

# **Отчёт по лабораторной работе №5**

**Дисциплина: Архитектура компьютера**

Ищенко Ирина Олеговна

# Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выполнение заданий для самостоятельной работы	8
4	Выводы	10

# Список иллюстраций

2.1	Создание каталога . . . . .	6
2.2	Создание файла hello.asm . . . . .	6
2.3	Файл hello.asm . . . . .	6
2.4	Создание объектного файла . . . . .	7
2.5	Компиляция файлов . . . . .	7
2.6	Создание исполняемого файла hello . . . . .	7
2.7	Создание исполняемого файла main . . . . .	7
2.8	Запуск исполняемого файла . . . . .	7
3.1	Файл lab5.asm . . . . .	8
3.2	Компоновка и запуск файла . . . . .	8
3.3	Загрузка на github . . . . .	9

## **Список таблиц**

# 1 Цель работы

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

## 2 Выполнение лабораторной работы

Создаем каталог для работы с программами на языке ассемблера NASM и переходим в него (рис. 2.1).

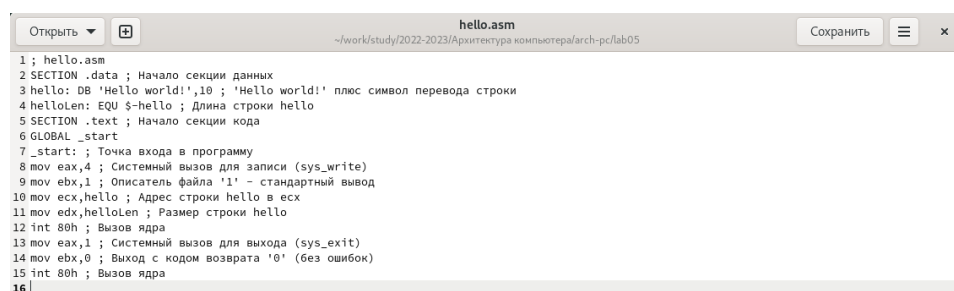
```
[ioithenko@fedora ~]$ mkdir /home/ioithenko/work/study/2022-2023/"Архитектура компьютера"/arch-pc/lab05  
[ioithenko@fedora ~]$ cd /home/ioithenko/work/study/2022-2023/"Архитектура компьютера"/arch-pc/lab05  
[ioithenko@fedora lab05]$
```

Рис. 2.1: Создание каталога

Создаем текстовый файл с именем hello.asm и открываем его в текстовом редакторе (рис. 2.2). Вводим текст (рис. 2.3).

```
[ioithenko@fedora lab05]$ touch hello.asm  
[ioithenko@fedora lab05]$ gedit hello.asm
```

Рис. 2.2: Создание файла hello.asm



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

Рис. 2.3: Файл hello.asm

Преобразуем текст программы в объектный код и проверяем, создался ли объектный файл (рис. 2.4). Созданный файл имеет имя hello.o.

```
[ioithenko@fedora lab05]$ nasm -f elf hello.asm
[ioithenko@fedora lab05]$ ls
hello.asm  hello.o
[ioithenko@fedora lab05]$
```

Рис. 2.4: Создание объектного файла

С помощью команды `nasm -o obj.o -f elf -g -l list.lst hello.asm` скомпилируем исходный файл `hello.asm` в `obj.o`, также будет создан файл листинга `list.lst`. Проверим выполнение команды (рис. 2.5).

```
[ioithenko@fedora lab05]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
[ioithenko@fedora lab05]$ ls
hello.asm  hello.o  list.lst  obj.o
[ioithenko@fedora lab05]$
```

Рис. 2.5: Компиляция файлов

Создадим исполняемый файл `hello` и проведем проверку командой `ls` (рис. 2.6).

```
[ioithenko@fedora lab05]$ ld -m elf_i386 hello.o -o hello
[ioithenko@fedora lab05]$ ls
hello  hello.asm  hello.o  list.lst  obj.o
[ioithenko@fedora lab05]$
```

Рис. 2.6: Создание исполняемого файла `hello`

С помощью команды (рис. 2.7) создадим исполняемый файл с названием `main` из объектного файла `obj.o`.

```
[ioithenko@fedora lab05]$ ld -m elf_i386 obj.o -o main
```

Рис. 2.7: Создание исполняемого файла `main`

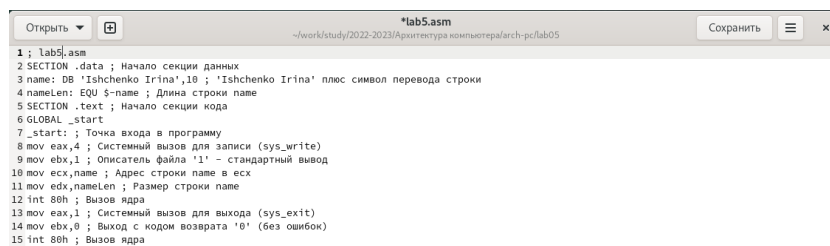
Запустим исполняемый файл (рис. 2.8).

```
[ioithenko@fedora lab05]$ ./hello
Hello world!
[ioithenko@fedora lab05]$
```

Рис. 2.8: Запуск исполняемого файла

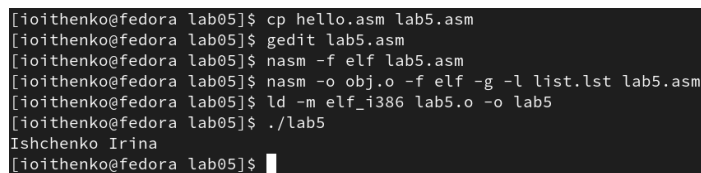
## 3 Выполнение заданий для самостоятельной работы

С помощью команды *cp* создадим копию файла *hello.asm* с именем *lab5.asm*. В текстовом редакторе внесем изменения в текст программы в файле *lab5.asm* (рис. 3.1). Оттранслируем полученный текст программы *lab5.asm* в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл (рис. 3.2).



```
1; lab5.asm
2 SECTION .data ; Начало секции данных
3 name: DB 'Ishchenko Irina',10 ; 'Ishchenko Irina' плюс символ перевода строки
4 nameLen: EQU $-name ; Длина строки name
5 SECTION .text ; Начало секции кода
6 GLOBAL _start
7 _start: ; Точка входа в программу
8 mov eax,4 ; Системный вызов для записи (sys_write)
9 mov ebx,1 ; Описатель файла '1' - стандартный вывод
10 mov ecx,name ; Адрес строки name в ecx
11 mov edx,nameLen ; Размер строки name
12 int 80h ; Вызов ядра
13 mov eax,1 ; Системный вызов для выхода (sys_exit)
14 mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
15 int 80h ; Вызов ядра
```

Рис. 3.1: Файл lab5.asm



```
[ioithenko@fedora lab05]$ cp hello.asm lab5.asm
[ioithenko@fedora lab05]$ gedit lab5.asm
[ioithenko@fedora lab05]$ nasm -f elf lab5.asm
[ioithenko@fedora lab05]$ nasm -o obj.o -f elf -g -l list.lst lab5.asm
[ioithenko@fedora lab05]$ ld -m elf_i386 lab5.o -o lab5
[ioithenko@fedora lab05]$ ./lab5
Ishchenko Irina
[ioithenko@fedora lab05]$
```

Рис. 3.2: Компоновка и запуск файла

Скопируем файлы *hello.asm* и *lab5.asm* в локальный репозиторий в соответствующий каталог. Загрузим файлы на Github (рис. 3.3).



```

[ioithenko@fedora lab05]$ cp hello.asm lab5.asm ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/labs/lab05/
[ioithenko@fedora lab05]$ cd ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/labs/lab05
[ioithenko@fedora lab05]$ git add .
[ioithenko@fedora lab05]$ git commit -am 'feat(main): add files lab-5'
[master 6505c2b] feat(main): add files lab-5
2 files changed, 31 insertions(+)
 create mode 100644 labs/lab05/hello.asm
 create mode 100644 labs/lab05/lab5.asm
[ioithenko@fedora lab05]$ git push
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 1.04 КиБ | 1.04 МБ/с, готово.
Всего 6 (изменений 3), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To github.com:ioithenko/study_2022-2023_arh-pc.git
   d9049a2..6505c2b  master -> master
[ioithenko@fedora lab05]$

```

Рис. 3.3: Загрузка на github

## **4 Выводы**

В ходе выполнения лабораторной работы я освоила процедуры компиляции и сборки программ, написанных на языке ассемблера NASM.