

Отчет по лабораторной работе №4

**Создание и процесс обработки программ на языке ассемблера
NASM**

Майоров Дмитрий Андреевич

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	12
	Список литературы	13

Список иллюстраций

3.1	Создаем каталог для работы с программами на языке ассемблера NASM	7
3.2	Переходим в созданный каталог	7
3.3	Создаем текстовый файл с именем hello.asm и открываем его . . .	7
3.4	Вводим нужный текст	8
3.5	Используем команду nasm и проверяем, создался ли объектный файл	8
3.6	Компилируем исходный файл и проверяем как сработала команда	8
3.7	Передаем объектный файл на обработку компоновщику и проверяем, создался ли файл hello	9
3.8	Передаем объектный файл на обработку компоновщику и проверяем, создался ли файл main	9
3.9	Запускаем исполняемый файл	9
3.10	Создаем копию файла hello.asm и открываем его	9
3.11	Редактируем файл	10
3.12	Прописываем те же команды что и в первой программе	10
3.13	Копируем файлы в локальный репозиторий	10
3.14	Переходим в каталог лабораторных работ и отправляем на GitHub	11

Список таблиц

1 Цель работы

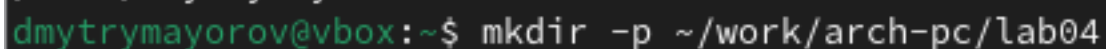
Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

2 Задание

Написать 2 программы (“Hello world!”, “Имя Фамилия”)

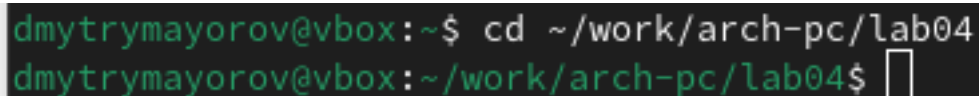
3 Выполнение лабораторной работы

##3.1 Программа Hello world!



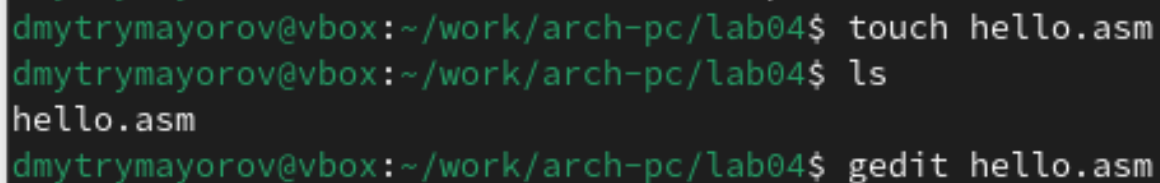
```
dmytrymayorov@vbox:~$ mkdir -p ~/work/arch-pc/lab04
```

Рис. 3.1: Создаем каталог для работы с программами на языке ассемблера NASM



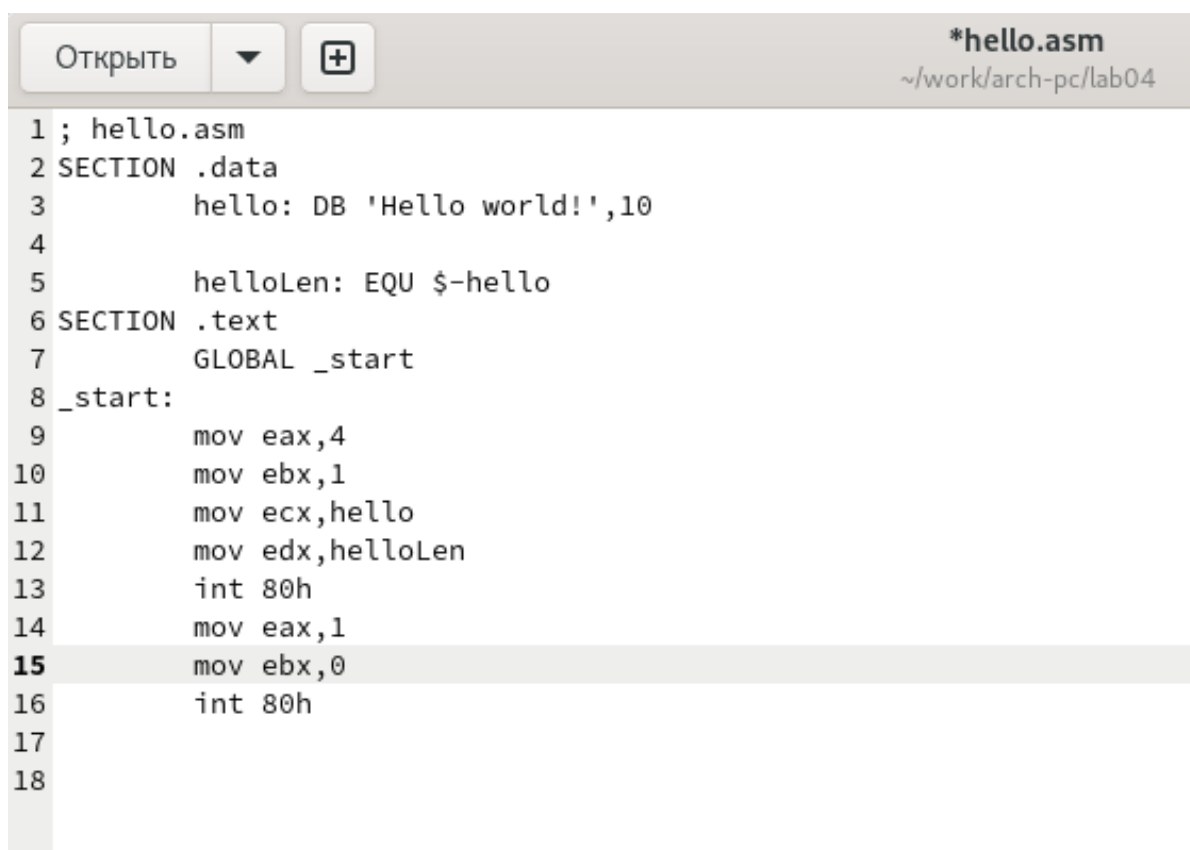
```
dmytrymayorov@vbox:~$ cd ~/work/arch-pc/lab04  
dmytrymayorov@vbox:~/work/arch-pc/lab04$
```

Рис. 3.2: Переходим в созданный каталог



```
dmytrymayorov@vbox:~/work/arch-pc/lab04$ touch hello.asm  
dmytrymayorov@vbox:~/work/arch-pc/lab04$ ls  
hello.asm  
dmytrymayorov@vbox:~/work/arch-pc/lab04$ gedit hello.asm
```

Рис. 3.3: Создаем текстовый файл с именем hello.asm и открываем его

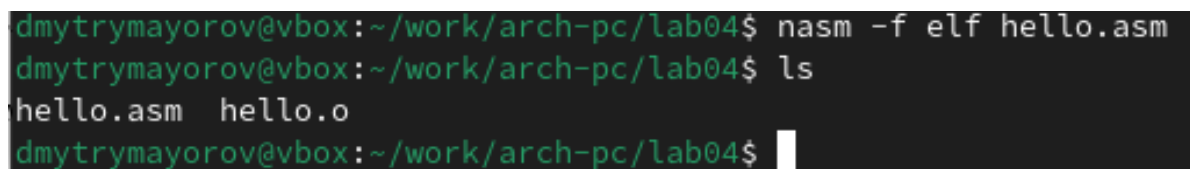


The screenshot shows a text editor window titled '*hello.asm' with the path '~/work/arch-pc/lab04'. The code is as follows:

```
1 ; hello.asm
2 SECTION .data
3     hello: DB 'Hello world!',10
4
5     helloLen: EQU $-hello
6 SECTION .text
7     GLOBAL _start
8 _start:
9     mov eax,4
10    mov ebx,1
11    mov ecx,hello
12    mov edx,helloLen
13    int 80h
14    mov eax,1
15    mov ebx,0
16    int 80h
17
18
```

Рис. 3.4: Вводим нужный текст

##Транслятор NASM

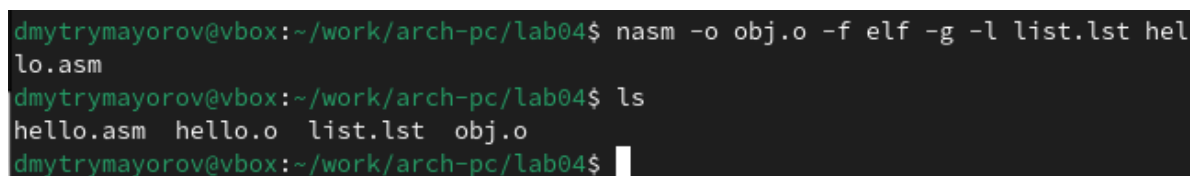


The screenshot shows a terminal window with the following commands and output:

```
dmytrymayorov@vbox:~/work/arch-pc/lab04$ nasm -f elf hello.asm
dmytrymayorov@vbox:~/work/arch-pc/lab04$ ls
hello.asm  hello.o
dmytrymayorov@vbox:~/work/arch-pc/lab04$
```

Рис. 3.5: Используем команду nasm и проверяем, создался ли объектный файл

##Расширенный синтаксис командной строки NASM



The screenshot shows a terminal window with the following commands and output:

```
dmytrymayorov@vbox:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst hel
lo.asm
dmytrymayorov@vbox:~/work/arch-pc/lab04$ ls
hello.asm  hello.o  list.lst  obj.o
dmytrymayorov@vbox:~/work/arch-pc/lab04$
```

Рис. 3.6: Компилируем исходный файл и проверяем как сработала команда

##Компоновщик LD

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

Рис. 3.7: Передаем объектный файл на обработку компоновщику и проверяем, создался ли файл hello

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

Рис. 3.8: Передаем объектный файл на обработку компоновщику и проверяем, создался ли файл main

##Запуск исполняемого файла

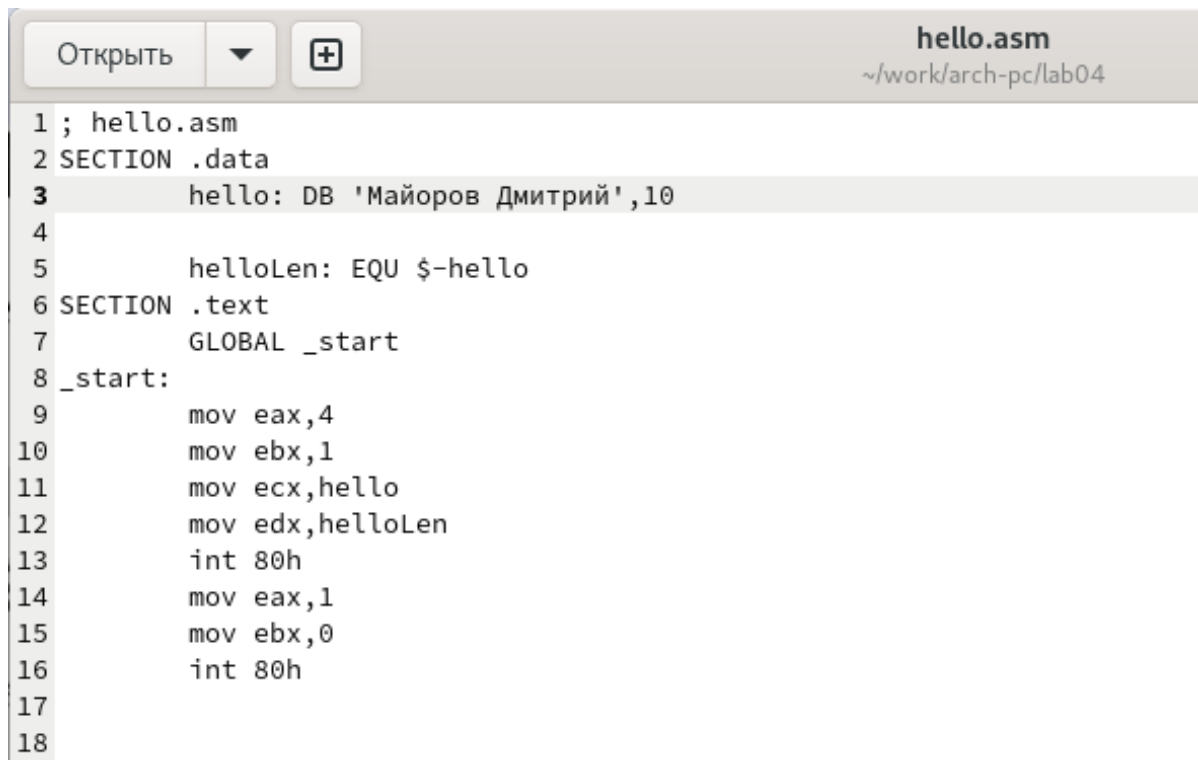
```
dmytrymayorov@vbox:~/work/arch-pc/lab04$ ./hello
Hello world!
dmytrymayorov@vbox:~/work/arch-pc/lab04$
```

Рис. 3.9: Запускаем исполняемый файл

##Задание для самостоятельной работы

```
dmytrymayorov@vbox:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
dmytrymayorov@vbox:~/work/arch-pc/lab04$ gedit hello.asm
```

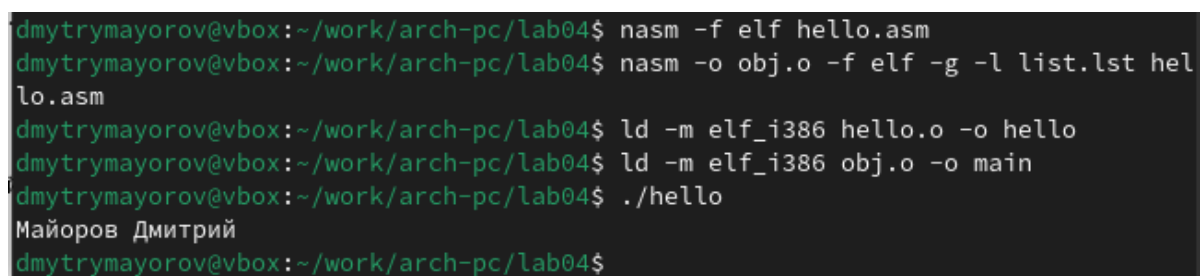
Рис. 3.10: Создаем копию файла hello.asm и открываем его



The screenshot shows a text editor window titled 'hello.asm' with the path '~/.work/arch-pc/lab04'. The editor contains the following assembly code:

```
1 ; hello.asm
2 SECTION .data
3     hello: DB 'Майоров Дмитрий',10
4
5     helloLen: EQU $-hello
6 SECTION .text
7     GLOBAL _start
8 _start:
9     mov eax,4
10    mov ebx,1
11    mov ecx,hello
12    mov edx,helloLen
13    int 80h
14    mov eax,1
15    mov ebx,0
16    int 80h
17
18
```

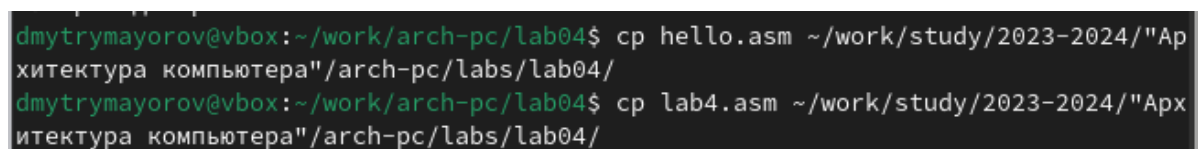
Рис. 3.11: Редактируем файл



The screenshot shows a terminal window with the following commands and output:

```
dmytrymayorov@vbox:~/work/arch-pc/lab04$ nasm -f elf hello.asm
dmytrymayorov@vbox:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst hel
lo.asm
dmytrymayorov@vbox:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
dmytrymayorov@vbox:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
dmytrymayorov@vbox:~/work/arch-pc/lab04$ ./hello
Майоров Дмитрий
dmytrymayorov@vbox:~/work/arch-pc/lab04$
```

Рис. 3.12: Прописываем те же команды что и в первой программе



The screenshot shows a terminal window with the following commands:

```
dmytrymayorov@vbox:~/work/arch-pc/lab04$ cp hello.asm ~/.work/study/2023-2024/"Арх
итектура компьютера"/arch-pc/labs/lab04/
dmytrymayorov@vbox:~/work/arch-pc/lab04$ cp lab4.asm ~/.work/study/2023-2024/"Арх
итектура компьютера"/arch-pc/labs/lab04/
```

Рис. 3.13: Копируем файлы в локальный репозиторий

```

dmytrymayorov@vbox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git add .
dmytrymayorov@vbox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git commit -am 'feat(main): add files lab-4'
[master bba44de] feat(main): add files lab-4
14 files changed, 36 insertions(+)
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/lab4.asm
create mode 100644 labs/lab04/report/image/01.png
create mode 100644 labs/lab04/report/image/02.png
create mode 100644 labs/lab04/report/image/03.png
create mode 100644 labs/lab04/report/image/04.png
create mode 100644 labs/lab04/report/image/05.png
create mode 100644 labs/lab04/report/image/06.png
create mode 100644 labs/lab04/report/image/07.png
create mode 100644 labs/lab04/report/image/08.png
create mode 100644 labs/lab04/report/image/09.png
create mode 100644 labs/lab04/report/image/10.png
create mode 100644 labs/lab04/report/image/11.png
create mode 100644 labs/lab04/report/image/12.png
dmytrymayorov@vbox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git push
Перечисление объектов: 25, готово.
Подсчет объектов: 100% (25/25), готово.
Сжатие объектов: 100% (20/20), готово.
Запись объектов: 100% (20/20), 191.13 КиБ | 1.38 МиБ/с, готово.
Total 20 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 3 local objects.
To github.com:DmytryMay/study_2023-2024_arhpc.git
6811962..bba44de master -> master
dmytrymayorov@vbox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$

```

Рис. 3.14: Переходим в каталог лабораторных работ и отправляем на GitHub

4 Выводы

Я освоил процедуры компиляции и сборки программ, написал 2 программы.

Список литературы