

Лабораторная работа №4

Архитектура компьютеров

Ермишина Мария Кирилловна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Программа Hello world!	6
2.2	Транслятор NASM	7
2.3	Компоновщик LD	8
2.4	Выполнение заданий для самостоятельной работы.	9
3	Выводы	11

Список иллюстраций

2.1	Создание каталога и файла	6
2.2	Программа hello.asm	7
2.3	Трансляция hello.asm	7
2.4	Трансляция hello.asm с дополнительными опциями	8
2.5	Линковка программы	8
2.6	Линковка программы	8
2.7	Запуск программ	9
2.8	Код программы в файле lab4.asm	9
2.9	Запуск программы lab4.asm	10

Список таблиц

1 Цель работы

Целью работы является освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

2 Выполнение лабораторной работы

2.1 Программа Hello world!

Создаю каталог lab04 командой `mkdir`, перехожу в него с помощью команды `cd`, создаю файл `hello.asm`.

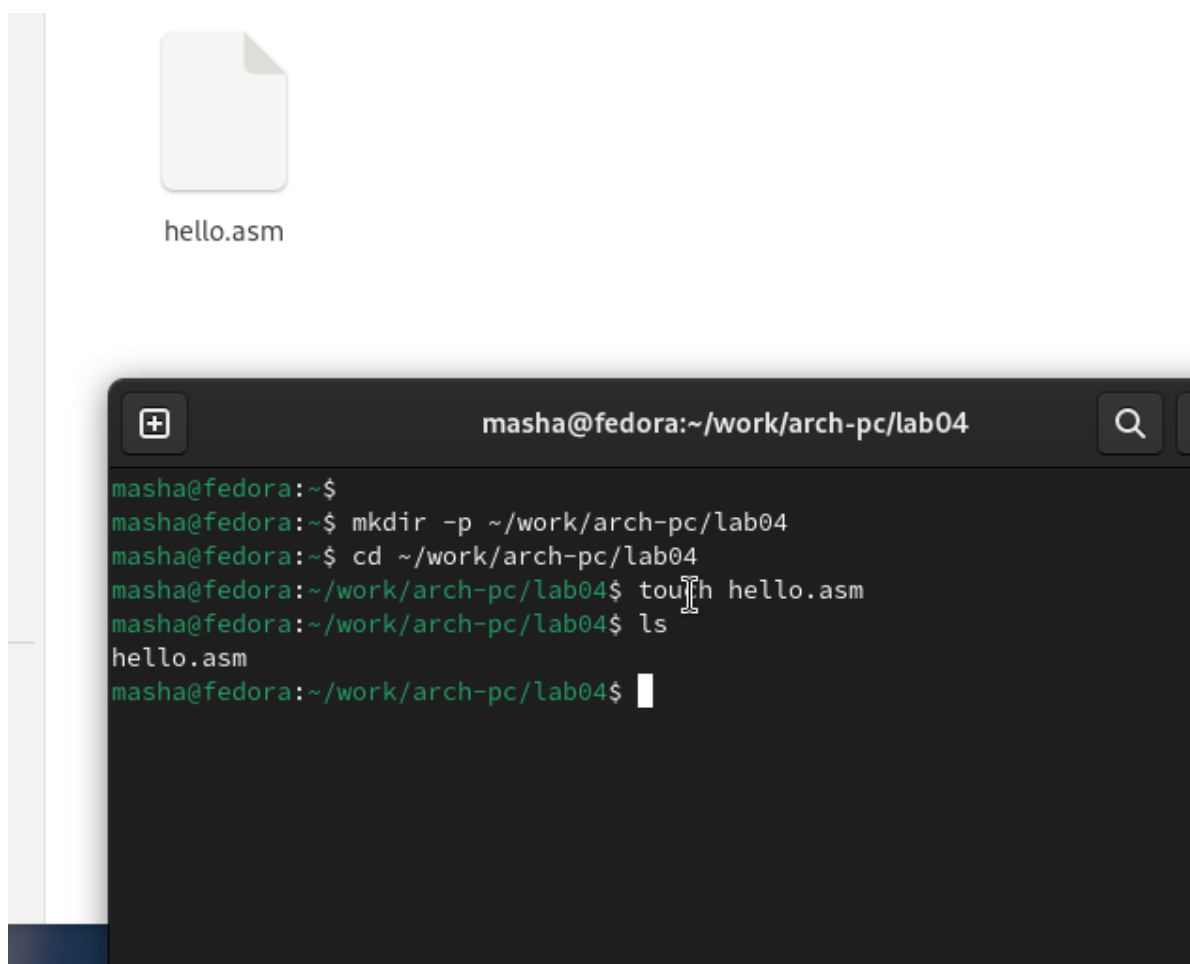
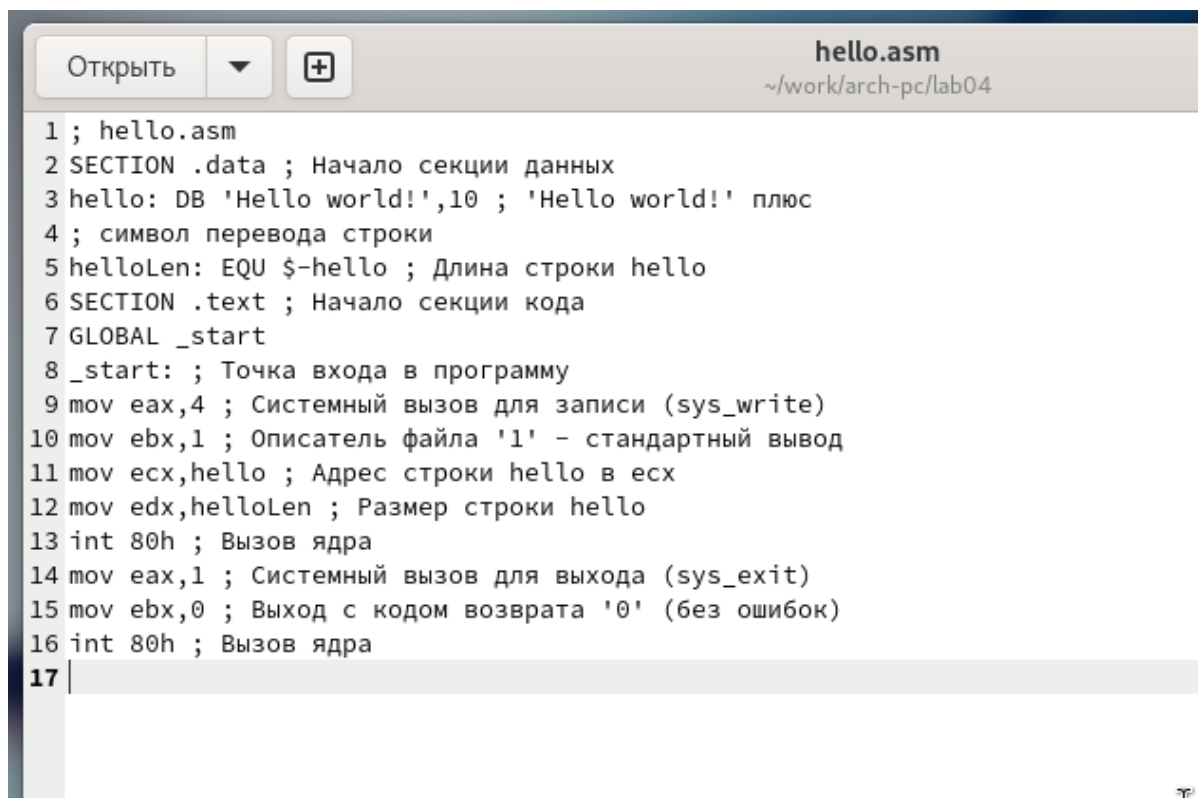


Рис. 2.1: Создание каталога и файла

Открываю файл и пишу код программы по заданию.

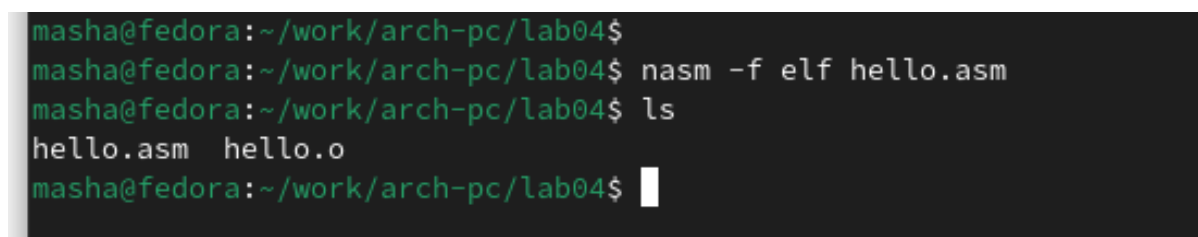


```
1 ; hello.asm
2 SECTION .data ; Начало секции данных
3 hello: DB 'Hello world!',10 ; 'Hello world!' плюс
4 ; символ перевода строки
5 helloLen: EQU $-hello ; Длина строки hello
6 SECTION .text ; Начало секции кода
7 GLOBAL _start
8 _start: ; Точка входа в программу
9 mov eax,4 ; Системный вызов для записи (sys_write)
10 mov ebx,1 ; Описатель файла '1' - стандартный вывод
11 mov ecx,hello ; Адрес строки hello в ecx
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
17 |
```

Рис. 2.2: Программа hello.asm

2.2 Транслятор NASM

Транслирую файл командой `nasm`. Получился объектный файл `hello.o`.



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

Рис. 2.3: Трансляция hello.asm

Транслирую файл командой `nasm` с дополнительными опциями. Получился файл листинга `list.lst`, объектный файл `obj.o`, и в программу добавилась отла-

дочная информация.

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

Рис. 2.4: Трансляция hello.asm с дополнительными опциями

2.3 Компоновщик LD

Выполняю линковку командой `ld` и получаю исполняемый файл.

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

Рис. 2.5: Линковка программы

Еще раз выполняю линковку для объектного файла `obj.o` и получаю исполняемый файл `main`.

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

Рис. 2.6: Линковка программы

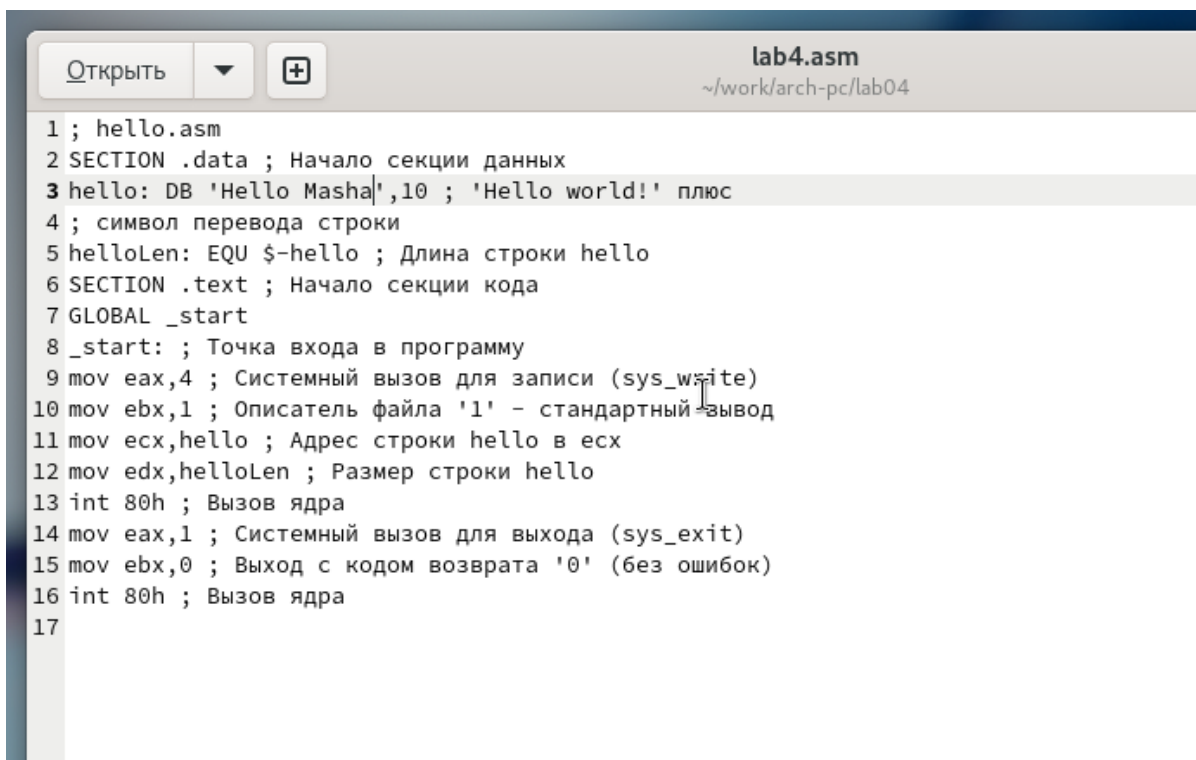
Запускаю исполняемые файлы.


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

Рис. 2.7: Запуск программ

2.4 Выполнение заданий для самостоятельной работы.

Копирую программу в новый файл, изменяю сообщение Hello world на свое имя и запускаю новую программу.



```
1 ; hello.asm  
2 SECTION .data ; Начало секции данных  
3 hello: DB 'Hello Masha',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 ecx  
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  
17
```

Рис. 2.8: Код программы в файле lab4.asm

```
masha@fedora:~/work/arch-pc/lab04$  
masha@fedora:~/work/arch-pc/lab04$  
masha@fedora:~/work/arch-pc/lab04$ nasm -f elf lab4.asm  
masha@fedora:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4  
masha@fedora:~/work/arch-pc/lab04$ ./lab4  
Hello Masha  
masha@fedora:~/work/arch-pc/lab04$
```

Рис. 2.9: Запуск программы lab4.asm

3 Выводы

При выполнении данной лабораторной работы освоили процесс компиляции и сборки программ, написанных на ассемблере `nasm`.