

Отчёт по лабораторной работе №4

**Создание и процесс обработки программ на языке ассемблера
NASM**

Новикова Анастасия Андреевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Создание программы Hello world!	7
3.2	Работа с транслятором NASM	8
3.3	Работа с расширенным синтаксисом командной строки NASM . . .	8
3.4	Работа с компоновщиком LD	9
3.5	Запуск исполняемого файла	9
3.6	Задание для самостоятельной работы	10
4	Выводы	12

Список иллюстраций

3.1	Создание каталогов и переход в них	7
3.2	Создание пустого текстового файл и его открытие	7
3.3	Заполнение файла для программы вывода	8
3.4	Компиляция текста программы	8
3.5	Преобразование файла hello.asm в obj.o	9
3.6	Передача объектного файла на обработку компоновщику	9
3.7	Передача объектного файла на обработку компоновщику	9
3.8	Запуск исполняемого файла	9
3.9	Создание копии файла	10
3.10	Редактирование файла	10
3.11	Преобразование файла lab4.asm в файл lab4.o	10
3.12	Передача объектного файла на обработку компоновщику	11
3.13	Запуск исполняемого файла lab4	11
3.14	Копирование файлов lab4.asm и hello.asm	11

Список таблиц

1 Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM

2 Задание

Написать программу “Hello, world!” и программу, выводящую имя и фамилию.

3 Выполнение лабораторной работы

3.1 Создание программы Hello world!

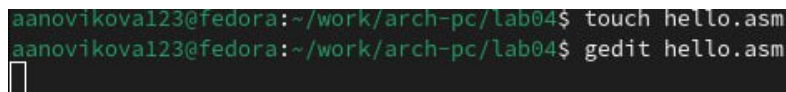
Создаю каталог для работы с программами на языке ассемблера NASM. Перехожу в созданный каталог. (рис. 3.1).



```
aanovikova123@fedora:~$ mkdir -p ~/work/arch-pc/lab04
aanovikova123@fedora:~$ cd ~/work/arch-pc/lab04
aanovikova123@fedora:~/work/arch-pc/lab04$
```

Рис. 3.1: Создание каталогов и переход в них

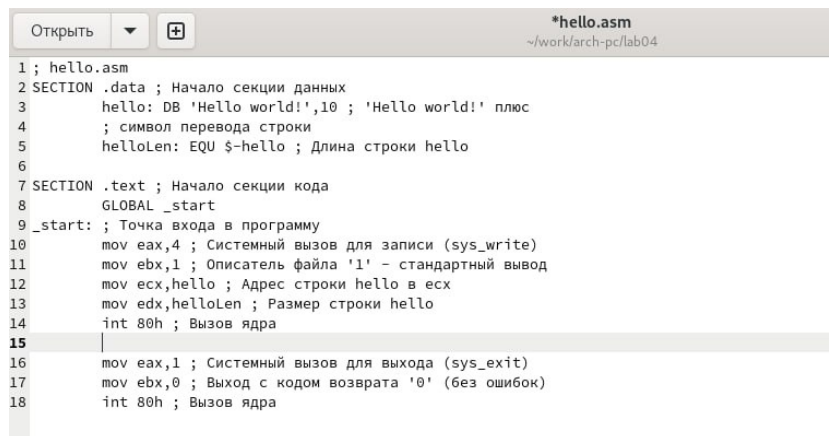
Создаю в текущем каталоге пустой текстовый файл hello.asm и открываю его с помощью текстового редактора gedit (рис. 3.2)



```
aanovikova123@fedora:~/work/arch-pc/lab04$ touch hello.asm
aanovikova123@fedora:~/work/arch-pc/lab04$ gedit hello.asm
```

Рис. 3.2: Создание пустого текстового файл и его открытие

Начинаю заполнять файл, вставляя программу для вывода “Hello world!”. (рис. 3.3)



```

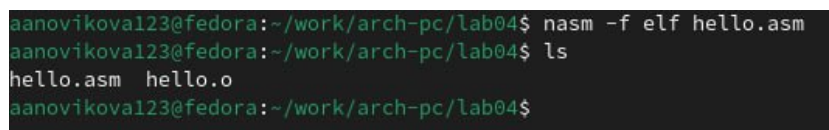
1; hello.asm
2 SECTION .data ; Начало секции данных
3     hello: DB 'Hello world!',10 ; 'Hello world!' плюс
4             ; символ перевода строки
5     helloLen: EQU $-hello ; Длина строки hello
6
7 SECTION .text ; Начало секции кода
8     GLOBAL _start
9 _start: ; Точка входа в программу
10     mov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

```

Рис. 3.3: Заполнение файла для программы вывода

3.2 Работа с транслятором NASM

Превращаю текст программы для вывода “Hello world!” в объектный код с помощью транслятора NASM, используя команду `nasm -f elf hello.asm`. Ключ `-f` указывает транслятору NASM, что следует создать бинарный файл в формате ELF. Далее проверяю правильность выполнения команды с помощью `ls`. Действительно, файл был создан. (рис. 3.4)



```

aanovikova123@fedora:~/work/arch-pc/lab04$ nasm -f elf hello.asm
aanovikova123@fedora:~/work/arch-pc/lab04$ ls
hello.asm  hello.o
aanovikova123@fedora:~/work/arch-pc/lab04$

```

Рис. 3.4: Компиляция текста программы

3.3 Работа с расширенным синтаксисом командной строки NASM

Ввожу команду, которая скомпилирует файл `hello.asm` в файл `obj.o`, при этом в файл будут включены символы для отладки (ключ `-g`), а также с помощью ключа `-l` будет создан файл листинга `list.lst`. Далее проверяю командой `ls` правильность выполнения команд. (рис. 3.5)


```
aanovikova123@fedora:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst hello.asm
aanovikova123@fedora:~/work/arch-pc/lab04$ ls
hello.asm  hello.o  list.lst  obj.o
```

Рис. 3.5: Преобразование файла hello.asm в obj.o

3.4 Работа с компоновщиком LD

Передаю объектный файл hello.o на обработку компоновщику LD, чтобы получить исполняемый файл hello. Ключ -o задаёт имя создаваемого исполняемого файла. Командой ls проверяю правильность выполнения команд. (рис. 3.6)

```
aanovikova123@fedora:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
aanovikova123@fedora:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  obj.o
```

Рис. 3.6: Передача объектного файла на обработку компоновщику

Выполняю следующую команду (рис. 3.7). Исполняемый файл будет иметь имя main, т.к. после ключа -o было задано значение main. Объектный файл, из которого собран этот исполняемый файл, имеет имя obj.o.

```
aanovikova123@fedora:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
aanovikova123@fedora:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
```

Рис. 3.7: Передача объектного файла на обработку компоновщику

3.5 Запуск исполняемого файла

Запускаю на выполнение созданный исполняемый файл hello. (рис. 3.8)

```
aanovikova123@fedora:~/work/arch-pc/lab04$ ./hello
Hello world!
aanovikova123@fedora:~/work/arch-pc/lab04$
```

Рис. 3.8: Запуск исполняемого файла

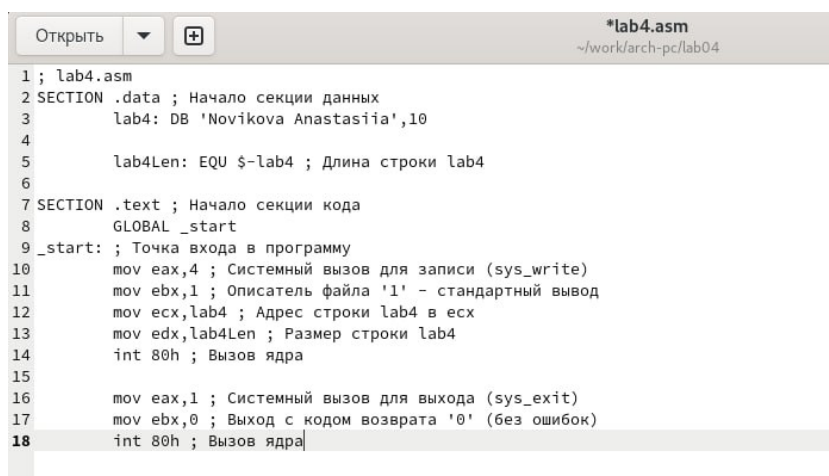
3.6 Задание для самостоятельной работы

Создаю копию файла `hello.asm` с новым именем `lab4.asm` (рис. fig. 3.9).

```
aanovikova123@fedora:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
aanovikova123@fedora:~/work/arch-pc/lab04$
```

Рис. 3.9: Создание копии файла

Открываю файл с помощью текстового редактора и вношу изменения так, чтобы выводились мои имя и фамилия. (рис. 3.10)



```
*lab4.asm
~/work/arch-pc/lab04

1 ; lab4.asm
2 SECTION .data ; Начало секции данных
3     lab4: DB 'Novikova Anastasiia',10
4
5     lab4Len: EQU $-lab4 ; Длина строки lab4
6
7 SECTION .text ; Начало секции кода
8     GLOBAL _start
9 _start: ; Точка входа в программу
10    mov eax,4 ; Системный вызов для записи (sys_write)
11    mov ebx,1 ; Описатель файла '1' - стандартный вывод
12    mov ecx,lab4 ; Адрес строки lab4 в ecx
13    mov edx,lab4Len ; Размер строки lab4
14    int 80h ; Вызов ядра
15
16    mov eax,1 ; Системный вызов для выхода (sys_exit)
17    mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
18    int 80h ; Вызов ядра
```

Рис. 3.10: Редактирование файла

Компилирую текст программы в объектный файл. С помощью команды `ls` проверяю, что файл `lab4.o` был создан. (рис. 3.11)

```
aanovikova123@fedora:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
aanovikova123@fedora:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4.asm lab4.o list.lst main obj.o
aanovikova123@fedora:~/work/arch-pc/lab04$
```

Рис. 3.11: Преобразование файла `lab4.asm` в файл `lab4.o`

Передаю объектный файл `lab4.o` на обработку компоновщику LD, чтобы получить исполняемый файл `lab4`. (рис. 3.12)

```

aanovikova123@fedora:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
aanovikova123@fedora:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  lab4  lab4.asm  lab4.o  list.lst  main  obj.o
aanovikova123@fedora:~/work/arch-pc/lab04$

```

Рис. 3.12: Передача объектного файла на обработку компоновщику

Запускаю исполняемый файл lab4, на экран действительно выводятся мои имя и фамилия. (рис. 3.13)

```

aanovikova123@fedora:~/work/arch-pc/lab04$ ./lab4
Novikova Anastasiia
aanovikova123@fedora:~/work/arch-pc/lab04$

```

Рис. 3.13: Запуск исполняемого файла lab4

Затем копирую файлы в локальный репозиторий для лабораторных работ. (рис. 3.14)

```

aanovikova123@fedora:~/work/arch-pc/lab04$ cp hello.asm ~/work/study/2024-2025/"Архитектура комп
ьютера"/arch-pc/labs/lab04
aanovikova123@fedora:~/work/arch-pc/lab04$ cp lab4.asm ~/work/study/2024-2025/"Архитектура комп
ьютера"/arch-pc/labs/lab04
aanovikova123@fedora:~/work/arch-pc/lab04$

```

Рис. 3.14: Копирование файлов lab4.asm и hello.asm

Выбираю файлы для загрузки на github с помощью команды `git add .`, с помощью команды `git commit -am` комментирую загружаемые файлы и отправляю их на сервер командой `git push`.

4 Выводы

В ходе лабораторной работы были изучены процедуры компиляции и сборки программ на ассемблере NASM.