

# **Отчёт по лабораторной работе №5**

**Простейший вариант**

Борунов Семён Сергеевич

# Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	9

## Список иллюстраций

2.1	создание файла . . . . .	6
2.2	написание кода . . . . .	6
2.3	трансляция . . . . .	6
2.4	создание исполняемого файла . . . . .	7
2.5	создание другого исполняемого файла . . . . .	7
2.6	запуск исполняемого файла . . . . .	7
2.7	создание lab05.asm . . . . .	7
2.8	редактирование файла . . . . .	8
2.9	создание объектного файла, компиляция и проверка работы . . . .	8

## Список таблиц

# 1 Цель работы

Освоение сборки программ, написанных на NASM.

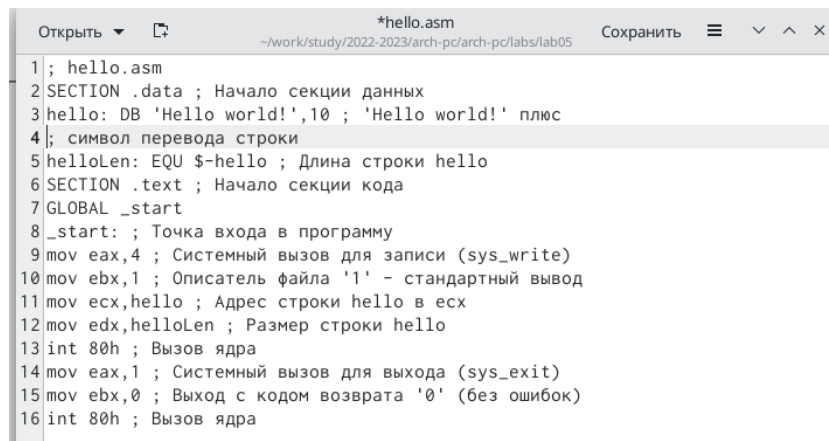
## 2 Выполнение лабораторной работы

Откроем нужную папку, создадим файл и откроем его в gedit (рис. 2.1)

```
ssborunov@dk2n24 ~ $ cd work/study/2022-2023/arch-pc/arch-pc/
ssborunov@dk2n24 ~/work/study/2022-2023/arch-pc/arch-pc $ cd labs/lab05
ssborunov@dk2n24 ~/work/study/2022-2023/arch-pc/arch-pc/labs/lab05 $ touch hello.asm
ssborunov@dk2n24 ~/work/study/2022-2023/arch-pc/arch-pc/labs/lab05 $ gedit hello.asm
```

Рис. 2.1: создание файла

Запишем код на языке NASM в файл с помощью gedit (рис. 2.2)



```
Открыть ▾  *hello.asm  Сохранить  ▸  ▾  ▹  ✕
~/work/study/2022-2023/arch-pc/arch-pc/labs/lab05
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 ехх
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. 2.2: написание кода

Протранслируем файл hello.asm и получим объектный файл hello.o (рис. 2.3)

```
ssborunov@dk2n24 ~/work/study/2022-2023/arch-pc/arch-pc/labs/lab05 $ nasm -f elf hello.asm
ssborunov@dk2n24 ~/work/study/2022-2023/arch-pc/arch-pc/labs/lab05 $ ls
hello.asm hello.o presentation report
ssborunov@dk2n24 ~/work/study/2022-2023/arch-pc/arch-pc/labs/lab05 $
```

Рис. 2.3: трансляция

Далее мы получаем объектный файл `obj.o` (он будет в формате `elf`) и файл листинга `list.lst`. Затем скомпилируем наш исполняемый файл `hello` (рис. 2.4)

```
ssborunov@dk2n24 ~/work/study/2022-2023/arch-pc/arch-pc/labs/lab05 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
ssborunov@dk2n24 ~/work/study/2022-2023/arch-pc/arch-pc/labs/lab05 $ ls
hello.asm  hello.o  list.lst  obj.o  presentation  report
ssborunov@dk2n24 ~/work/study/2022-2023/arch-pc/arch-pc/labs/lab05 $ ld -m elf_i386 hello.o -o hello
ssborunov@dk2n24 ~/work/study/2022-2023/arch-pc/arch-pc/labs/lab05 $ ls
hello  hello.asm  hello.o  list.lst  obj.o  presentation  report
ssborunov@dk2n24 ~/work/study/2022-2023/arch-pc/arch-pc/labs/lab05 $
```

Рис. 2.4: создание исполняемого файла

Тут мы создаем исполняемый файл `main` из объектного файла `obj.o`. Делать этот файл будет то же, что и файл `hello`. (рис. 2.5)

```
ssborunov@dk2n24 ~/work/study/2022-2023/arch-pc/arch-pc/labs/lab05 $ ld -m elf_i386 obj.o -o main
ssborunov@dk2n24 ~/work/study/2022-2023/arch-pc/arch-pc/labs/lab05 $ ls
hello  hello.asm  hello.o  list.lst  main  obj.o  presentation  report
ssborunov@dk2n24 ~/work/study/2022-2023/arch-pc/arch-pc/labs/lab05 $
```

Рис. 2.5: создание другого исполняемого файла

Запустим файл `hello` (рис. 2.6)

```
ssborunov@dk2n24 ~/work/study/2022-2023/arch-pc/arch-pc/labs/lab05 $ ./hello
Hello world!
```

Рис. 2.6: запуск исполняемого файла

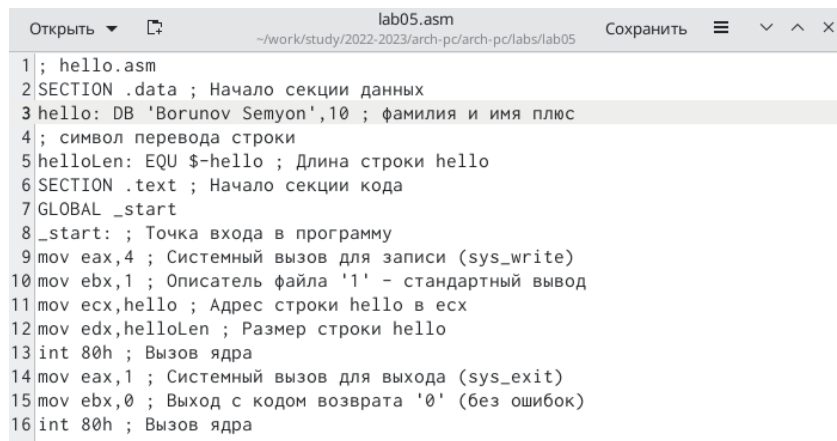
#Задания для самостоятельной работы

Скопируем файл `hello.asm` как `lab05.asm` и откроем его через `gedit`. (рис. 2.7)

```
ssborunov@dk2n24 ~/work/study/2022-2023/arch-pc/arch-pc/labs/lab05 $ cp hello.asm lab05.asm
ssborunov@dk2n24 ~/work/study/2022-2023/arch-pc/arch-pc/labs/lab05 $ ls
hello  hello.asm  hello.o  lab05.asm  list.lst  main  obj.o  presentation  report
ssborunov@dk2n24 ~/work/study/2022-2023/arch-pc/arch-pc/labs/lab05 $ gedit lab05.asm
```

Рис. 2.7: создание `lab05.asm`

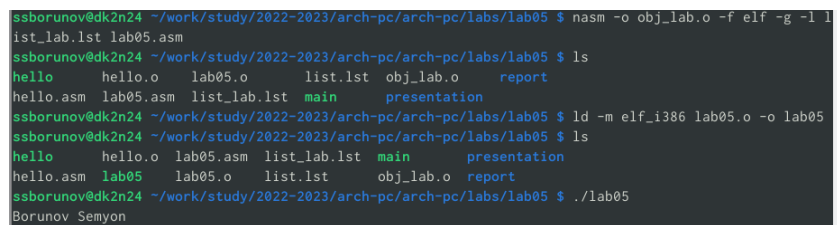
Отредактируем файл. (рис. 2.8)



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

Рис. 2.8: редактирование файла

Далее создадим объектный файл с именем `obj_lab.o` в формате `elf` и листинг `list_lab.lst`. Затем скомпилируем `obj_lab.o` под именем `lab05` и запустим. В выводе именно то, что мы хотели. (рис. 2.9)



```
ssborunov@dk2n24 ~/work/study/2022-2023/arch-pc/arch-pc/labs/lab05 $ nasm -o obj_lab.o -f elf -g -l 1
ist_lab.lst lab05.asm
ssborunov@dk2n24 ~/work/study/2022-2023/arch-pc/arch-pc/labs/lab05 $ ls
hello    hello.o    lab05.o    list.lst  obj_lab.o  report
hello.asm lab05.asm  list_lab.lst  main      presentation
ssborunov@dk2n24 ~/work/study/2022-2023/arch-pc/arch-pc/labs/lab05 $ ld -m elf_i386 lab05.o -o lab05
ssborunov@dk2n24 ~/work/study/2022-2023/arch-pc/arch-pc/labs/lab05 $ ls
hello    hello.o    lab05.asm  list_lab.lst  main      presentation
hello.asm lab05     lab05.o    list.lst     obj_lab.o  report
ssborunov@dk2n24 ~/work/study/2022-2023/arch-pc/arch-pc/labs/lab05 $ ./lab05
Borunov Semyon
```

Рис. 2.9: создание объектного файла, компиляция и проверка работы



## 3 Выводы

Были получены навыки по сборке кода, написанного с помощью NASM, в исполняющий файл.