

Отчет по лабораторной работе №4

Дисциплина: архитектура компьютера

Кузнецова Елизавета Андреевна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Выполнение заданий для самостоятельной работы	12
6	Выводы	14

Список иллюстраций

4.1	Создание каталога для работы и перемещение между директориями	9
4.2	Создание файла и его открытие в текстовом редакторе	9
4.3	Заполнение файла	9
4.4	Компиляция текста программы	10
4.5	Компиляция текста программы	10
4.6	Передача объектного файла на обработку компоновщику	10
4.7	Передача объектного файла на обработку компоновщику	10
4.8	Запуск исполняемого файла	11
5.1	Создание копии файла	12
5.2	Изменение программы	12
5.3	Компиляция текста программы	12
5.4	Передача объектного файла на обработку компоновщику	13
5.5	Запуск исполняемого файла	13
5.6	Загрузка файлов на локальный репозиторий	13
5.7	Добавление файлов на GitHub	13
5.8	Отправка файлов	13

Список таблиц

1 Цель работы

Целью работы является освоение процедуры компиляции и сборки программ, написанных на ассамблере NASM.

2 Задание

1. Создание программы Hello, world!
2. В каталоге `~/work/arch-pc/lab04` с помощью команды `cp` создайте копию файла `hello.asm` с именем `lab4.asm`
3. С помощью любого текстового редактора внесите изменения в текст программы в файле `lab4.asm` так, чтобы вместо `Hello world!` на экран выводилась строка с вашими фамилией и именем.
4. Оттранслируйте полученный текст программы `lab4.asm` в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл.
5. Скопируйте файлы `hello.asm` и `lab4.asm` в Ваш локальный репозиторий в каталог `~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/`. Загрузите файлы на Github

3 Теоретическое введение

Основными функциональными элементами любой электронно-вычислительной машины (ЭВМ) являются центральный процессор, память и периферийные устройства. Взаимодействие этих устройств осуществляется через общую шину, к которой они подключены. Физически шина представляет собой большое количество проводников, соединяющих устройства друг с другом. В современных компьютерах проводники выполнены в виде электропроводящих 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 арифметические 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 состав 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, преобразование (арифме-

тические или логические операции) данных хранящихся в регистрах. Другим важным узлом ЭВМ является оперативное запоминающее устройство (ОЗУ). ОЗУ — это быстродействующее энергозависимое запоминающее устройство, которое непосредственно взаимодействует с узлами процессора, предназначенное для хранения программ и данных, с которыми процессор непосредственно работает в текущий момент. ОЗУ состоит из одинаковых пронумерованных ячеек памяти. Номер ячейки памяти — это адрес хранящихся в ней данных.

4 Выполнение лабораторной работы

Создаю каталог для работы с программами на языке ассемблера NASM и с помощью команды `cd` перемещаюсь в созданный каталог (рис. [4.1]).

```
eakuznecova@dk8n68 ~ $ mkdir -p ~/work/arch-pc/lab04
eakuznecova@dk8n68 ~ $ cd ~/work/arch-pc/lab04
```

Рис. 4.1: Создание каталога для работы и перемещение между директориями

Создаю в текущем каталоге текстовый файл `hello.asm` с помощью утилиты `touch` и открываю созданный файл в текстовом редакторе `gedit` (рис. [4.2]).

```
eakuznecova@dk2n22 ~/work/arch-pc/lab04 $ touch hello.asm
eakuznecova@dk2n22 ~/work/arch-pc/lab04 $ gedit hello.asm
```

Рис. 4.2: Создание файла и его открытие в текстовом редакторе

Заполняю файл, вставляя в него программу (рис. [4.3]).

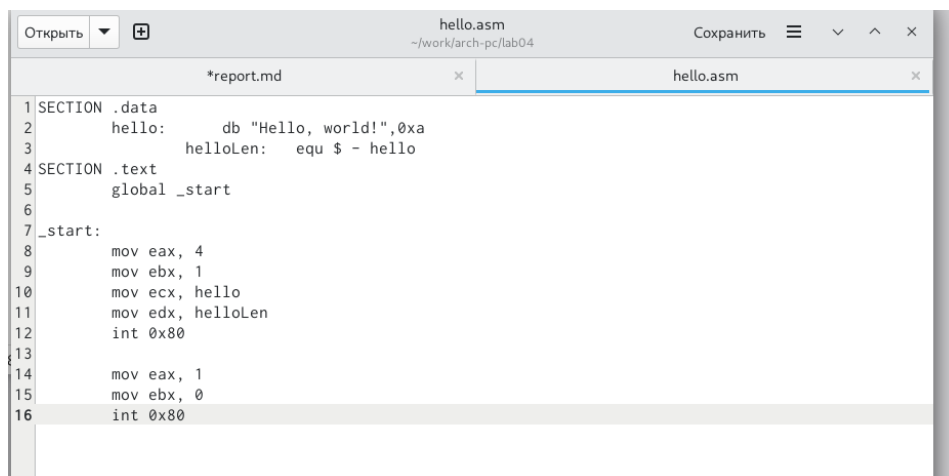


Рис. 4.3: Заполнение файла

Превращаю текст программы в объектный код с помощью транслятора NASM, используя команду `nasm -f elf hello.asm`. Далее проверяю правильность выполнения команды с помощью команды `ls` (рис. [4.4]).

```
eakuznecova@dk2n22 ~/work/arch-pc/lab04 $ nasm -f elf hello.asm
eakuznecova@dk2n22 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o
```

Рис. 4.4: Компиляция текста программы

Использую команду, которая скомпилирует исходный файл `hello.asm` в `obj.o`, при этом формат выходного файла будет `elf` и в него будут включены символы для отладки (ключ `-g`), кроме того, будет создан файл листинга `list.lst`. Далее с помощью команды `ls` проверяю, что файлы были созданы (рис. [4.5]).

```
eakuznecova@dk2n22 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
eakuznecova@dk2n22 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o  list.lst  obj.o
```

Рис. 4.5: Компиляция текста программы

Передаю объектный файл `hello.o` на обработку компоновщику LD, чтобы получить исполняемый файл `hello`. Далее проверяю с помощью команды `ls` правильность выполнения команды (рис. [4.6]).

```
eakuznecova@dk2n22 ~/work/arch-pc/lab04 $ ld -m elf_i386 hello.o -o hello
eakuznecova@dk2n22 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  list.lst  obj.o
```


Рис. 4.6: Передача объектного файла на обработку компоновщику

Использую команду, которая позволяет исполняемому файлу иметь имя `main`, потому что с помощью ключа `-o` было создано значение `main`. Объектный файл будет иметь имя `obj.o` (рис. [4.7]).

```
eakuznecova@dk2n22 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
eakuznecova@dk2n22 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
```

Рис. 4.7: Передача объектного файла на обработку компоновщику

Запускаю на выполнение созданный исполняемый файл hello (рис. [4.8]).

A terminal window with a dark background. The prompt is 'eakuznecova@dk2n22 ~/work/arch-pc/lab04 \$'. The command './hello' has been entered and executed, resulting in the output 'Hello, world!' on the next line.

```
eakuznecova@dk2n22 ~/work/arch-pc/lab04 $ ./hello
Hello, world!
```

Рис. 4.8: Запуск исполняемого файла

5 Выполнение заданий для самостоятельной работы

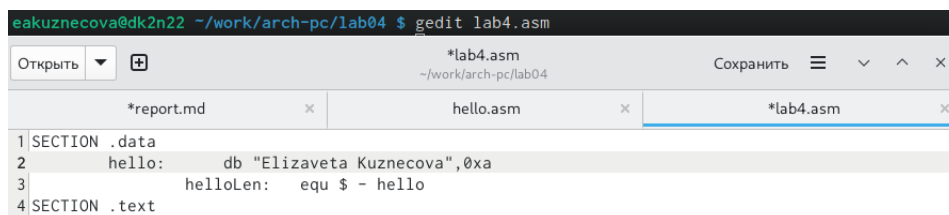
С помощью команды `cp` создаю в текущем каталоге копию файла `hello.asm` с именем `lab4.asm` (рис. [5.1]).

```
eakuznecova@dk2n22 ~/work/arch-pc/lab04 $ cp hello.asm lab4.asm
```

Рис. 5.1: Создание копии файла

С помощью текстового редактора `gedit` открываю файл `lab4.asm`, вношу изменения в программу, чтобы она выводила мои имя и фамилию (рис. [5.2]).

```
eakuznecova@dk2n22 ~/work/arch-pc/lab04 $ gedit lab4.asm
```



```
1 SECTION .data
2     hello:      db "Elizaveta Kuznecova",0xa
3               helloLen: equ $ - hello
4 SECTION .text
5     main:
6         mov     edi,hello
7         mov     ecx,helloLen
8         mov     ebx,0
9         mov     esi,0
10        mov     edi,hello
11        mov     ecx,helloLen
12        mov     ebx,0
13        mov     esi,0
```

Рис. 5.2: Изменение программы

Компилирую текст программы в объектный файл. Проверяю с помощью команды `ls`, что файл `lab4.o` создан (рис. [5.3]).

```
eakuznecova@dk2n22 ~/work/arch-pc/lab04 $ nasm -f elf lab4.asm
eakuznecova@dk2n22 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  lab4.asm  lab4.o  list.lst  main  obj.o
```

Рис. 5.3: Компиляция текста программы

Передаю объектный файл lab4.o на обработку компоновщику LD, чтобы получить исполняемый файл lab4 (рис. [5.4]).

```
eakuznecova@dk2n22 ~/work/arch-pc/lab04 $ ld -m elf_i386 lab4.o -o lab4
eakuznecova@dk2n22 ~/work/arch-pc/lab04 $ ls
hello hello.asm hello.o lab4 lab4.asm lab4.o list.lst main obj.o
```

Рис. 5.4: Передача объектного файла на обработку компоновщику

Запускаю исполняемый файл lab4 (рис. [5.5]).

```
eakuznecova@dk2n22 ~/work/arch-pc/lab04 $ ./lab4
Elizaveta Kuznecova
```

Рис. 5.5: Запуск исполняемого файла

Копирую файлы hello.asm и lab4.asm в свой локальный репозиторий в каталог ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/. Перехожу в этот каталог и проверяю наличие файлов в нем (рис. [5.6]).

```
eakuznecova@dk2n22 ~/work/arch-pc/lab04 $ cp hello.asm lab4.asm ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/
eakuznecova@dk2n22 ~/work/arch-pc/lab04 $ cd ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/
eakuznecova@dk2n22 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04 $ ls
hello.asm lab4.asm presentation report
```

Рис. 5.6: Загрузка файлов на локальный репозиторий

С помощью команд git add . и git commit добавляю файлы на GitHub (рис. [5.7]).

```
eakuznecova@dk2n22 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04 $ git add .
eakuznecova@dk2n22 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04 $ git commit -m "Add fales for lab04"
[master c185409] Add fales for lab04
17 files changed, 81 insertions(+), 33 deletions(-)
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/lab4.asm
```

Рис. 5.7: Добавление файлов на GitHub

Отправляю файлы на сервер с помощью команды git push (рис. [5.8]).

```
eakuznecova@dk2n22 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04 $ git push
Перечисление объектов: 29, готово.
Подсчет объектов: 100% (29/29), готово.
При сжатии изменений используется до 6 потоков
```

Рис. 5.8: Отправка файлов

6 Выводы

Я освоила процедуры компиляции и сборки программ, написанных на ассемблере NASM.