

Отчёт о лабораторной работе №5

Дисциплина: архитектура компьютера

Барсегян Вардан Левонович НПИбд-01-22

Содержание

Цель работы.....	1
Выполнение лабораторной работы.....	1
Программа Hello World!.....	1
Транслятор NASM.....	3
Расширенный синтаксис командной строки NASM.....	4
Компоновщик LD.....	4
Запуск исполняемого файла.....	5
Задание для самостоятельной работы.....	6
Выводы.....	7

Цель работы

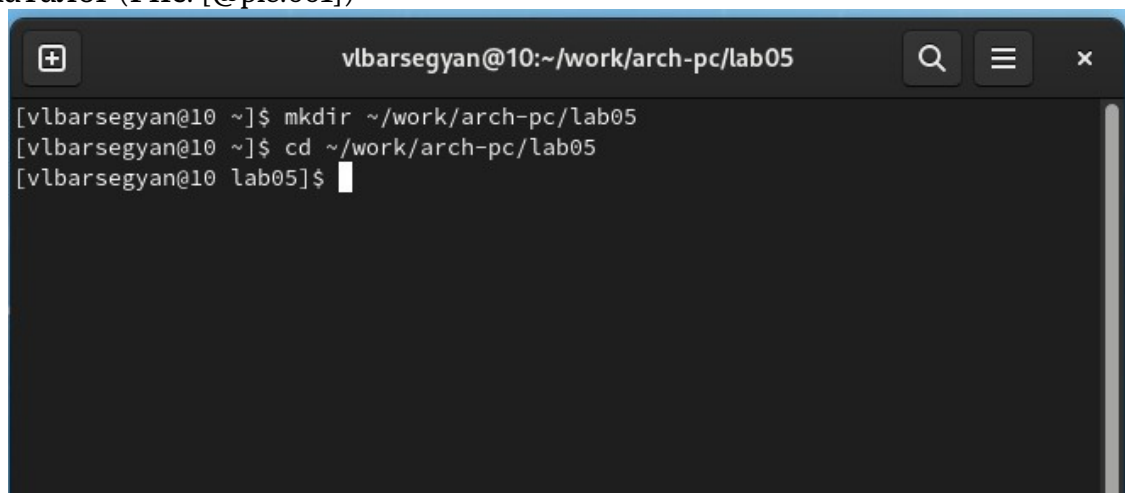
Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

Выполнение лабораторной работы

Программа Hello World!

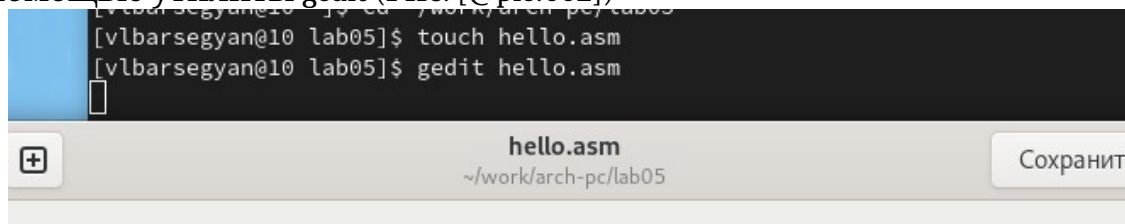
1. Создаю каталог для работы с программами на языке ассемблера NASM с помощью команды `mkdir ~/work/arch-pc/lab05` и перехожу в данный

каталог (Рис. [pic:001])



```
vlbarsegyan@10:~/work/arch-pc/lab05
[vlbarsegyan@10 ~]$ mkdir ~/work/arch-pc/lab05
[vlbarsegyan@10 ~]$ cd ~/work/arch-pc/lab05
[vlbarsegyan@10 lab05]$
```

2. Создайте текстовый файл с именем hello.asm и открываю его с помощью утилиты gedit (Рис. [pic:002])

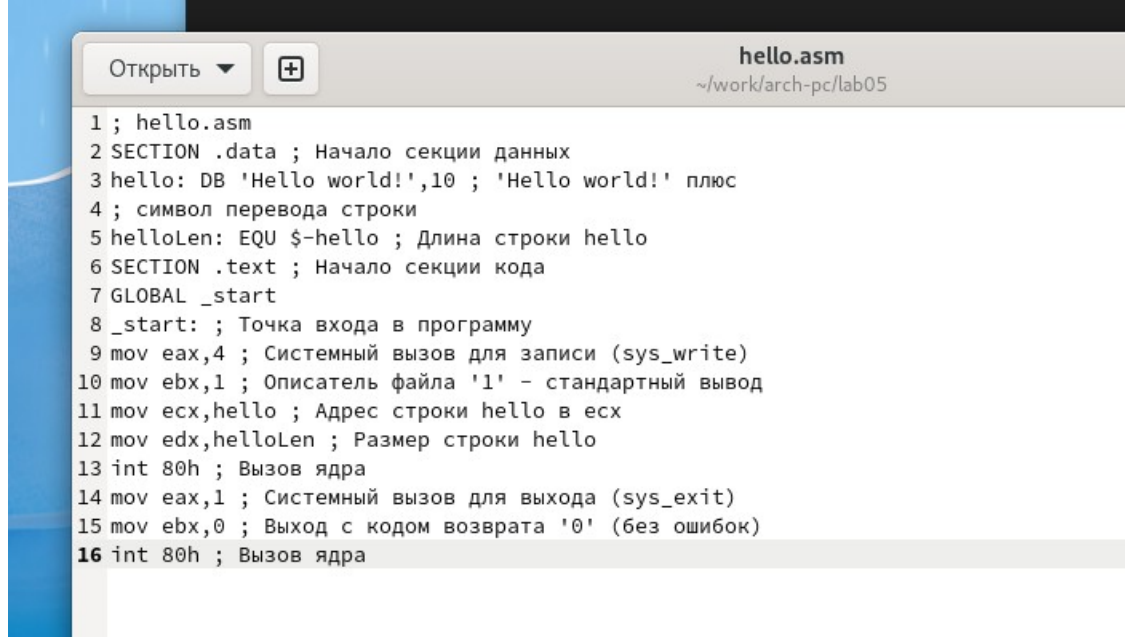


```
vlbarsegyan@10:~/work/arch-pc/lab05
[vlbarsegyan@10 lab05]$ touch hello.asm
[vlbarsegyan@10 lab05]$ gedit hello.asm
```

hello.asm
~/work/arch-pc/lab05

Сохранить

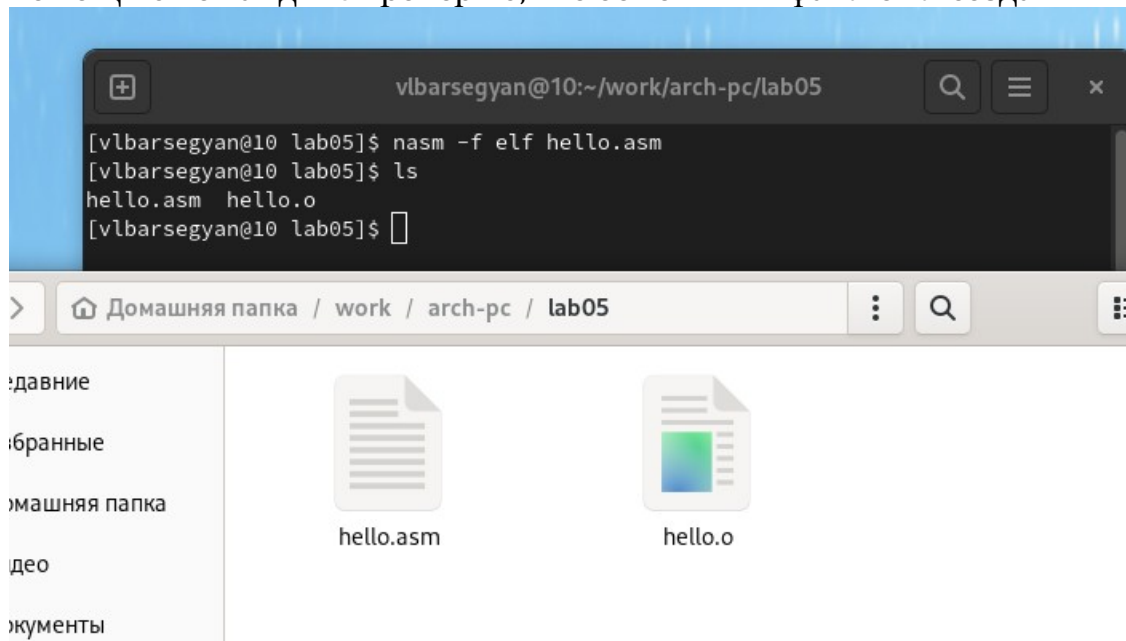
3. Копирую текст из лабораторной работы и вставляю его в файл (Рис. [pic:003])

A screenshot of a text editor window titled 'hello.asm' with a path of '~/.work/arch-pc/lab05'. The editor contains 16 lines of assembly code for a 'hello world' program. The code includes section declarations for data and text, variable definitions for 'hello' and 'helloLen', and instructions for writing to stdout and exiting the program.

```
1 ; hello.asm
2 SECTION .data ; Начало секции данных
3 hello: DB 'Hello world!',10 ; 'Hello world!' плюс
4 ; символ перевода строки
5 helloLen: EQU $-hello ; Длина строки hello
6 SECTION .text ; Начало секции кода
7 GLOBAL _start
8 _start: ; Точка входа в программу
9 mov eax,4 ; Системный вызов для записи (sys_write)
10 mov ebx,1 ; Описатель файла '1' - стандартный вывод
11 mov ecx,hello ; Адрес строки hello в ecx
12 mov edx,helloLen ; Размер строки hello
13 int 80h ; Вызов ядра
14 mov eax,1 ; Системный вызов для выхода (sys_exit)
15 mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
16 int 80h ; Вызов ядра
```

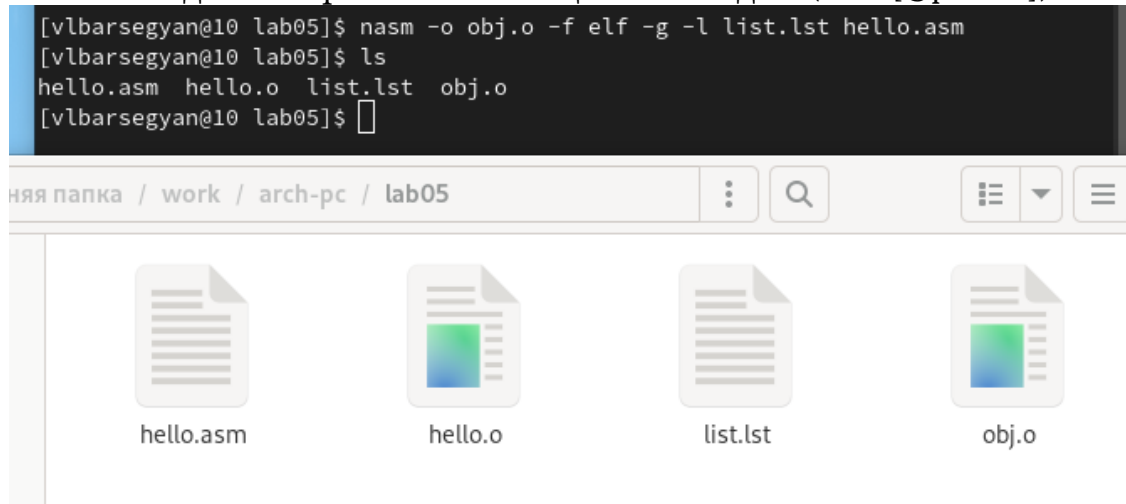
Транслятор NASM

4. Компилирую программу из файла hello.asm в объектный код с помощью команды `nasm -f elf hello.asm` (Рис. [pic:004]). Полученный файл имеет такое же название, как и исходный, но с расширением `.o`. Также с помощью команды `ls` проверяю, что объектный файл был создан



Расширенный синтаксис командной строки NASM

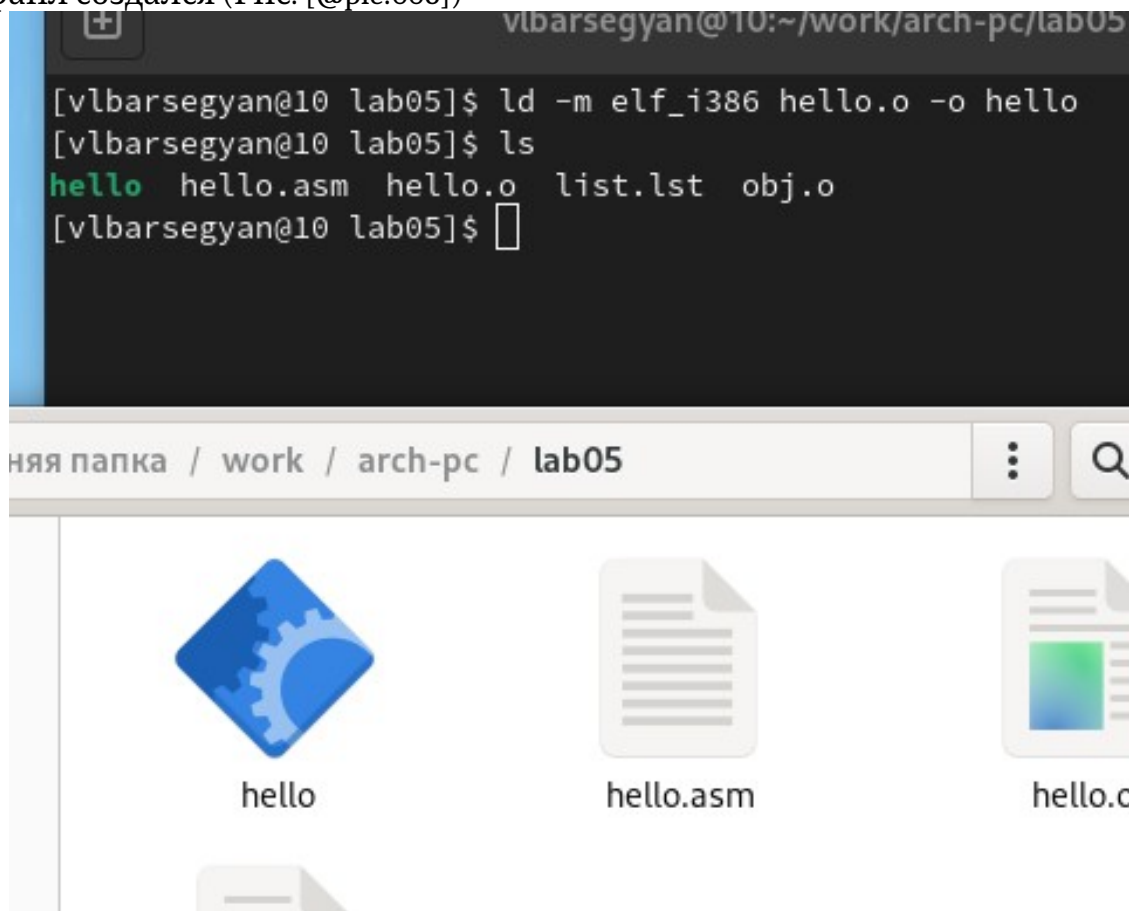
5. Выполняю команду `nasm -o obj.o -f elf -g -l list.lst hello.asm` и проверяю наличие созданных файлов с помощью команды `ls` (Рис. [pic:005])



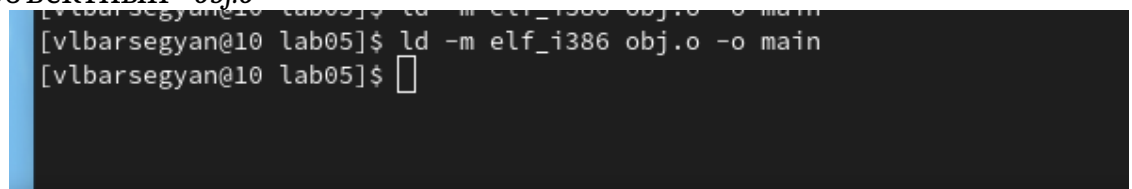
Компоновщик LD

6. Передаю на обработку компоновщику объектный файл, чтобы получить исполняемую программу, используя команду `ld -m elf_i386 hello.o -o hello`. Также с помощью команды `ls` проверяю, что исполняемый

файл создался (Рис. [pic:006])

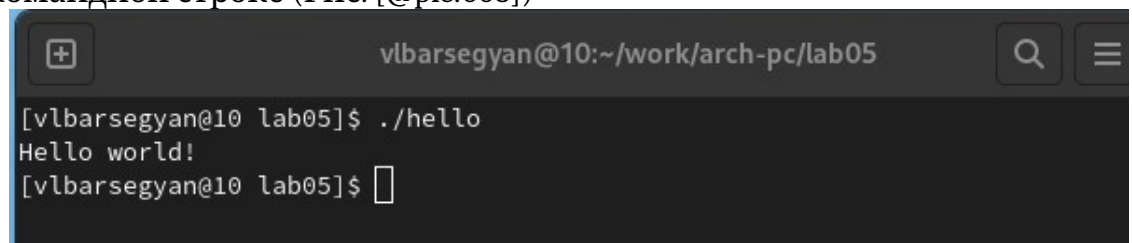


7. Выполняю следующую команду: `ld -m elf_i386 obj.o -o main` (Рис. [pic:007]). Полученный исполняемый файл имеет название *main*, а исходный объектный - *obj.o*



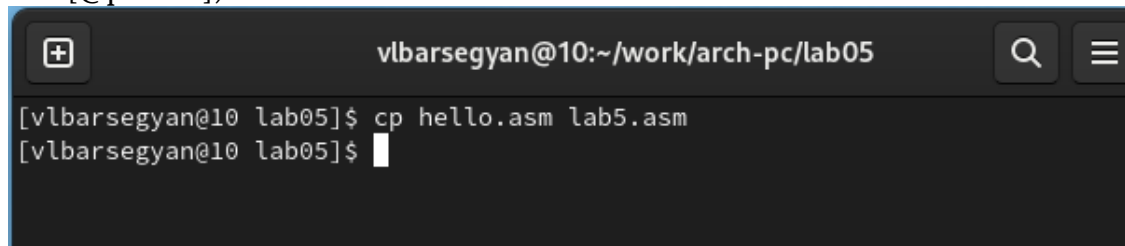
Запуск исполняемого файла

8. Запускаю созданный исполняемый файл с помощью команды `./hello` в командной строке (Рис. [pic:008])



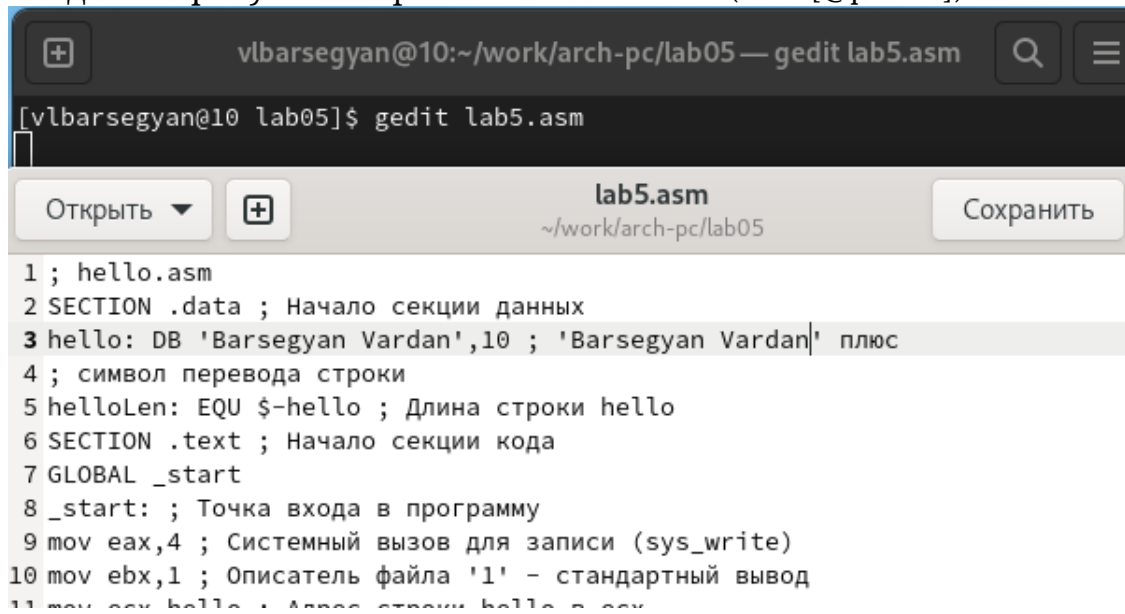
Задание для самостоятельной работы

1. С помощью команды `cp` создаю копию файла `hello.asm` с именем `lab5.asm` (Рис. [pic:009])



```
vlbarsegyan@10:~/work/arch-pc/lab05
[vlbarsegyan@10 lab05]$ cp hello.asm lab5.asm
[vlbarsegyan@10 lab05]$
```

2. С помощью утилиты `gedit` открываю файл `lab5.asm` и меняю текст для вывода на строку с моей фамилией и именем (Рис. [pic:010])



```
vlbarsegyan@10:~/work/arch-pc/lab05 — gedit lab5.asm
[vlbarsegyan@10 lab05]$ gedit lab5.asm

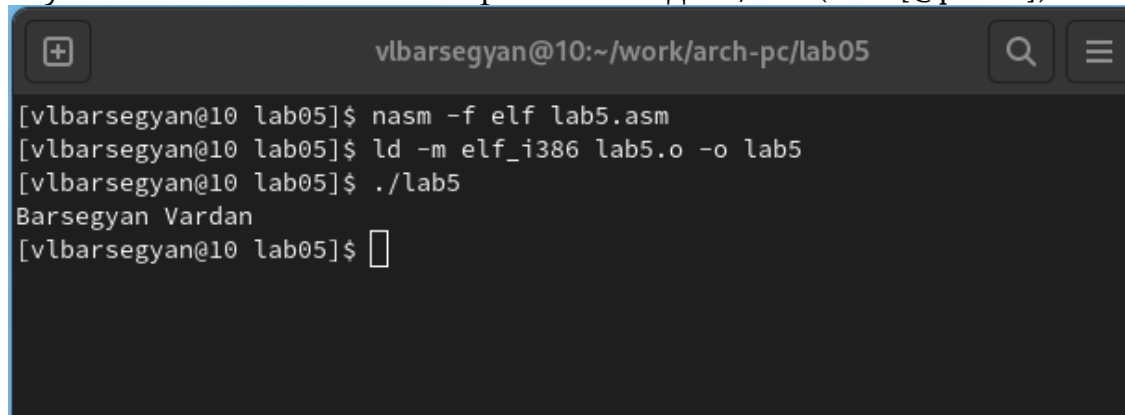
Открыть ▼ lab5.asm Сохранить
~/work/arch-pc/lab05

1 ; hello.asm
2 SECTION .data ; Начало секции данных
3 hello: DB 'Barsegyan Vardan',10 ; 'Barsegyan Vardan' плюс
4 ; символ перевода строки
5 helloLen: EQU $-hello ; Длина строки hello
6 SECTION .text ; Начало секции кода
7 GLOBAL _start
8 _start: ; Точка входа в программу
9 mov eax,4 ; Системный вызов для записи (sys_write)
10 mov ebx,1 ; Описатель файла '1' - стандартный вывод
11 mov ecx,hello ; Адрес строки hello в ecx
```

Текст измененной программы: ; hello.asm SECTION .data ; Начало секции данных hello: DB 'Barsegyan Vardan',10 ; 'Barsegyan Vardan' плюс ; символ перевода строки helloLen: EQU \$-hello ; Длина строки hello SECTION .text ; Начало секции кода GLOBAL _start _start: ; Точка входа в программу mov eax,4 ; Системный вызов для записи (sys_write) mov ebx,1 ; Описатель файла '1' - стандартный вывод mov ecx,hello ; Адрес строки hello в ecx mov edx,helloLen ; Размер строки hello int 80h ; Вызов ядра mov eax,1 ; Системный вызов для выхода (sys_exit) mov ebx,0 ; Выход с кодом возврата '0' (без ошибок) int 80h ; Вызов ядра

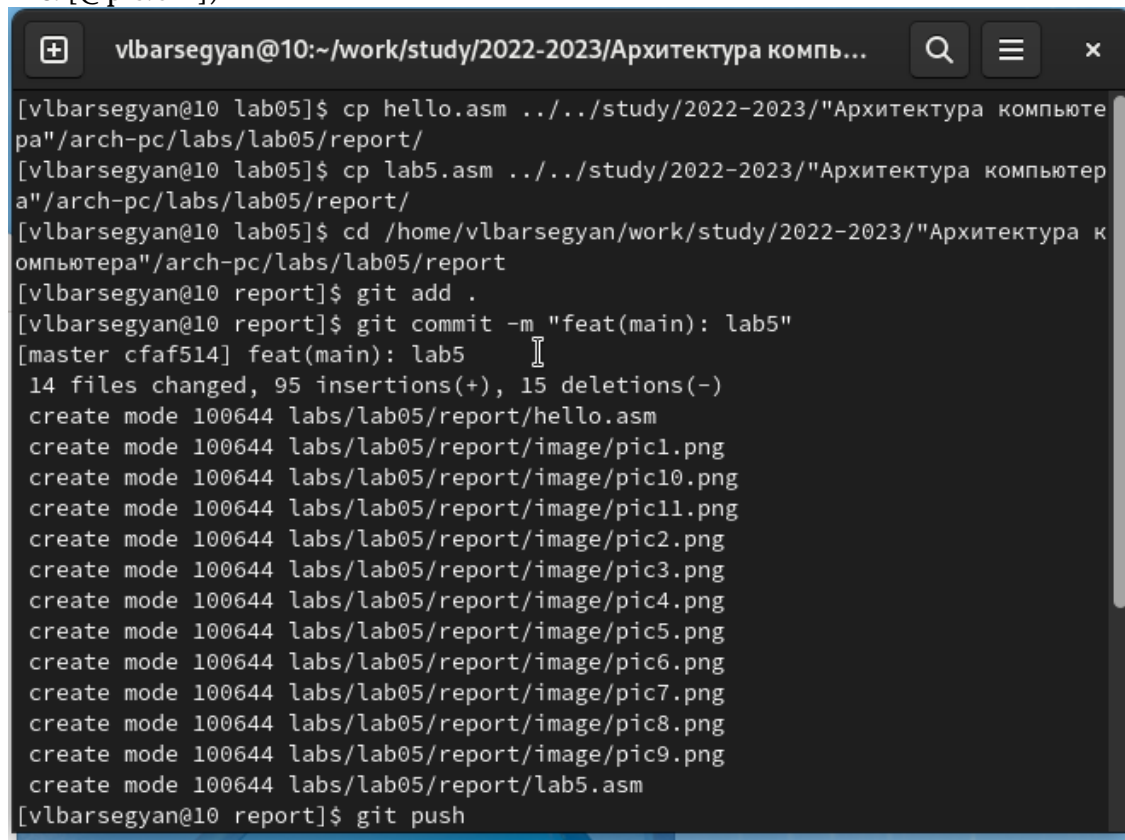
3. Транслирую полученный текст программы `lab5.asm` в объектный файл с помощью команды `nasm -f elf lab5.asm`, выполняю компоновку объектного файла (команда `d -m elf_i386 lab5.o -o lab5`) и запускаю

получившийся исполняемый файл командой `./lab5` (Рис. [pic:011])



```
vlbarsegyan@10:~/work/arch-pc/lab05
[vlbarsegyan@10 lab05]$ nasm -f elf lab5.asm
[vlbarsegyan@10 lab05]$ ld -m elf_i386 lab5.o -o lab5
[vlbarsegyan@10 lab05]$ ./lab5
Barsegyan Vardan
[vlbarsegyan@10 lab05]$
```

4. Копирую файлы `hello.asm` и `lab5.asm` в локальный репозиторий в каталог `~/work/study/2022-2023/“Архитектура компьютера”/arch-pc/labs/lab05/` (Рис. [pic:012])



```
vlbarsegyan@10:~/work/study/2022-2023/Архитектура компь...
[vlbarsegyan@10 lab05]$ cp hello.asm ../../study/2022-2023/"Архитектура компьюте
pa"/arch-pc/labs/lab05/report/
[vlbarsegyan@10 lab05]$ cp lab5.asm ../../study/2022-2023/"Архитектура компьюте
a"/arch-pc/labs/lab05/report/
[vlbarsegyan@10 lab05]$ cd /home/vlbarsegyan/work/study/2022-2023/"Архитектура к
омпьютера"/arch-pc/labs/lab05/report
[vlbarsegyan@10 report]$ git add .
[vlbarsegyan@10 report]$ git commit -m "feat(main): lab5"
[master cfaf514] feat(main): lab5
14 files changed, 95 insertions(+), 15 deletions(-)
create mode 100644 labs/lab05/report/hello.asm
create mode 100644 labs/lab05/report/image/pic1.png
create mode 100644 labs/lab05/report/image/pic10.png
create mode 100644 labs/lab05/report/image/pic11.png
create mode 100644 labs/lab05/report/image/pic2.png
create mode 100644 labs/lab05/report/image/pic3.png
create mode 100644 labs/lab05/report/image/pic4.png
create mode 100644 labs/lab05/report/image/pic5.png
create mode 100644 labs/lab05/report/image/pic6.png
create mode 100644 labs/lab05/report/image/pic7.png
create mode 100644 labs/lab05/report/image/pic8.png
create mode 100644 labs/lab05/report/image/pic9.png
create mode 100644 labs/lab05/report/lab5.asm
[vlbarsegyan@10 report]$ git push
```

Выводы

Я освоил процедуры компиляции и сборки программ, написанных на ассемблере NASM, научился пользоваться транслятором NASM, расширенным синтаксисом командной строки NASM, компоновщиком LD и запускать исполняемый файл. Также я загрузил обновления на github.