

Отчет по лабораторной работе №4

дисциплина: Архитектура компьютера

Сергеев Даниил Олегович

Содержание

1	Цель лабораторной работы	4
2	Задание	5
3	Ход выполнения лабораторной работы	6
4	Ход выполнения заданий для самостоятельной работы	10
5	Вывод	14

Список иллюстраций

3.1	Создание файла <code>hello.asm</code>	7
3.2	Вид из программы <code>gedit</code>	8
3.3	Файлы успешно созданы	9
3.4	Файл работает успешно	9
4.1	Вид отчета лабораторной работы №2 в редакторе <code>gedit</code> . .	10
4.2	Вид в <code>gedit</code>	11
4.3	Проверка компиляции файла	11
4.4	Файл работает успешно	12
4.5	Копирование файлов	12
4.6	Отправляем файлы на github (1)	12
4.7	Отправляем файлы на github (2)	13

1 Цель лабораторной работы

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

2 Задание

Необходимо создать файл `hello.asm`, открыть его с помощью текстового редактора и ввести текст из примера. Далее необходимо скомпилировать измененный файл и передать объектный файл на обработку компоновщику, проверить корректность введенных команд и запустить исполняемый файл.

3 Ход выполнения лабораторной работы

1. Открываем терминал, создаем каталоги `~/work/arch-pc/lab04`, переходим в них. Создаем файл `hello.asm`, проверяем и открываем его с помощью `gedit`.

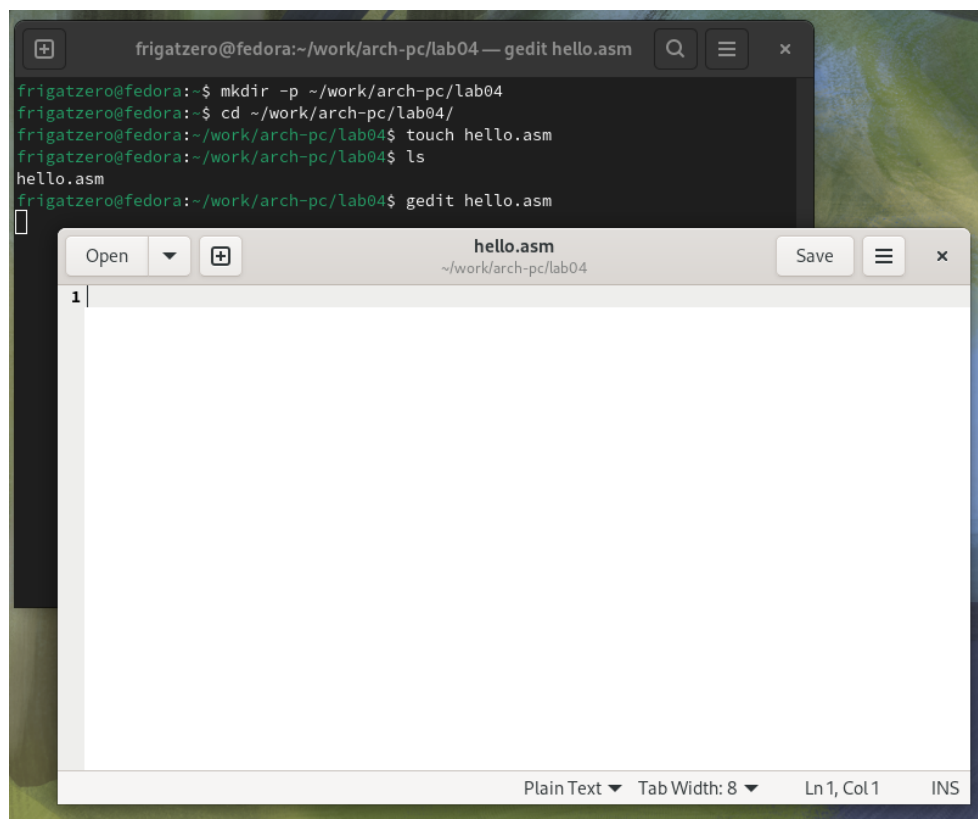


Рис. 3.1: Создание файла `hello.asm`

2. Вводим текст из примера и сохраняем.



```
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
7 SECTION .text ; Начало секции кода
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 ecx
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
```

Рис. 3.2: Вид из программы gedit

3. Компилируем файл `hello.asm` сначала с помощью упрощенной команды `nasm -f elf hello.asm`, а потом с помощью расширенной: `nasm -o obj.o -f elf -g -l list.lst hello.asm`. В результате получим три файла: `obj.o`, `hello.o` и файл листинга `list.lst`. Компилируем полученные файлы с помощью указанных команд. Во втором случае файл получил имя `main`, его объектный файл назывался `obj.o`.


```

frigatzero@fedora:~/work/arch-pc/lab04$
frigatzero@fedora:~/work/arch-pc/lab04$ ls
hello.asm
frigatzero@fedora:~/work/arch-pc/lab04$ nasm -f elf hello.asm
frigatzero@fedora:~/work/arch-pc/lab04$ ls
hello.asm  hello.o
frigatzero@fedora:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst hello.asm
frigatzero@fedora:~/work/arch-pc/lab04$ ls
hello.asm  hello.o  list.lst  obj.o
frigatzero@fedora:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
frigatzero@fedora:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  obj.o
frigatzero@fedora:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
frigatzero@fedora:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
frigatzero@fedora:~/work/arch-pc/lab04$

```

Рис. 3.3: Файлы успешно созданы

4. Запускаем файл.

```

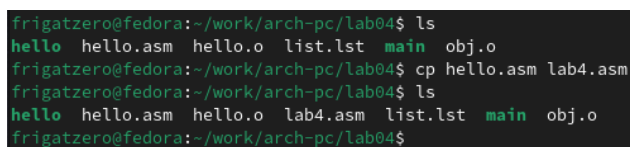
frigatzero@fedora:~/work/arch-pc/lab04$ ./hello
Hello world!
frigatzero@fedora:~/work/arch-pc/lab04$

```

Рис. 3.4: Файл работает успешно

4 Ход выполнения заданий для самостоятельной работы

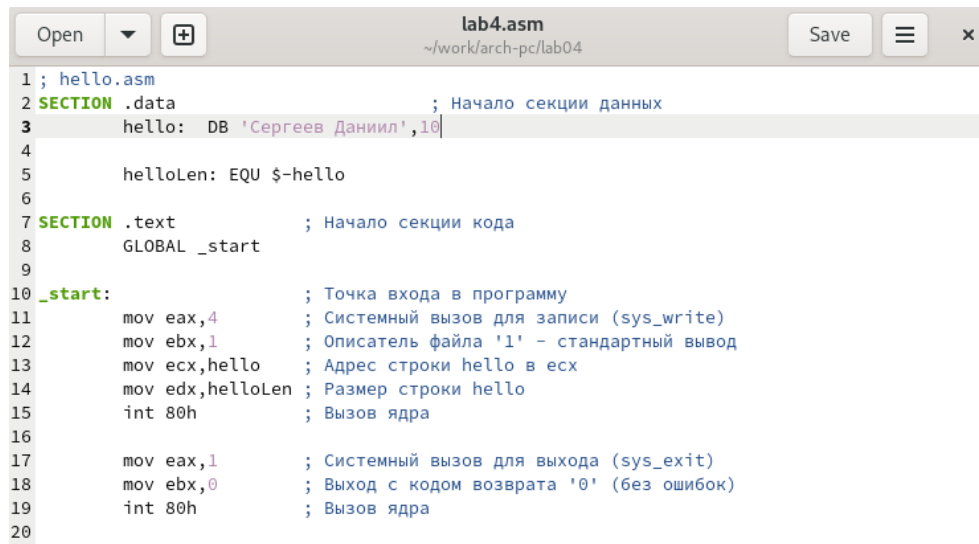
5. В текущем каталоге создаем копию файла `hello.asm` с именем `lab4.asm`.



```
frigatzero@fedora:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o list.lst main obj.o
frigatzero@fedora:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
frigatzero@fedora:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4.asm list.lst main obj.o
frigatzero@fedora:~/work/arch-pc/lab04$
```

Рис. 4.1: Вид отчета лабораторной работы №2 в редакторе gedit

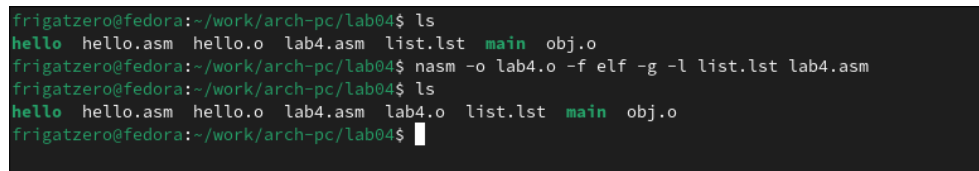
6. Открываем файл с помощью `gedit`. Делаем так, чтобы вместо `Hello world!` на экран выводились фамилия и имя.



```
1 ; hello.asm
2 SECTION .data ; Начало секции данных
3 hello: DB 'Сергеев Даниил',10
4
5 helloLen: EQU $-hello
6
7 SECTION .text ; Начало секции кода
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 ecx
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
```

Рис. 4.2: Вид в gedit

7. Оттранслируем (создаём объектный файл) lab4.asm.



```
frigatzero@fedora:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4.asm list.lst main obj.o
frigatzero@fedora:~/work/arch-pc/lab04$ nasm -o lab4.o -f elf -g -l list.lst lab4.asm
frigatzero@fedora:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4.asm lab4.o list.lst main obj.o
frigatzero@fedora:~/work/arch-pc/lab04$
```

Рис. 4.3: Проверка компиляции файла

8. Компонируем файл, используя соответствующее имя, запускаем файл.

```

frigatzero@fedora:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4.asm list.lst main obj.o
frigatzero@fedora:~/work/arch-pc/lab04$ nasm -o lab4.o -f elf -g -l list.lst lab4.asm
frigatzero@fedora:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4.asm lab4.o list.lst main obj.o
frigatzero@fedora:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
frigatzero@fedora:~/work/arch-pc/lab04$ ./lab4
Сергеев Даниил
frigatzero@fedora:~/work/arch-pc/lab04$

```

Рис. 4.4: Файл работает успешно

9. Копируем файлы `hello.asm` и `lab4.asm` в локальный репозиторий.

```

frigatzero@fedora:~/work/arch-pc/lab04$ cp hello.asm lab4.asm ~/work/
2024-2025/ arch-pc/
frigatzero@fedora:~/work/arch-pc/lab04$ cp hello.asm lab4.asm ~/work/2024-2025/Архитектура\ компьютера/arch-pc/labs/lab04/
frigatzero@fedora:~/work/arch-pc/lab04$ ls ~/work/2024-2025/Архитектура\ компьютера/arch-pc/labs/lab04/
hello.asm lab4.asm report

```

Рис. 4.5: Копирование файлов

10. Загружаем файлы на github.

```

frigatzero@fedora:~$ cd work/2024-2025/Архитектура\ компьютера/arch-pc/
frigatzero@fedora:~/work/2024-2025/Архитектура\ компьютера/arch-pc$ git add .
frigatzero@fedora:~/work/2024-2025/Архитектура\ компьютера/arch-pc$ git commit -m "feat(main): add lab04"
[master a830f17] feat(main): add lab04
27 files changed, 4286 insertions(+)
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/lab4.asm
create mode 100644 labs/lab04/report/Makefile

```

Рис. 4.6: Отправляем файлы на github (1)

```
create mode 100644 labs/lab04/report/pandoc/filters/pandocxnos/pandocattributes.py
create mode 100644 labs/lab04/report/report.md
frigmatzero@fedora:~/work/2024-2025/Архитектура компьютера/arch-pc$ git push
Enumerating objects: 40, done.
Counting objects: 100% (40/40), done.
Delta compression using up to 2 threads
Compressing objects: 100% (34/34), done.
Writing objects: 100% (38/38), 1.36 MiB | 2.29 MiB/s, done.
Total 38 (delta 6), reused 15 (delta 3), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (6/6), completed with 2 local objects.
To github.com:FrigatZero/study_2024-2025_arch-pc.git
   138bec3..a830f17  master -> master
frigmatzero@fedora:~/work/2024-2025/Архитектура компьютера/arch-pc$
```

Рис. 4.7: Отправляем файлы на github (2)

5 Вывод

После выполнения заданий лабораторной работы и заданий для самостоятельной работы я освоил процедуру компиляции и компоновки (сборки) программ, написанных на ассемблере NASM, через командную строку.