

# **Отчёт по лабораторной работе №5**

**Дисциплина: Архитектура компьютера**

Соснина Виктория Евгеньевна

# Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выполнение заданий для самостоятельной работы	9
4	Выводы	12
	Список литературы	13

# Список иллюстраций

2.1	Создание текстового файла . . . . .	6
2.2	Открытие файла с помощью gedit . . . . .	6
2.3	Ввод текста программы . . . . .	7
2.4	Создание объектного файла hello.o, проверка . . . . .	7
2.5	Компиляция в obj.o и создания файла листинга, проверка . . . . .	7
2.6	Передача объектного файла hello.o на обработку компоновщику .	8
2.7	Получение исполняемого файла main . . . . .	8
2.8	Запуск на выполнение исполняемого файла . . . . .	8
3.1	Копирование текстового файла . . . . .	9
3.2	Редактирование текстового файла . . . . .	9
3.3	Получение объектного файла . . . . .	10
3.4	Компановка объектного файла . . . . .	10
3.5	Запуск исполняемого файла . . . . .	10
3.6	Загрузка на github . . . . .	10
3.7	Загрузка на github . . . . .	11

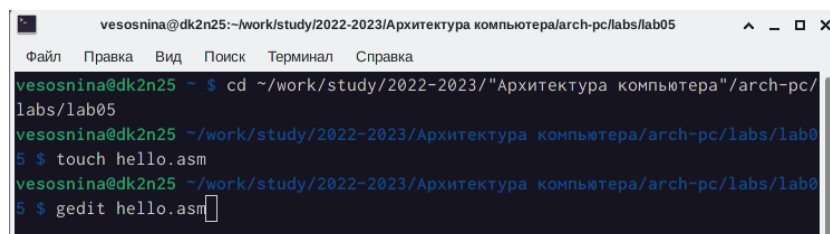
## Список таблиц

# 1 Цель работы

Цель данной работы — изучение основ процедур компиляции и сборки программ, написанных на ассемблере NASM.

## 2 Выполнение лабораторной работы

Перейдем каталог текущей лабораторной работы. Используя команду `touch`, создадим текстовый файл `hello.asm` и откроем его с помощью текстового редактора `gedit`.



```
vesosnina@dk2n25:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05
vesosnina@dk2n25 ~ $ cd ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05
vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ touch hello.asm
vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ gedit hello.asm
```

Рис. 2.1: Создание текстового файла

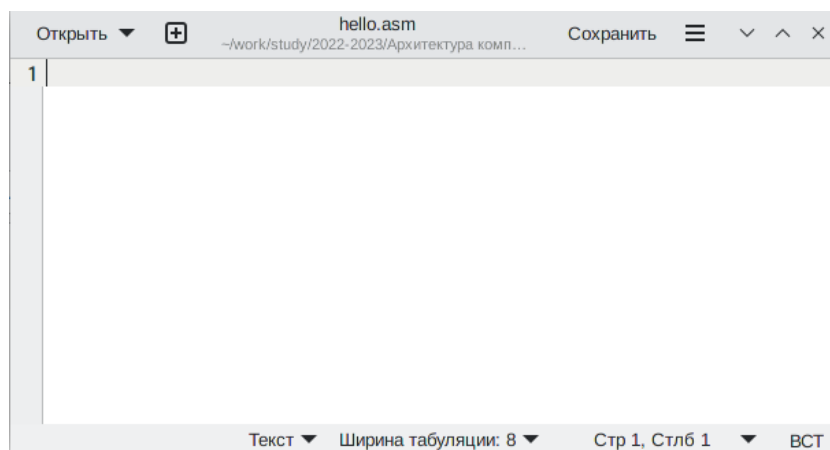
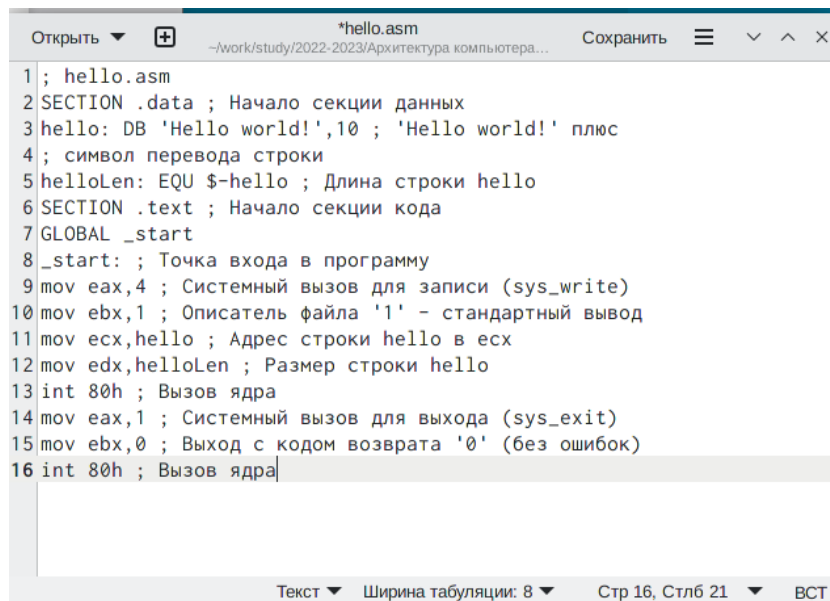


Рис. 2.2: Открытие файла с помощью `gedit`

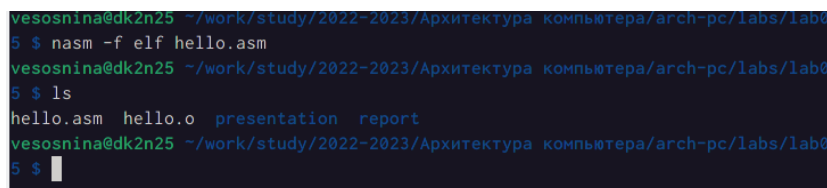
Введем программу, которая выведет текст “Hello world!”.



```
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
```

Рис. 2.3: Ввод текста программы

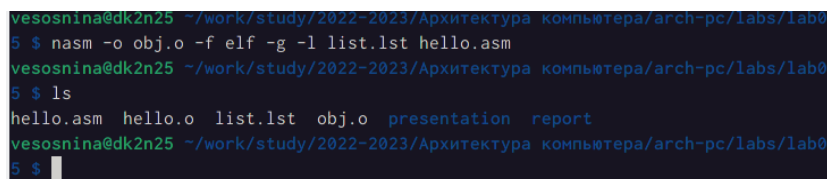
Создадим объектный код, то есть скомпилируем текст программы, которую мы написали ранее. Для этого используем команду `nasm -f elf hello.asm`



```
vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab0
5 $ nasm -f elf hello.asm
vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab0
5 $ ls
hello.asm hello.o presentation report
vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab0
5 $
```

Рис. 2.4: Создание объектного файла hello.o, проверка

Скомпилируем файл `hello.asm` в `obj.o` и создадим файл листинга `list.lst`



```
vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab0
5 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab0
5 $ ls
hello.asm hello.o list.lst obj.o presentation report
vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab0
5 $
```

Рис. 2.5: Компиляция в obj.o и создания файла листинга, проверка

Передадим объектный файл `hello.o` на обработку компоновщику. Получим при этом исполняемый файл `hello`. Введем для этого команду `ld -m elf_i386 hello.o -o hello`. Проверим правильность выполненных действий командой `ls`.

```

vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab0
5 $ ld -m elf_i386 hello.o -o hello
vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab0
5 $ ls
hello hello.asm hello.o list.lst obj.o presentation report
vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab0
5 $

```

Рис. 2.6: Передача объектного файла hello.o на обработку компоновщику

Теперь передадим объектный файл obj.o на обработку компоновщику. Получим при этом исполняемый файл main. Введем для этого команду `ld -m elf_i386 obj.o -o main`. Проверим правильность выполненных действий командой `ls`.

```

vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab0
5 $ ld -m elf_i386 obj.o -o main
vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab0
5 $ ls
hello hello.asm hello.o list.lst main obj.o presentation report
vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab0
5 $

```

Рис. 2.7: Получение исполняемого файла main

Запустим созданный ранее исполняемый файл hello. Введем для этого команду `./hello`

```

vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab0
5 $ ./hello
Hello world!
vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab0
5 $

```

Рис. 2.8: Запуск на выполнение исполняемого файла

Выполнение данных заданий позволило мне ознакомиться с процедурой компиляции и сборки программ на ассемблере NASM.



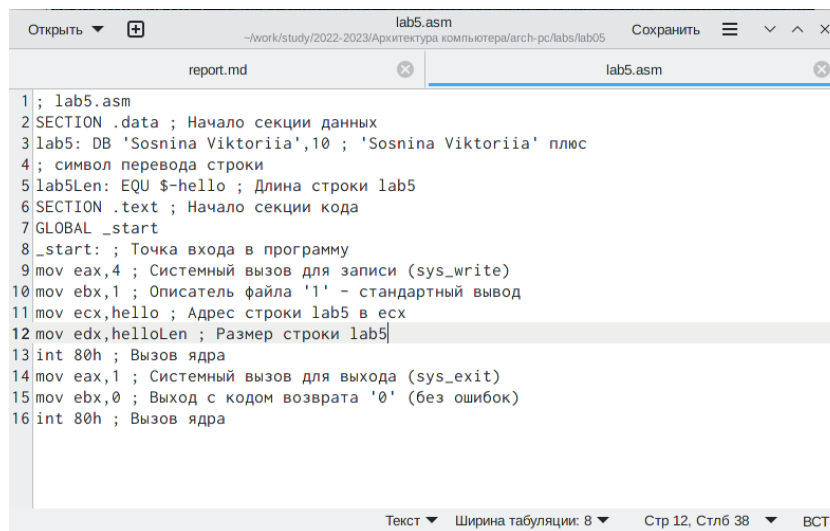
### 3 Выполнение заданий для самостоятельной работы

Скопируем файл `hello.asm` и зададим ему имя `lab5.asm`

```
vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab0
5 $ cp hello.asm lab5.asm
vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab0
5 $ ls
hello      hello.o    list.lst   obj.o      report
hello.asm  lab5.asm  main       presentation
vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab0
5 $
```

Рис. 3.1: Копирование текстового файла

Отредактируем файл так, чтобы программа выводила имя и фамилию.



```
Открыть  +  lab5.asm  Сохранить  ≡  ▾  ^  ×
~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05
report.md  lab5.asm
1 ; lab5.asm
2 SECTION .data ; Начало секции данных
3 lab5: DB 'Sosnina Viktoriia',10 ; 'Sosnina Viktoriia' плюс
4 ; символ перевода строки
5 lab5Len: EQU $-hello ; Длина строки lab5
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 lab5 в ecx
12 mov edx,helloLen ; Размер строки lab5
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
Текст  Ширина табуляции: 8  Стр 12, Стлб 38  ВСТ
```

Рис. 3.2: Редактирование текстового файла

Оттранслируем `lab5.asm` в объектный файл и создадим файл листинга. Испол-

зуем для этого команду `nasm -o obj_lab5.o elf -g -l list_lab5.lst lab5.asm`.

```
vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05
$ nasm -o obj_lab5.o -f elf -g -l list_lab5.lst lab5.asm
vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05
$ ls
lab5.asm  list_lab5.lst  obj_lab5.o  presentation  report  work_lab05
vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05
$
```

Рис. 3.3: Получение объектного файла

Выполним компоновку полученного объектного файла. Введем для этого команду `ld -m elf_i386 obj_lab5.o -o lab5_uwu`

```
vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05
$ ld -m elf_i386 obj_lab5.o -o lab5_uwu
vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05
$ ls
lab5.asm  lab5_uwu  list_lab5.lst  obj_lab5.o  presentation  report  work_lab05
vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05
$
```

Рис. 3.4: Компоновка объектного файла

Запустим исполняемый файл `lab5_uwu`. Введем для этого команду `./lab5_uwu`

```
vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05
$ ./lab5_uwu
Viktoria Sosnina
vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05
$
```

Рис. 3.5: Запуск исполняемого файла

Загрузим изменения на github.

```
Viktoria Sosnina
vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05
$ git add .
vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05
$ git commit -am 'lab05'
[master 0f17711] lab05
 24 files changed, 205 insertions(+), 119 deletions(-)
 create mode 100644 labs/lab05/individual_work/lab5.asm
 create mode 100755 labs/lab05/individual_work/lab5_uwu
 create mode 100644 labs/lab05/individual_work/list_lab5.lst
 create mode 100644 labs/lab05/individual_work/obj_lab5.o
 create mode 100644 labs/lab05/report/image/5_1.png
 create mode 100644 labs/lab05/report/image/5_10.png
 create mode 100644 labs/lab05/report/image/5_11.png
 create mode 100644 labs/lab05/report/image/5_12.png
```

Рис. 3.6: Загрузка на github

```

create mode 100755 labs/lab05/work_lab05/main
create mode 100644 labs/lab05/work_lab05/obj.o
vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab0
5 $ git push
Перечисление объектов: 38, готово.
Подсчет объектов: 100% (38/38), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (32/32), готово.
Запись объектов: 100% (32/32), 431.48 КиБ | 21.57 МиБ/с, готово.
Всего 32 (изменений 9), повторно использовано 0 (изменений 0), повторно использо
вано пакетов 0
remote: Resolving deltas: 100% (9/9), completed with 3 local objects.
To github.com:ViktoriiaSosnina/study_2022-2023_arh-pc.git
   4a9aff7..0f17711  master -> master
vesosnina@dk2n25 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab0
5 $

```

Рис. 3.7: Загрузка на github

Выполнение заданий для самостоятельной работы позволило мне на практике применить полученные знания по компиляции и сборке программ на ассемблере NASM.

## 4 Выводы

В результате выполнения лабораторной работы я освоила процедуру компиляции и сборки программ, написанных на ассемблере NASM. Полученные знания позволят мне писать более сложные программы в будущем.

## **Список литературы**