

Отчёт по лабораторной работе 4

дисциплина: Архитектура компьютера

Тяпкова Альбина НММбд-04-24

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	Запуск исполняемого файла	8
2.5	Задание для самостоятельной работы	9
3	Выводы	11

Список иллюстраций

2.1	Создан каталог для работы и файл для программы	6
2.2	Программа в файле hello.asm	7
2.3	Трансляция программы	8
2.4	Компоновка программы	8
2.5	Запуск программы	9
2.6	Программа в файле lab4.asm	9
2.7	Проверка программы lab4.asm	10

Список таблиц

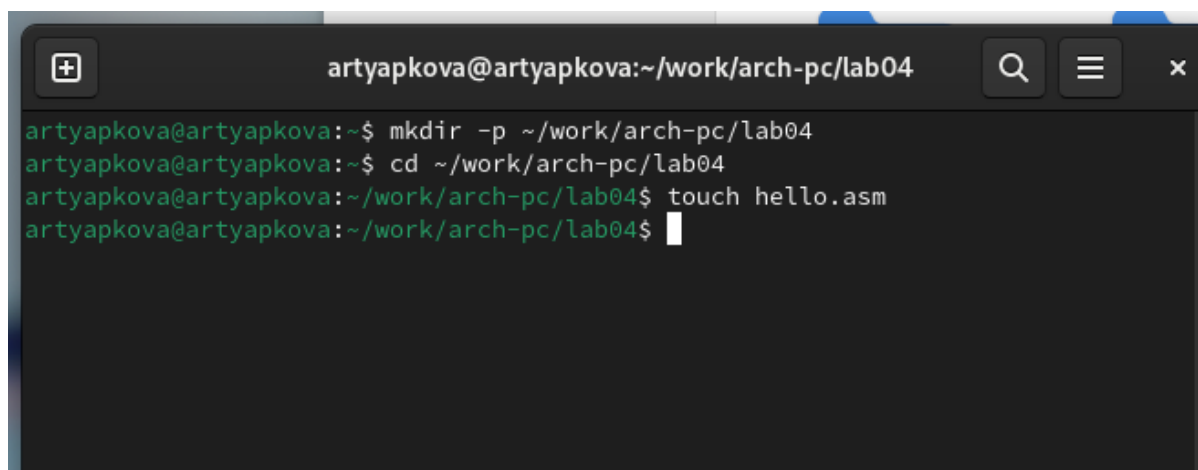
1 Цель работы

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

2 Выполнение лабораторной работы

2.1 Программа Hello world!

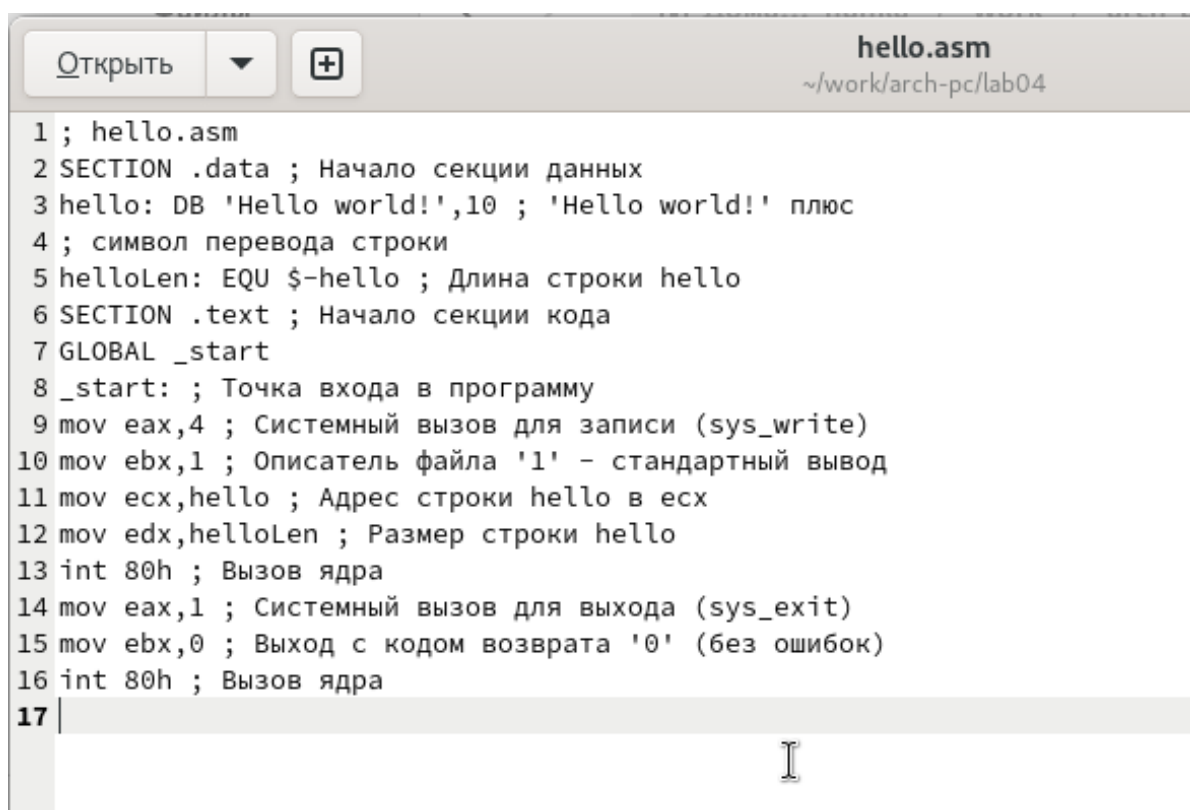
Я создала каталог lab04 с помощью команды `mkdir`, затем перешла в него, используя команду `cd`, и создала файл `hello.asm`, в который будет написана программа. Чтобы убедиться, что файл был успешно создан, я выполнила команду `ls`.

A screenshot of a terminal window with a dark background. The window title bar shows the user 'artyapkova' and the current directory '~/work/arch-pc/lab04'. The terminal contains the following commands and their outputs:

```
artyapkova@artyapkova:~$ mkdir -p ~/work/arch-pc/lab04
artyapkova@artyapkova:~$ cd ~/work/arch-pc/lab04
artyapkova@artyapkova:~/work/arch-pc/lab04$ touch hello.asm
artyapkova@artyapkova:~/work/arch-pc/lab04$
```

Рис. 2.1: Создан каталог для работы и файл для программы

После этого я написала программу на языке ассемблера 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 в файле hello.asm

2.2 Транслятор NASM

Транслятор **NASM** превращает текст программы в объектный код. Если в тексте программы нет ошибок, транслятор преобразует её из файла `hello.asm` в объектный файл `hello.o`.

Я скомпилировала файл командой `nasm` и получила объектный файл `hello.o`.

Затем я повторно скомпилировала файл `hello.asm`, но уже с дополнительными опциями:

- с опцией `-l` был получен файл листинга `list.lst`;
- с опцией `-f` получен объектный файл `obj.o`;
- с опцией `-g` в программу добавлена отладочная информация.

```

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

```

Рис. 2.3: Трансляция программы

2.3 Компоновщик LD

Чтобы получить исполняемую программу, необходимо передать объектный файл на обработку компоновщику **LD**.

Я выполнила команду `ld` и скомпоновала объектный файл `hello.o`, получив исполняемый файл `hello`.

После этого я повторила команду `ld`, но уже для файла `obj.o`, что позволило мне получить исполняемый файл `main`.

```

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

```

Рис. 2.4: Компоновка программы

2.4 Запуск исполняемого файла

Запустила оба исполняемых файла, чтобы проверить их работу.

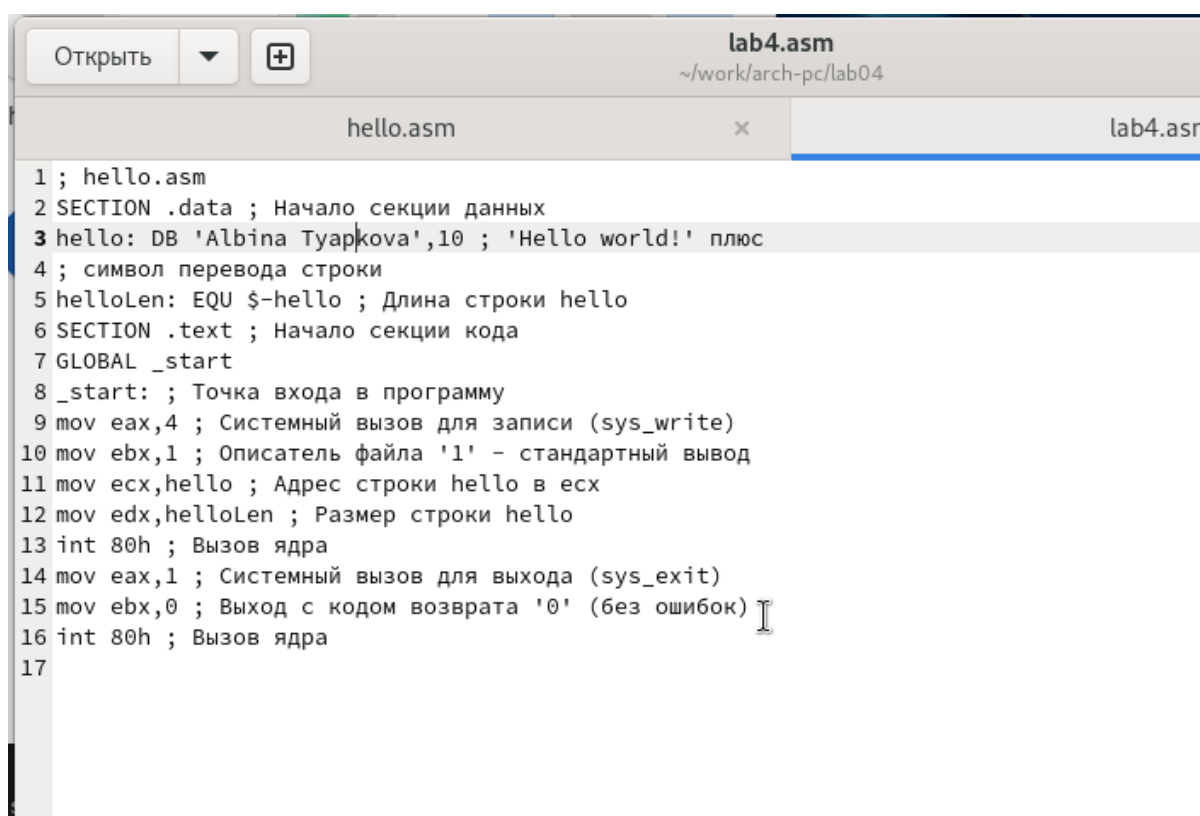

```
artyapkova@artyapkova:~/work/arch-pc/lab04$  
artyapkova@artyapkova:~/work/arch-pc/lab04$ ./hello  
Hello world!  
artyapkova@artyapkova:~/work/arch-pc/lab04$ ./main  
Hello world!  
artyapkova@artyapkova:~/work/arch-pc/lab04$
```

Рис. 2.5: Запуск программы

2.5 Задание для самостоятельной работы

Для выполнения самостоятельного задания я скопировала файл `hello.asm` в новый файл `lab4.asm` командой `cp`.

В файле `lab4.asm` я изменила текст сообщения с “Hello world” на своё имя.



```
lab4.asm  
~/work/arch-pc/lab04  
hello.asm x lab4.asm  
1 ; hello.asm  
2 SECTION .data ; Начало секции данных  
3 hello: DB 'Albina Tyapkova',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. 2.6: Программа в файле `lab4.asm`

Затем запустила изменённую программу и проверила результат.

```
artyapkova@artyapkova:~/work/arch-pc/lab04$  
artyapkova@artyapkova:~/work/arch-pc/lab04$ cp hello.asm lab4.asm  
artyapkova@artyapkova:~/work/arch-pc/lab04$  
artyapkova@artyapkova:~/work/arch-pc/lab04$ nasm -f elf lab4.asm  
artyapkova@artyapkova:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4  
artyapkova@artyapkova:~/work/arch-pc/lab04$ ./lab4  
Albina Tyapkova  
artyapkova@artyapkova:~/work/arch-pc/lab04$
```

Рис. 2.7: Проверка программы lab4.asm

3 Выводы

Таким образом, я успешно выполнила лабораторную работу, используя **NASM** и **LD** для трансляции, компоновки и запуска программ на языке ассемблера.