

# **Отчёт по лабораторной работе №5**

**Лабораторная работа №5. Создание и процесс обработки программ на языке ассемблера NASM**

Боровиков Даниил Александрович

# Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Выводы	9

## Список иллюстраций

2.1	Создание каталога для работы с программами . . . . .	5
2.2	Переход в созданный каталог . . . . .	5
2.3	Создание текстового файла . . . . .	5
2.4	Открытие созданного ранее текстового файла . . . . .	6
2.5	Ввод текста программы . . . . .	6
2.6	Компиляция программы . . . . .	6
2.7	Компиляция исходного файла hello.asm в obj.o . . . . .	6
2.8	Создание исполняемого файла hello . . . . .	7
2.9	Запуск исполняемого файла . . . . .	7
2.10	Копирование файла . . . . .	7
2.11	Изменение вывода программы . . . . .	7
2.12	трансляция компоновка и запуск именованного файла . . . . .	8
2.13	Копирование файлов в локальный репозиторий и загрузка на git hub	8

# 1 Цель работы

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

## 2 Выполнение лабораторной работы

Создадим каталог для работы с программами на языке ассемблера NASM командой: `mkdir ~/work/arch-pc/lab05`(рис. 2.1)

```
user@daborovikov:~$ mkdir ~/work/arch-pc/  
user@daborovikov:~$ mkdir ~/work/arch-pc/lab05  
user@daborovikov:~$ ls ~/work/arch-pc/  
lab05  
user@daborovikov:~$
```

Рис. 2.1: Создание каталога для работы с программами

Перейдем в созданный каталог(рис. 2.2)

```
user@daborovikov:~$ cd ~/work/arch-pc/lab05  
user@daborovikov:~/work/arch-pc/lab05$
```

Рис. 2.2: Переход в созданный каталог

Создадим текстовый файл с именем `hello.asm` командой: `touch hello.asm`(рис. 2.3)

```
user@daborovikov:~/work/arch-pc/lab05$ touch hello.asm  
user@daborovikov:~/work/arch-pc/lab05$ ls  
hello.asm  
user@daborovikov:~/work/arch-pc/lab05$
```

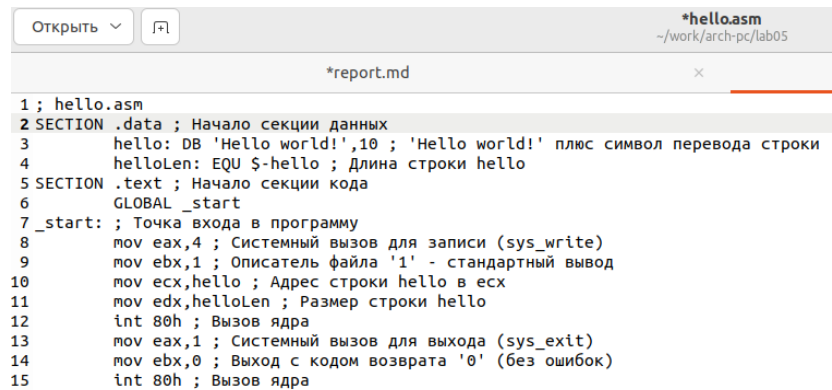
Рис. 2.3: Создание текстового файла

Откроем этот файл с помощью текстового редактора `gedit`: `gedit hello.asm`(рис. 2.4)

```
user@daborovikov:~/work/arch-pc/lab05$ gedit hello.asm
user@daborovikov:~/work/arch-pc/lab05$
```

Рис. 2.4: Открытие созданного ранее текстового файла

Введем файл текст программы(рис. 2.5)



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

*report.md
x

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
```

Рис. 2.5: Ввод текста программы

Скомпилируем программу(рис. 2.6)

```
user@daborovikov:~/work/arch-pc/lab05$ nasm -f elf hello.asm
user@daborovikov:~/work/arch-pc/lab05$ ls
hello.asm  hello.o
user@daborovikov:~/work/arch-pc/lab05$
```

Рис. 2.6: Компиляция программы

Выполним следующую команду: `nasm -o obj.o -f elf -g -l list.lst hello.asm`(рис. 2.7)

```
user@daborovikov:~/work/arch-pc/lab05$ nasm -o obj.o -f elf -g -l list.lst hello.asm
user@daborovikov:~/work/arch-pc/lab05$ ls
hello.asm  hello.o  list.lst  obj.o
user@daborovikov:~/work/arch-pc/lab05$
```

Рис. 2.7: Компиляция исходного файла hello.asm в obj.o

Передадим объектный файл на обработку компоновщику: `ld -m elf_i386 hello.o -o hellom`(рис. 2.8)

```

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

```

Рис. 2.8: Создание исполняемого файла hello

Запустим созданный исполняемый файл, находящийся в текущем каталоге командой: `./hello`(рис. 2.9)

```

user@daborovikov:~/work/arch-pc/lab05$ ./hello
Hello world!
user@daborovikov:~/work/arch-pc/lab05$

```

Рис. 2.9: Запуск исполняемого файла

### Самостоятельная работа

В каталоге `~/work/arch-pc/lab05` с помощью команды `cp` создадим копию файла `hello.asm` с именем `lab5.asm`(рис. 2.10)

```

user@daborovikov:~/work/arch-pc/lab05$ cp ~/work/arch-pc/lab05/hello.asm ~/work/arch-pc/lab05/lab5.asm
user@daborovikov:~/work/arch-pc/lab05$ ls
hello hello.asm hello.o lab5.asm list.lst obj.o
user@daborovikov:~/work/arch-pc/lab05$

```

Рис. 2.10: Копирование файла

С помощью текстового редактора `gedit` внесем изменения в текст программы в файле `lab5.asm` так, чтобы вместо `Hello world!` на экран выводилась строка с фамилией и именем.(рис. 2.11)

```

*lab5.asm
~/work/arch-pc/lab05

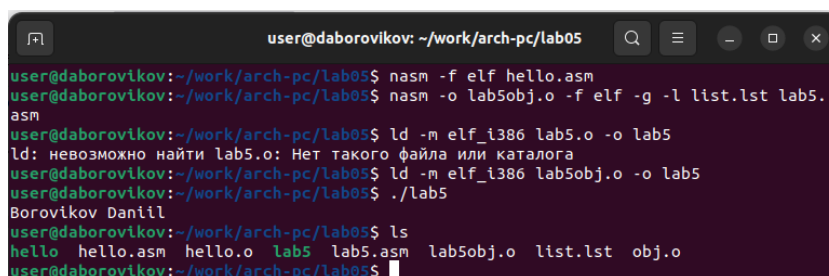
*report.md
x

1; hello.asm
2 SECTION .data ; Начало секции данных
3     hello: DB 'Borovikov Daniil',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

```

Рис. 2.11: Изменение вывода программы

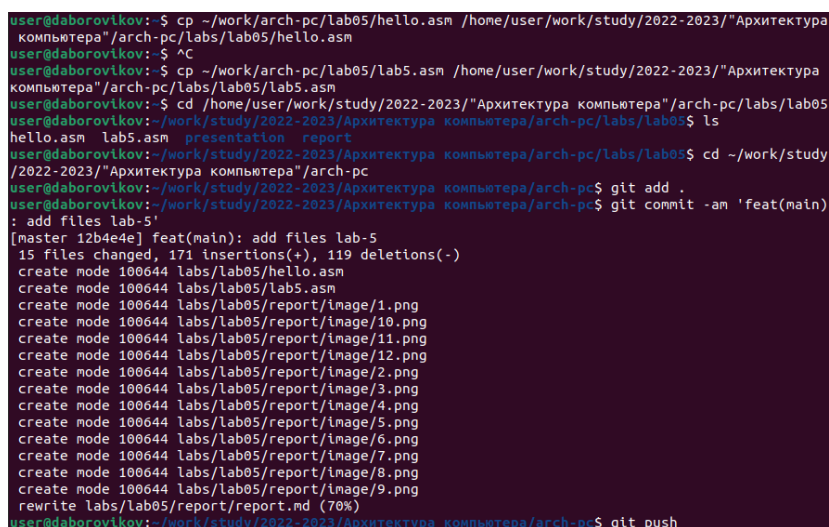
Оттранслируем полученный текст программы lab5.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл.(рис. 2.12)

A terminal window with a dark background and light text. The prompt is 'user@daborovikov: ~/work/arch-pc/lab05'. The user enters several commands: 'nasm -f elf hello.asm', 'nasm -o lab5obj.o -f elf -g -l list.lst lab5.asm', 'ld -m elf\_i386 lab5.o -o lab5', and another 'ld' command that fails with an error. Finally, the user runs 'ls' which shows the files 'hello hello.asm hello.o lab5 lab5.asm lab5obj.o list.lst obj.o'.

```
user@daborovikov: ~/work/arch-pc/lab05
user@daborovikov:~/work/arch-pc/lab05$ nasm -f elf hello.asm
user@daborovikov:~/work/arch-pc/lab05$ nasm -o lab5obj.o -f elf -g -l list.lst lab5.asm
user@daborovikov:~/work/arch-pc/lab05$ ld -m elf_i386 lab5.o -o lab5
ld: невозможно найти lab5.o: Нет такого файла или каталога
user@daborovikov:~/work/arch-pc/lab05$ ld -m elf_i386 lab5obj.o -o lab5
user@daborovikov:~/work/arch-pc/lab05$ ./lab5
Borovikov Danil
user@daborovikov:~/work/arch-pc/lab05$ ls
hello hello.asm hello.o lab5 lab5.asm lab5obj.o list.lst obj.o
user@daborovikov:~/work/arch-pc/lab05$
```

Рис. 2.12: трансляция компоновка и запуск именного файла

Скопируем файлы hello.asm и lab5.asm в локальный репозиторий в каталог ~/work/study/2022-2023/“Архитектура компьютера”/arch- pc/labs/lab05/. Загрузим файлы на Github.(рис. 2.13)

A terminal window showing the process of copying files and committing them to a git repository. The user copies 'hello.asm' and 'lab5.asm' from the lab directory to a study directory. Then, they run 'ls' to confirm the files are there. Finally, they run 'git add .' to stage the files, 'git commit -am 'feat(main): add files lab-5'' to commit them, and 'git push' to upload to the remote repository. The output shows the commit details and the files added.

```
user@daborovikov: $ cp ~/work/arch-pc/lab05/hello.asm /home/user/work/study/2022-2023/“Архитектура
компьютера”/arch-pc/labs/lab05/hello.asm
user@daborovikov: $ ^C
user@daborovikov: $ cp ~/work/arch-pc/lab05/lab5.asm /home/user/work/study/2022-2023/“Архитектура
компьютера”/arch-pc/labs/lab05/lab5.asm
user@daborovikov: $ cd /home/user/work/study/2022-2023/“Архитектура компьютера”/arch-pc/labs/lab05
user@daborovikov:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05$ ls
hello.asm lab5.asm presentation report
user@daborovikov:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05$ cd ~/work/study
/2022-2023/“Архитектура компьютера”/arch-pc
user@daborovikov:~/work/study/2022-2023/Архитектура компьютера/arch-pc$ git add .
user@daborovikov:~/work/study/2022-2023/Архитектура компьютера/arch-pc$ git commit -am 'feat(main)
: add files lab-5'
[master 12b4e4e] feat(main): add files lab-5
15 files changed, 171 insertions(+), 119 deletions(-)
create mode 100644 labs/lab05/hello.asm
create mode 100644 labs/lab05/lab5.asm
create mode 100644 labs/lab05/report/image/1.png
create mode 100644 labs/lab05/report/image/10.png
create mode 100644 labs/lab05/report/image/11.png
create mode 100644 labs/lab05/report/image/12.png
create mode 100644 labs/lab05/report/image/2.png
create mode 100644 labs/lab05/report/image/3.png
create mode 100644 labs/lab05/report/image/4.png
create mode 100644 labs/lab05/report/image/5.png
create mode 100644 labs/lab05/report/image/6.png
create mode 100644 labs/lab05/report/image/7.png
create mode 100644 labs/lab05/report/image/8.png
create mode 100644 labs/lab05/report/image/9.png
rewrite labs/lab05/report/report.md (70%)
user@daborovikov:~/work/study/2022-2023/Архитектура компьютера/arch-pc$ git push
```

Рис. 2.13: Копирование файлов в локальный репозиторий и загрузка на git hub

Ссылка на github: [https://github.com/daBorovikov/study\\_2022-2023\\_arh-pc](https://github.com/daBorovikov/study_2022-2023_arh-pc)



## **3 Выводы**

В ходе лабораторной работы мы освоили процедуры компиляции и сборки программ, написанных на ассемблере NASM, и написали собственную программу “hello world”