

# **Отчет по лабораторной работе №4**

Мухин Тимофей Владимирович

# Содержание

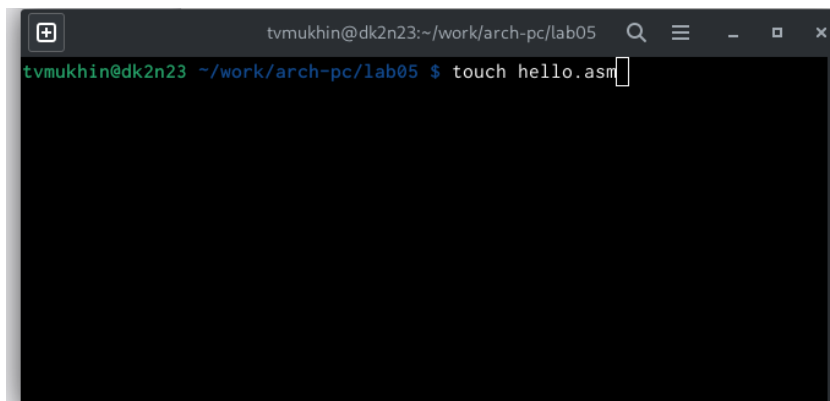
1	Цель работы	3
2	Выполнение лабораторной работы	4
3	Выводы	8

# 1 Цель работы

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

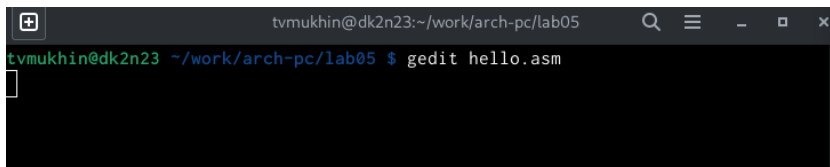
## 2 Выполнение лабораторной работы

1. Создаём каталог для работы с программами на языке ассемблера NASM, переходим в него и создаём текстовый файл с названием hello.asm



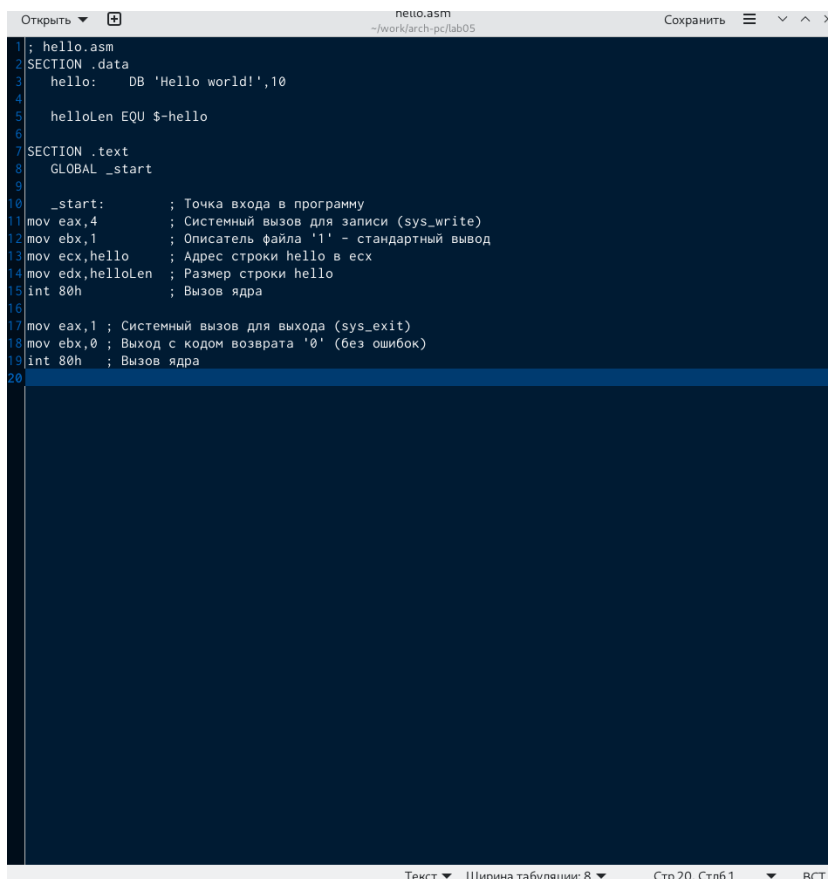
```
tvmukhin@dk2n23:~/work/arch-pc/lab05 $ touch hello.asm
```

2. Открываем файл с помощью текстового редактора gedit .



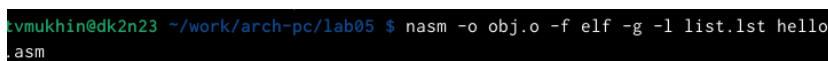
```
tvmukhin@dk2n23:~/work/arch-pc/lab05 $ gedit hello.asm
```

3. Вводим текст программы Hello World.



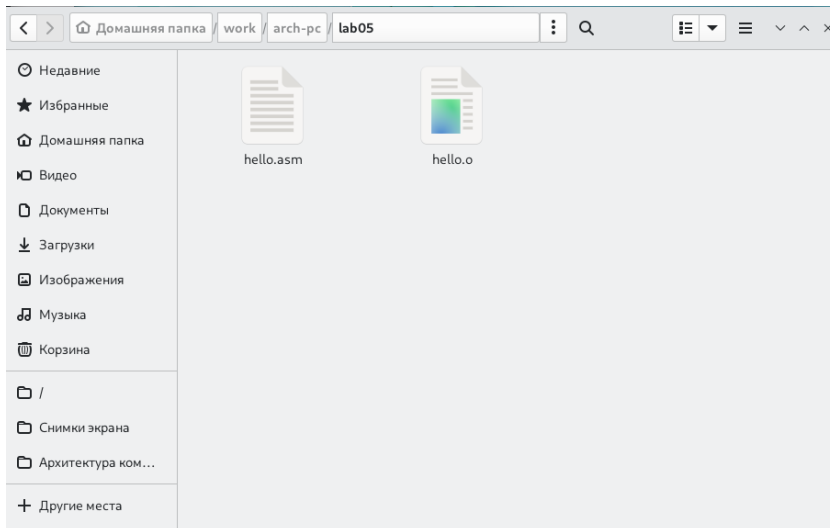
```
1; hello.asm
2SECTION .data
3    hello:    DB 'Hello world!',10
4
5    helloLen EQU $-hello
6
7SECTION .text
8    GLOBAL _start
9
10   _start:    ; Точка входа в программу
11   mov eax,4  ; Системный вызов для записи (sys_write)
12   mov ebx,1  ; Описатель файла '1' - стандартный вывод
13   mov ecx,hello ; Адрес строки hello в есх
14   mov edx,helloLen ; Размер строки hello
15   int 80h    ; Вызов ядра
16
17   mov eax,1 ; Системный вызов для выхода (sys_exit)
18   mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
19   int 80h    ; Вызов ядра
20
```

4. NASM превращает текст программы в объектный код. Если текст программы набран без ошибок, то транслятор преобразует текст программы из файла hello.asm в объектный код, который запишется в файл hello.o. Выполняем следующую команду:



```
tvmukhin@dk2n23 ~/work/arch-pc/lab05 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
```

5. Проверяем, что объектный файл был создан.



6. Чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику. С помощью команды `ls` проверяем, что исполняемый файл `hello` был создан.

```
tvmukhin@dk2n23:~/work/arch-pc/lab05
tvmukhin@dk2n23 ~/work/arch-pc/lab05 $ ld -m elf_i386 hello.o -o hello
tvmukhin@dk2n23 ~/work/arch-pc/lab05 $ ls
hello  hello.asm  hello.o  list.lst  obj.o
tvmukhin@dk2n23 ~/work/arch-pc/lab05 $
```

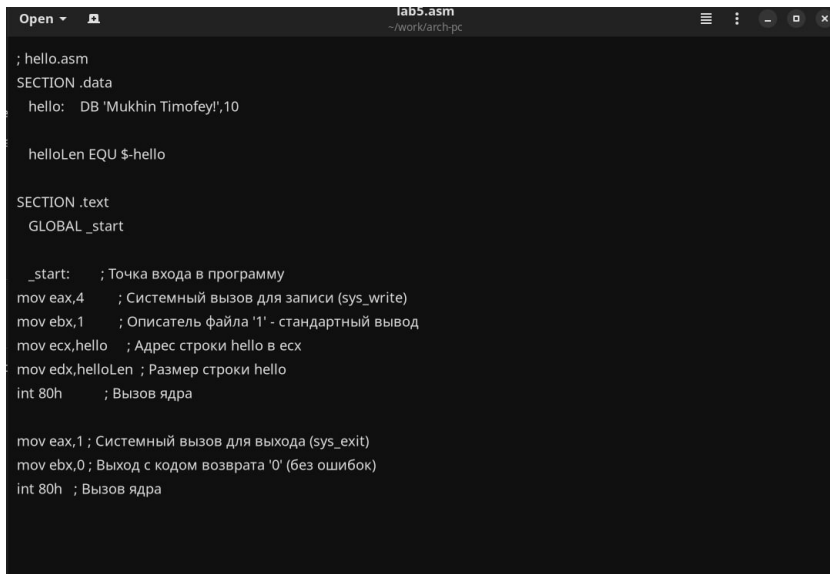
7. Запускаем исполняемый файл

```
tvmukhin@dk2n23:~/work/arch-pc/lab05
tvmukhin@dk2n23 ~/work/arch-pc/lab05 $ ./hello
Hello world!
tvmukhin@dk2n23 ~/work/arch-pc/lab05 $
```

8. С помощью команды `cp` создаём копию файла `hello.asm` с именем `lab5.asm`

```
tvmukhin@dk2n23:~/work/arch-pc/lab05
tvmukhin@dk2n23 ~/work/arch-pc/lab05 $ cp hello.asm lab5.asm
tvmukhin@dk2n23 ~/work/arch-pc/lab05 $
```

9. Вносим изменения в текст программы в файле так, чтобы вместо Hello world! на экран выводилась строка с фамилией и именем.



```
Open lab5.asm
~/work/arch-pc

; hello.asm
SECTION .data
hello: DB 'Mukhin Timofey!',10

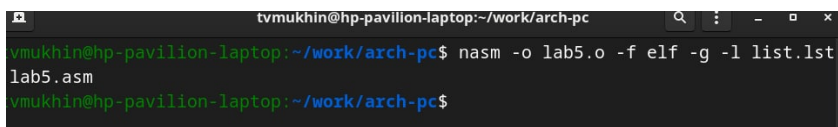
helloLen EQU $-hello

SECTION .text
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

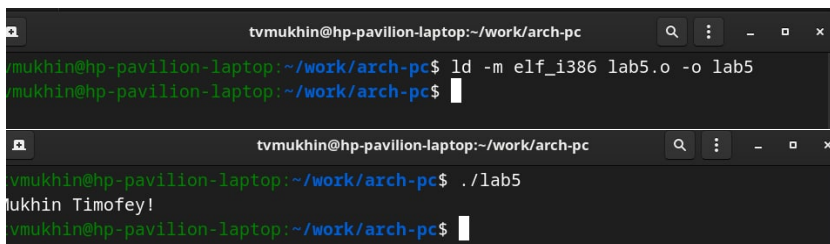
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
int 80h ; Вызов ядра
```

10. Транслируем полученный текст программы в объектный файл.



```
vmukhin@hp-pavilion-laptop:~/work/arch-pc$ nasm -o lab5.o -f elf -g -l list.lst lab5.asm
vmukhin@hp-pavilion-laptop:~/work/arch-pc$
```

11. Выполняем компоновку объектного файла и запускаем получившийся исполняемый файл.



```
vmukhin@hp-pavilion-laptop:~/work/arch-pc$ ld -m elf_i386 lab5.o -o lab5
vmukhin@hp-pavilion-laptop:~/work/arch-pc$ ./lab5
Mukhin Timofey!
vmukhin@hp-pavilion-laptop:~/work/arch-pc$
```

12. Копируем файлы в локальный репозиторий, загружаем на Github.

## **3 Выводы**

В ходе выполнения лабораторной работы я освоил процедуры компиляции и сборки программ, написанных на ассемблере NASM.