Отчёт по лабораторной работе №4

Дисциплина: архитектура компьютера

Ларина Наталья Денисовна

Содержание

# 1 Цель работы

Целью данной лабораторной работы является освоение процедуры и сборки программ, написанных на ассемблере NASM.

# 2 Задание

1. Создание программы Hello, world!
2. Работа с траснслятором NASM
3. Работа с расширенным синтаксисом командой строки NASM
4. Работа с компоновщиками LD
5. Запуск исполняемого файла
6. Выполнение заданий для самостоятельной работы

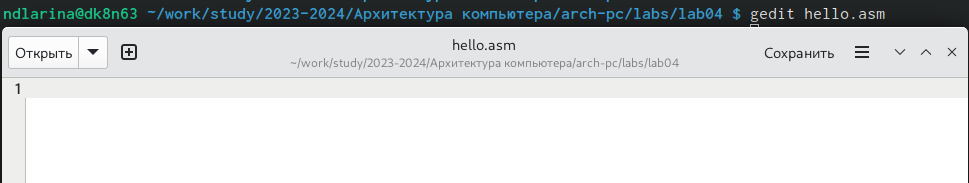
# 3 Теоретическое введение

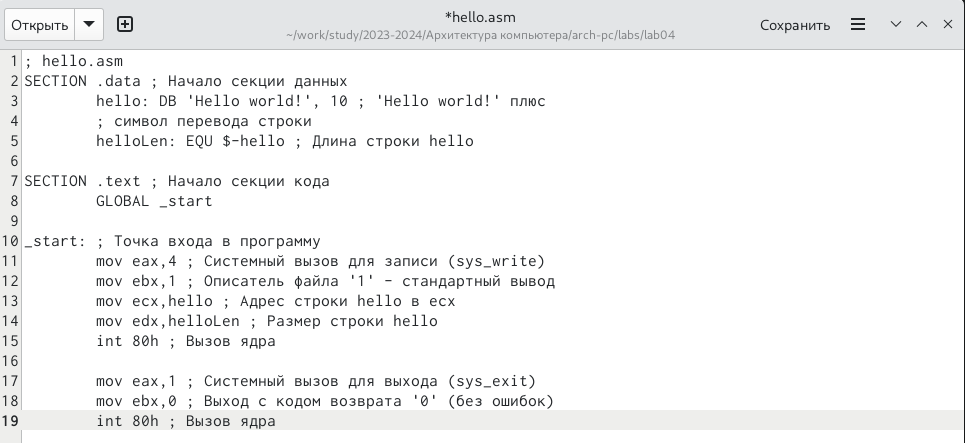
Основные принципы работы компьютера Основными функциональными элементами любой электронно-вычислительной машины (ЭВМ) являются центральный процессор, память и периферийные устройства (рис. 4.1). Взаимодействие этих устройств осуществляется через общую шину, к которой они подклю- чены. Физически шина представляет собой большое количество проводников, соединяющих устройства друг с другом. В современных компьютерах проводники выполнены в виде элек- тропроводящих дорожек на материнской (системной) плате. Основной задачей процессора является обработка информации, а также организация координации всех узлов компьютера. В состав центрального процессора (ЦП) входят следующие устройства: • арифметико-логическое устройство (АЛУ) — выполняет логические и арифметиче- ские действия, необходимые для обработки информации, хранящейся в памяти; • устройство управления (УУ) — обеспечивает управление и контроль всех устройств компьютера; • регистры — сверхбыстрая оперативная память небольшого объёма, входящая в со- став процессора, для временного хранения промежуточных результатов выполнения инструкций; регистры процессора делятся на два типа: регистры общего назначения и специальные регистры. Для того, чтобы писать программы на ассемблере, необходимо знать, какие регистры процессора существуют и как их можно использовать. Большинство команд в программах написанных на ассемблере используют регистры в качестве операндов. Практически все команды представляют собой преобразование данных хранящихся в регистрах процессора, это например пересылка данных между регистрами или между регистрами и памятью, пре- образование (арифметические или логические операции) данных хранящихся в регистрах.

# 4 Выполнение лабораторной работы

## 4.1 Создание программы “Hello world!”

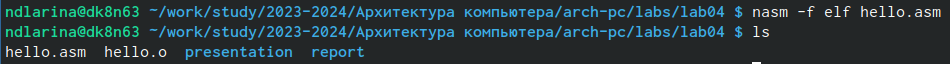
С помощью утилиты cd перемещаюсь в каталог, в котором буду выполнять работу, и создаю в текущем каталоге пустой текстовый файл hello.asm с помощью утилиты touch (рис. [??]). Перемещение между директориями и создание файла

Открываю созданный файл в текстовом редакторе gedit (рис. [??]). 

Заполняю файл, вставляя в него программу “Hello world!” (рис. [??]). 

## 4.2 Работа с транслятором NASM

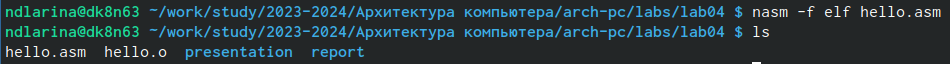
Превращаю текст программы для вывода “Hello world!” в объектный код с помощью транслятора NASM, используя команду nasm -f elf hello.asm, ключ -f ELF (рис. [??]). Далее проверяю правильность выполнения команды с помощью утилиты ls.



Компиляция текста программы

## 4.3 Работа с расширенным синтаксисом командой строки NASM

Сначала ввожу команду, которая скомпилирует файл hello.asm в файл obj.o, при этом в файл будут включены символы для отладки (ключ -g), также с помощью ключа -l будет создан файл листинга list.lst (рис. [??]). И проверяю правильность выполнения команды, используя утилиту ls.



Компиляция текста программы

## 4.4 Работа с компоновщиками LD

Передаю объектный файл hello.o на обработку компановщику LD, чтобы получить исполняемый файл hello (рис. [??]). Ключ -о задаёт имя создаваемого исполняемого файла. затем проверяю правильность выполнения команды с помощью ls.

Передача объектного файла на обработку компоновщику

Передача объектного файла на обработку компоновщику

Выполняю следующую команду (рис. [??]). Исполняемый файл будет иметь имя main, так как после ключа -о было задано значение main. Объектный файл, из которого собран этот исполняемый файл, имеет имя obj.o.

Передача объектного файла на обработку компановщику

Передача объектного файла на обработку компановщику

## 4.5 Запуск исполняемого файла

Запусаю на выполнение созданный исполняемый файл hello (рис. [??]). Запуск исполняемого файла

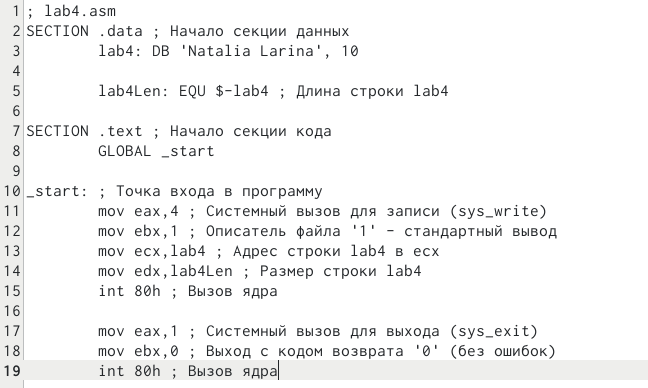
## 4.6 Выполнеие заданий для самостоятельной работы

С помощью команды ср создаю в текущем каталоге копию файла hello.asm c именем lab4.asm (рис. [??]).

Создание копии файла

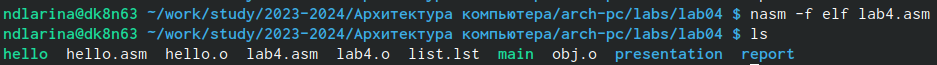
Создание копии файла

Далее с помощью текстового редактора gedit открываю файл lab4.asm и вношу изменения в программу так, чтобы она выводила моё имя и фамилию (рис. [??]).



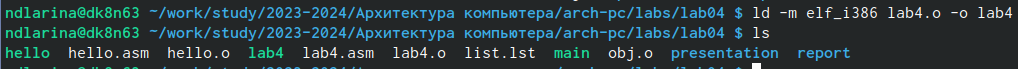
Изменение программы

Компилирую текст программы в объектный файл (рис. [??]). И проверяю правильность выполнения команды с помощью утилиты ls.



Компиляция текста программы

Передаю объектный файл lab4.o на обработку компоновщику LD, чтобы получить исполняемый файл lab4 (рис. [??]).



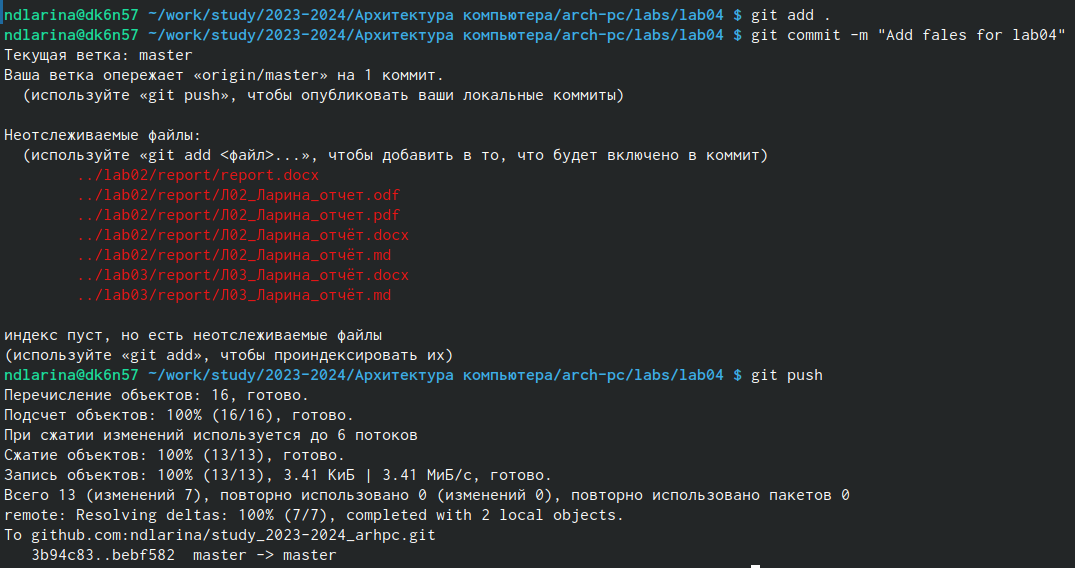
Передача объектного файла на обработку компоновщику

Запускаю исполняемый файл lab4. Всё работает (рис. [??]).

Запуск исполняемого файла

Запуск исполняемого файла

Далее с помощью команд git add ., git commit и git push добавляю файлы на github, комментируя действие как добавление файлов для лабораторной работы №4, и затем отправляю файлы на сервер (рис. [??]).



Добавление и отправка файлов на GitHub

# 5 Выводы

В результате выполнения лабораторной работы мне удалось освоить процедуры компиляции и сборки программ, написанных на ассемблере NASM.

# Список литературы

1. https://esystem.rudn.ru/mod/resource/view.php?id=1030552