

Отчёт по лабораторной работе 4

Архитектура компьютеров

ТРАОРЕ АНРИ НОЭЛЬ

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	10

Список иллюстраций

2.1	Создание каталога и файла	6
2.2	Программа в файле hello.asm	7
2.3	Трансляция, линковка и запуск программы	8
2.4	Программа в файле lab4.asm	9
2.5	Сборка и проверка программы lab4.asm	9

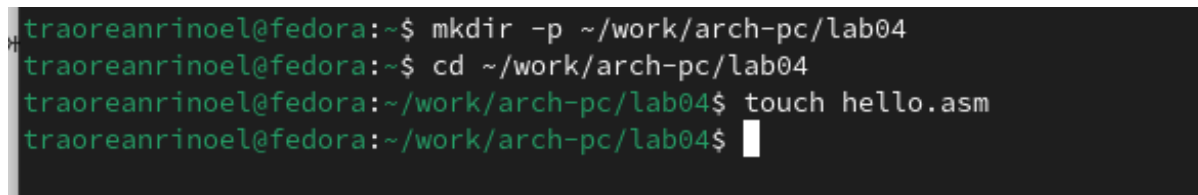
Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

Я создал каталог lab04 с помощью команды mkdir, затем перешел в него с помощью команды cd и создал файл hello.asm. (рис. 2.1)

A screenshot of a terminal window with a dark background and green text. It shows four lines of commands and their prompts. The first line is 'mkdir -p ~/work/arch-pc/lab04' with prompt 'traoreanrinoel@fedora:~\$'. The second line is 'cd ~/work/arch-pc/lab04' with prompt 'traoreanrinoel@fedora:~\$'. The third line is 'touch hello.asm' with prompt 'traoreanrinoel@fedora:~/work/arch-pc/lab04\$'. The fourth line shows the prompt 'traoreanrinoel@fedora:~/work/arch-pc/lab04\$' followed by a cursor.

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

Рис. 2.1: Создание каталога и файла

Открыл файл и написал код программы в соответствии с заданием. (рис. 2.2)



```
SECTION .data
    hello:      db "Hello, world!",0xa
    hellolen:   equ $ - hello

SECTION .text
    global _start

_start:
    mov eax, 4
    mov ebx, 1
    mov ecx, hello
    mov edx, hellolen
    int 0x80

    mov eax, 1
    mov ebx, 0
    int 0x80
```

Рис. 2.2: Программа в файле hello.asm

С помощью команды `nasm` я транслировал файл, что привело к созданию объектного файла `hello.o`.

Повторно транслировал файл с использованием дополнительных опций команды `nasm`. В результате были созданы файл листинга `list.lst`, объектный файл `obj.o`, а также в программу была добавлена отладочная информация.

С помощью команды `ld` я выполнил линковку и получил исполняемый файл.

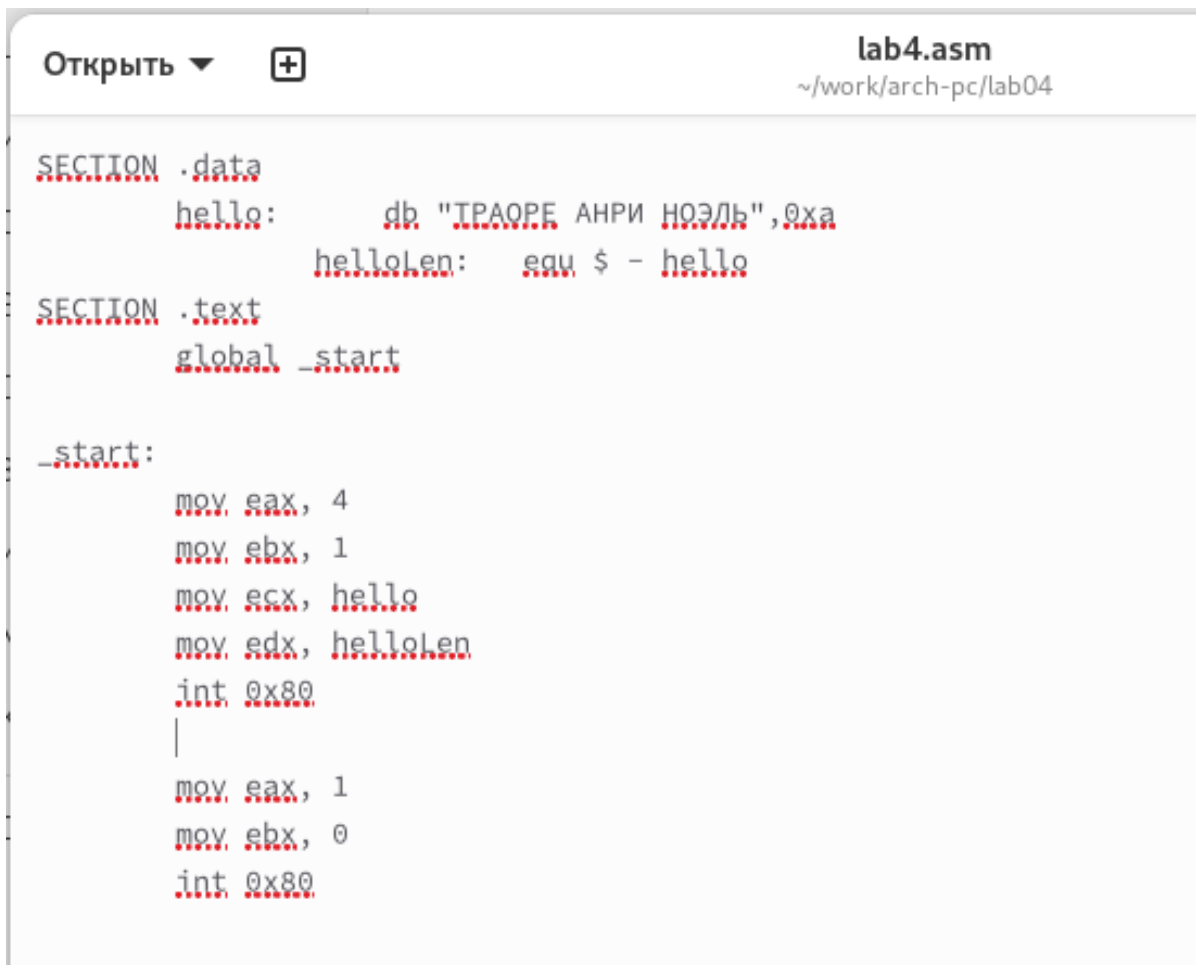
Выполнил еще одну линковку для объектного файла `obj.o` и получил исполняемый файл с именем `main`.

Запустил исполняемые файлы и проверил их работу. (рис. 2.3)

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

Рис. 2.3: Трансляция, линковка и запуск программы

Изменил сообщение Hello world на свое имя и запустил файл еще раз. (рис. 2.4)
(рис. 2.5)

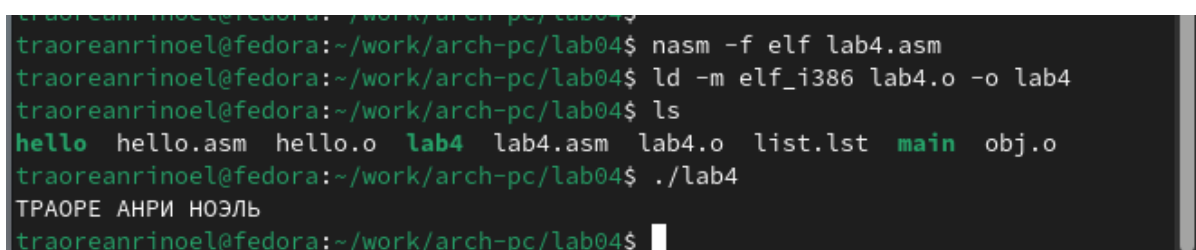


```
SECTION .data
    hello:      db "ТРАОРЕ АНРИ НОЭЛЬ",0xa
                hellolen: equ $ - hello

SECTION .text
    global _start

_start:
    mov eax, 4
    mov ebx, 1
    mov ecx, hello
    mov edx, hellolen
    int 0x80
    |
    mov eax, 1
    mov ebx, 0
    int 0x80
```

Рис. 2.4: Программа в файле lab4.asm



```
traoreanrinoel@fedora:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
traoreanrinoel@fedora:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
traoreanrinoel@fedora:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  lab4  lab4.asm  lab4.o  list.lst  main  obj.o
traoreanrinoel@fedora:~/work/arch-pc/lab04$ ./lab4
ТРАОРЕ АНРИ НОЭЛЬ
traoreanrinoel@fedora:~/work/arch-pc/lab04$
```

Рис. 2.5: Сборка и проверка программы lab4.asm

3 Выводы

Освоили процесс компиляции и сборки программ, написанных на ассемблере `nasm`.