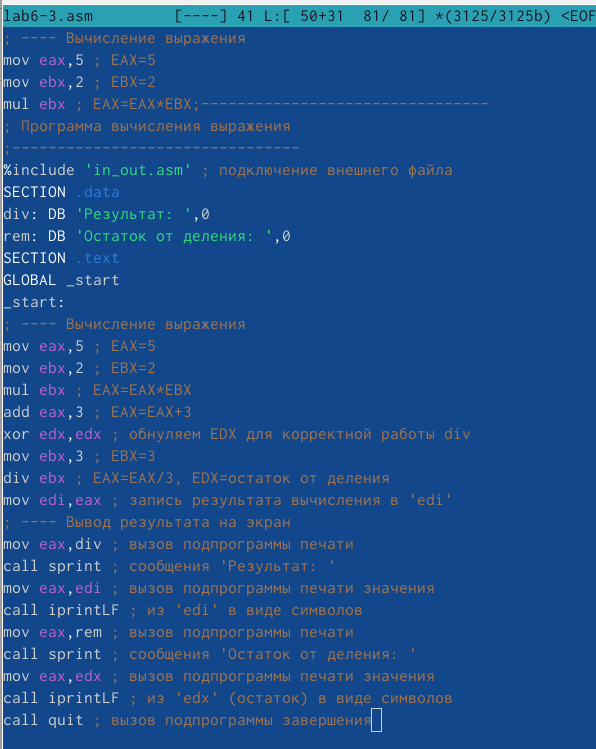


Рис. 15: Вводим текст в файл

1. Запускаем файл lab6-3.asm (рис. 16)

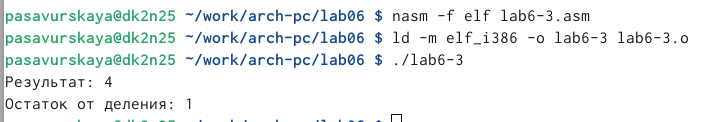


Рис. 16: Запуск файла

1. Изменим текст программы чтобы получить другой результат. Проверим правильность нашей работы путем запуска измененного файла (рис. 17)

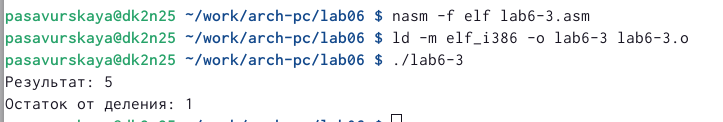


Рис. 17: Запуск файла

1. Создадим файл variant.asm в каталоге ~/work/arch-pc/lab06 (рис. 18)

Рис. 18: Создание файла variant.asm

Рис. 18: Создание файла variant.asm

1. Вводим в файл текст из листинга 4 и запускаем файл (рис. 19)

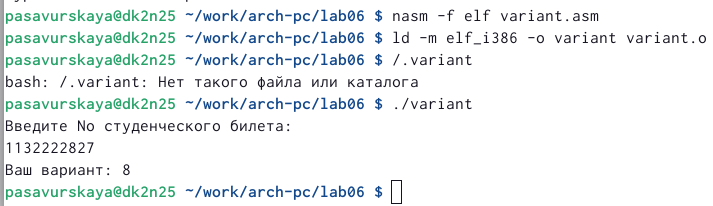


Рис. 19: Запуск файла

#Ответы на вопросы

1. Какие строки листинга 7.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’? Строчка rem: DB ‘Ваш вариант:’,0 отвечает за вывод сообщения.
2. Для чего используется следующие инструкции? nasm mov ecx, x mov edx, 80 call sread
3. Для чего используется инструкция “call atoi”? Функция call atoi преобразует ascii-код символа в целое число и записывает результат в регистр eax.
4. Какие строки листинга 7.4 отвечают за вычисления варианта? mov eax, msg call sprintLF
5. В какой регистр записывается остаток от деления при выполнении инструкции “div ebx”? Остаток записывается в регистр ah.
6. Для чего используется инструкция “inc edx”? Команда inc edx увеличивает значение регистра edx на 1.
7. Какие строки листинга 7.4 отвечают за вывод на экран результата вычислений? mov eax,rem call sprint mov eax,edx call iprintLF call quit

# 5 Выводы

Я освоила арифметические инструкции языка ассемблера NASM.

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.