

# **Отчёт по лабораторной работе 4**

**дисциплина: Архитектура компьютера**

Быкасов Владислав Дмитриевич НБИбд-01-24

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
2.1	Программа Hello world! . . . . .	6
2.2	Трансляция кода с помощью NASM . . . . .	7
2.3	Линковка с использованием LD . . . . .	8
2.4	Выполнение заданий для самостоятельной работы . . . . .	9
2.5	Выводы . . . . .	10

## 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!

Для начала создаю новый каталог **lab04** с помощью команды **mkdir**, затем перехожу в него, используя команду **cd**. После этого создаю файл **hello.asm**. На рис. 2.1 показан процесс создания каталога и файла.

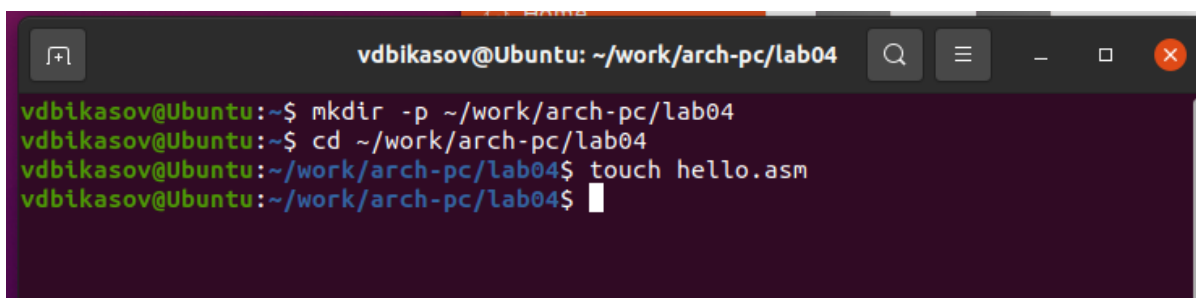
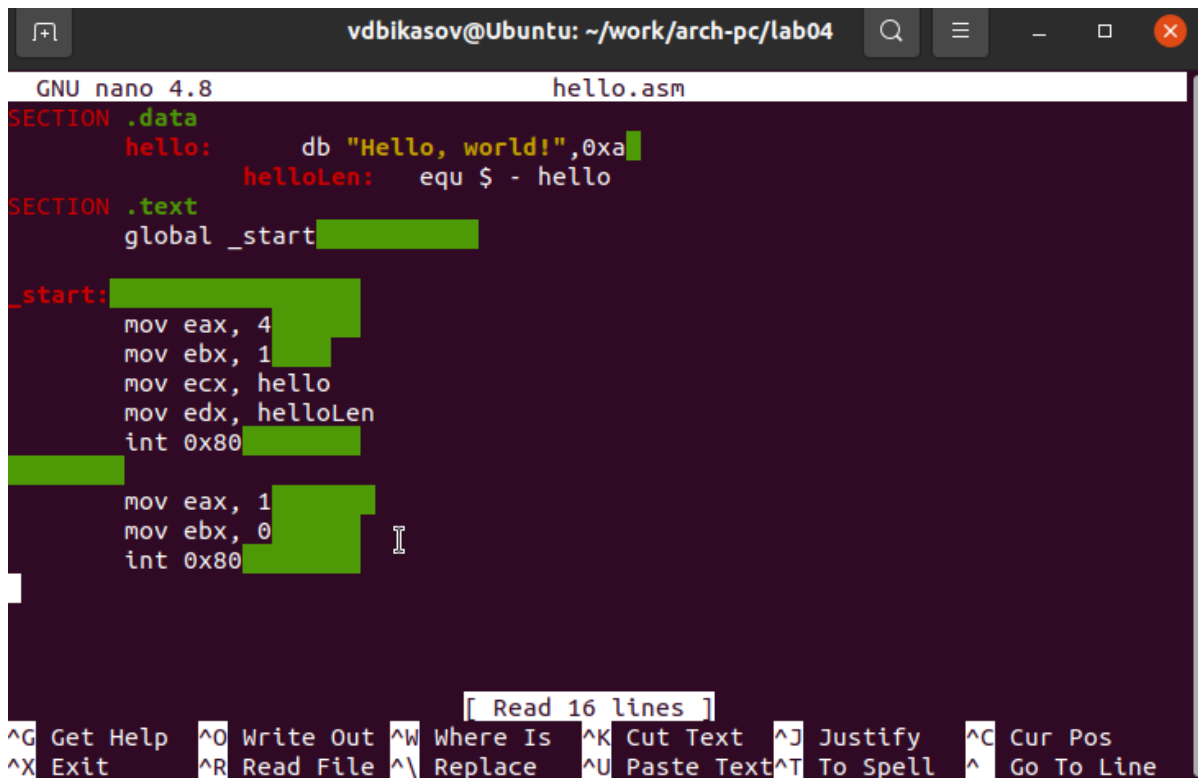
A screenshot of a terminal window with a dark purple background. The window title bar shows the user 'vdbikasov' on an 'Ubuntu' machine, with the current directory being '~/work/arch-pc/lab04'. The terminal displays four lines of commands and their outputs: 1. 'vdbikasov@Ubuntu:~\$ mkdir -p ~/work/arch-pc/lab04' followed by a new line. 2. 'vdbikasov@Ubuntu:~\$ cd ~/work/arch-pc/lab04' followed by a new line. 3. 'vdbikasov@Ubuntu:~/work/arch-pc/lab04\$ touch hello.asm' followed by a new line. 4. 'vdbikasov@Ubuntu:~/work/arch-pc/lab04\$' followed by a cursor. The window has standard Ubuntu window controls (minimize, maximize, close) on the right side of the title bar.

Рис. 2.1: Создание каталога и файла

Открываю файл **hello.asm** в текстовом редакторе и пишу код программы по заданию, как показано на рис. 2.2.



```
GNU nano 4.8 hello.asm
SECTION .data
    hello:    db "Hello, world!",0xa
    helloLen: equ $ - hello

SECTION .text
    global _start

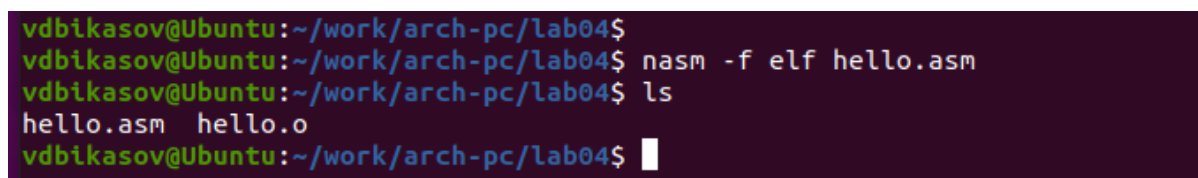
_start:
    mov eax, 4
    mov ebx, 1
    mov ecx, hello
    mov edx, helloLen
    int 0x80

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

Рис. 2.2: Программа hello.asm

## 2.2 Трансляция кода с помощью NASM

Чтобы скомпилировать файл, использую транслятор NASM. С помощью команды **nasm** создаю объектный файл **hello.o**, что показано на рис. 2.3.



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

Рис. 2.3: Трансляция hello.asm

Для более детального анализа программы использую команду **nasm** с дополнительными опциями, которые позволяют создать листинг (**list.lst**), объектный файл (**obj.o**), а также добавить отладочную информацию. Результат показан на рис. 2.4.

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

Рис. 2.4: Трансляция hello.asm с дополнительными опциями

## 2.3 Линковка с использованием LD

После успешной трансляции выполняю линковку объектного файла **hello.o**, используя компоновщик **ld**. Это позволяет создать исполняемый файл, как показано на рис. 2.5.

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

Рис. 2.5: Линковка программы

Затем повторяю этот процесс для объектного файла **obj.o**, в результате чего получаю исполняемый файл с именем **main**. Результат показан на рис. 2.6.

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

Рис. 2.6: Линковка программы

Запускаю оба полученных исполняемых файла, как видно на рис. 2.7.



```

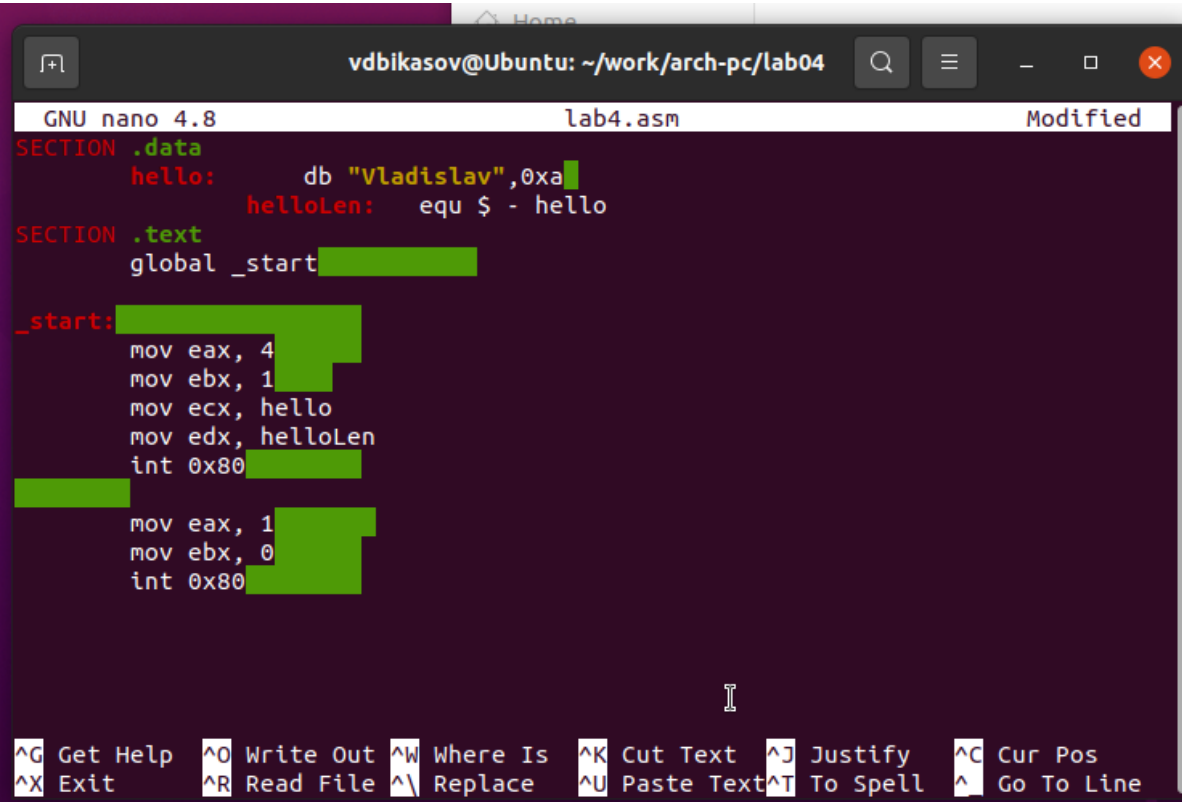
vdbikasov@Ubuntu:~/work/arch-pc/lab04$
vdbikasov@Ubuntu:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
vdbikasov@Ubuntu:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o list.lst main obj.o
vdbikasov@Ubuntu:~/work/arch-pc/lab04$
vdbikasov@Ubuntu:~/work/arch-pc/lab04$ ./hello
Hello, world!
vdbikasov@Ubuntu:~/work/arch-pc/lab04$

```

Рис. 2.7: Запуск программ

## 2.4 Выполнение заданий для самостоятельной работы

Для выполнения индивидуального задания копирую исходную программу в новый файл. Вношу изменения в код, заменяя сообщение “Hello world” на своё имя, что продемонстрировано на рис. 2.8. После этого запускаю изменённую программу (рис. 2.9).



```

GNU nano 4.8 lab4.asm Modified
SECTION .data
    hello:    db "Vladislav",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

```

Terminal window title: vdbikasov@Ubuntu: ~/work/arch-pc/lab04

Footer shortcuts: ^G Get Help, ^O Write Out, ^W Where Is, ^K Cut Text, ^J Justify, ^C Cur Pos, ^X Exit, ^R Read File, ^\ Replace, ^U Paste Text, ^T To Spell, ^\_ Go To Line

Рис. 2.8: Код программы в файле lab4.asm

```
vdbikarov@Ubuntu:~/work/arch-pc/lab04$  
vdbikarov@Ubuntu:~/work/arch-pc/lab04$ cp hello.asm lab4.asm  
vdbikarov@Ubuntu:~/work/arch-pc/lab04$ nano lab4.asm  
vdbikarov@Ubuntu:~/work/arch-pc/lab04$ nasm -f elf lab4.asm  
vdbikarov@Ubuntu:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4  
vdbikarov@Ubuntu:~/work/arch-pc/lab04$ ./lab4  
Vladislav  
vdbikarov@Ubuntu:~/work/arch-pc/lab04$
```

Рис. 2.9: Запуск программы lab4.asm

## 2.5 Выводы

В ходе выполнения лабораторной работы я ознакомился с основными этапами работы с программами на ассемблере с использованием NASM. Были освоены такие важные шаги, как создание объектных файлов, компиляция кода, использование компоновщика LD, а также работа с отладочной информацией и запуск готовых программ.