

# **Отчёт по лабораторной работе 4**

**Архитектура компьютеров**

Саммура Халед

# 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


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



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

Рис. 2.1: Создание каталога и файла

2. Открыл файл и написал код программы по заданию.

A screenshot of a text editor window titled 'hello.asm' with a path '~/.work/arch-pc/lab04'. The editor contains assembly code for a 'hello, world!' program. The code is divided into two sections: '.data' and '.text'. The '.data' section defines a string 'hello' and its length 'helloLen'. The '.text' section starts at '\_start' and contains instructions to set up registers (eax, ebx, ecx, edx) and call the 'int 0x80' system call twice. The first call sets up parameters for writing 4 bytes to stdout (ebx=1, ecx=hello, edx=helloLen). The second call sets up parameters for exiting (eax=1, ebx=0).

```
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

3. Транслировал файл командой `nasm`. Получился объектный файл `hello.o`
4. Транслировал файл командой `nasm` с дополнительными опциями. Получился файл листинга `list.lst`, объектный файл `obj.o`, в программу добавилась отладочная информация.
5. Выполнил линковку командой `ld` и получил исполняемый файл.
6. Еще раз выполнил линковку для объектного файла `obj.o` и получил испол-

няемый файл main.

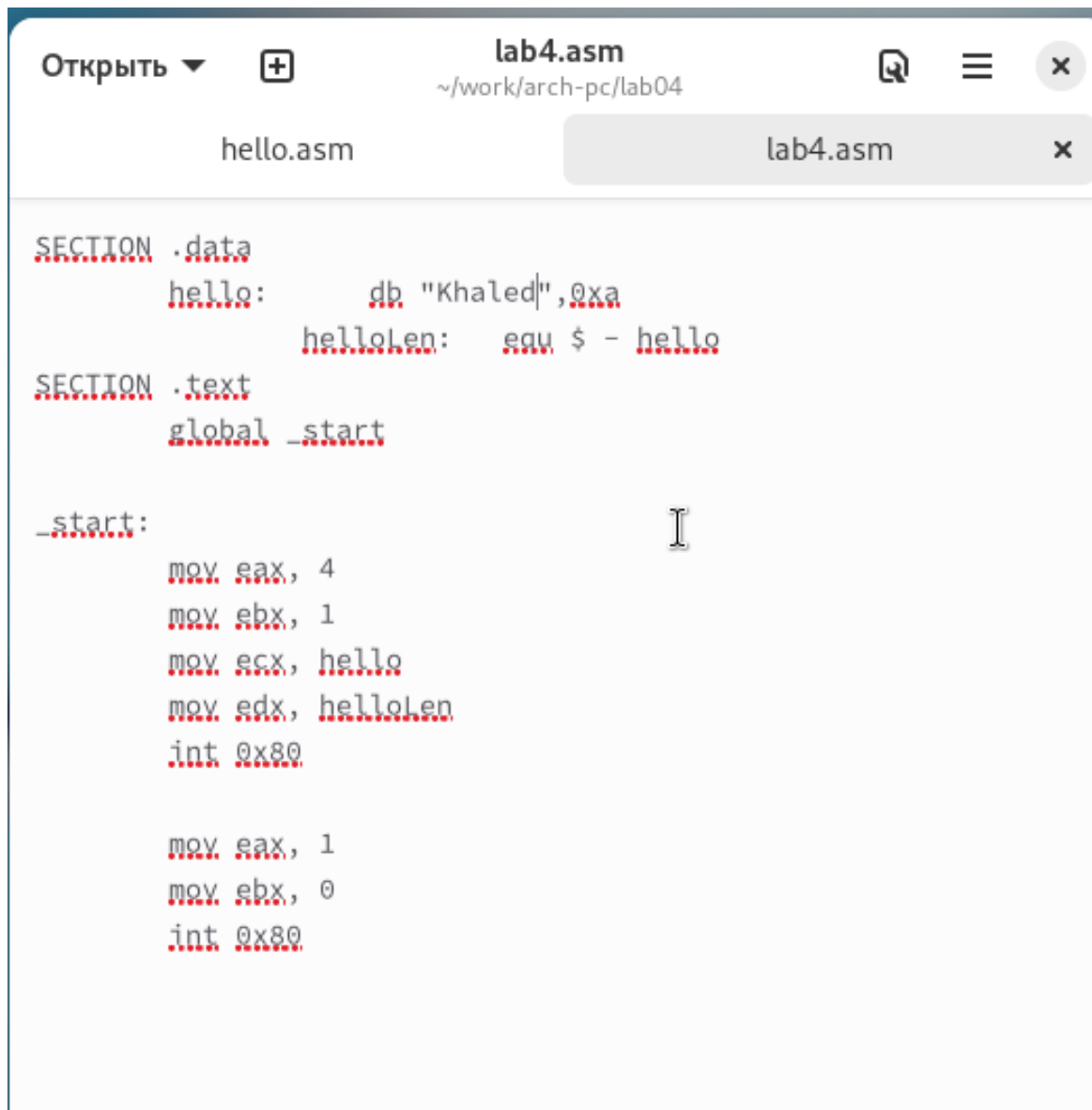
7. Запустил исполняемые файлы.

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

Рис. 2.3: Трансляция, линковка и запуск программы

8. Изменил сообщение Hello world на свое имя и запустил файл еще раз.





```
SECTION .data
hello:      db "Khaled",0xa
helloLen:   equ $ - hello

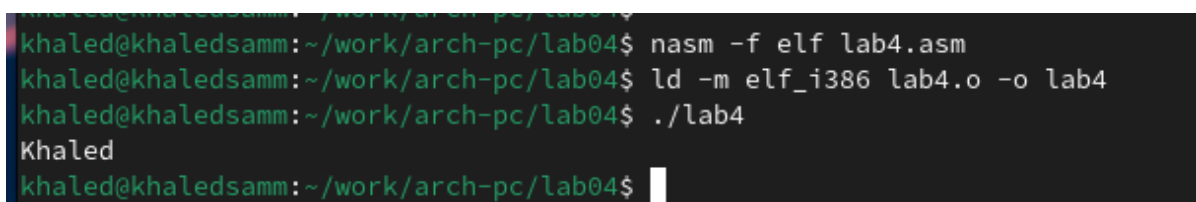
SECTION .text
global _start

_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



```
khaled@khaledsam: ~/work/arch-pc/lab04$ nasm -f elf lab4.asm
khaled@khaledsam: ~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
khaled@khaledsam: ~/work/arch-pc/lab04$ ./lab4
Khaled
khaled@khaledsam: ~/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`.