

# Содержание

<b>1</b>	<b>Российский университет дружбы народов</b>	<b>4</b>
1.1	Дисциплина «Архитектура компьютера» . . . . .	4
<b>2</b>	<b>1. Цель работы</b>	<b>5</b>
<b>3</b>	<b>2. Описание результатов выполнения лабораторной работы</b>	<b>6</b>
3.1	2.1 Программа Hello world! . . . . .	6
3.2	2.2 . Транслятор NASM . . . . .	7
3.3	2.3. Расширенный синтаксис командной строки NASM . . . . .	8
3.4	2.4. Компоновщик LD . . . . .	8
3.5	2.5 Запуск исполняемого файла . . . . .	8
<b>4</b>	<b>3. Задание для самостоятельной работы</b>	<b>10</b>
4.1	3.1 Создание и изменение lab4.asm . . . . .	10
4.2	3.2 Трансляция и компоновка lab4.asm . . . . .	11
4.3	3.3 Копирование файлов в каталог курса и загрузка на GitHub . . .	11
<b>5</b>	<b>4. Выводы</b>	<b>13</b>

## Список иллюстраций

3.1	Создание каталога lab04 . . . . .	6
3.2	Создание и редактирование hello.asm . . . . .	7
3.3	Результат трансляции hello.asm . . . . .	7
3.4	Расширенная трансляция NASM . . . . .	8
3.5	Компоновка hello и main . . . . .	8
3.6	Запуск программы hello . . . . .	9
4.1	Копирование и открытие lab4.asm . . . . .	10
4.2	Трансляция lab4.asm . . . . .	11
4.3	Компоновка и запуск lab4 . . . . .	11
4.4	Копирование файлов в каталог курса . . . . .	11
4.5	Проверка каталога lab04 . . . . .	12
4.6	Коммит и push на GitHub . . . . .	12

## **Список таблиц**

# 1 Российский университет дружбы народов

## 1.1 Дисциплина «Архитектура компьютера»

Лабораторная работа №4 «Создание и процесс обработки программ на языке ассемблера NASM»

Студент: [Лань Цяньин] Группа: [НПИбд-03-25] Номер студенческого билета: [1132254528]

Москва, 2025 г.

---

format:

pdf:

toc: false

lof: false

lot: false

docx:

toc: false

number-sections: false

---

## **2 1. Цель работы**

Освоить процедуру компиляции, компоновки и запуска программ, написанных на языке ассемблера NASM, а также закрепить на практике процесс создания простых программ и работу с файловой структурой курса.

## 3 2. Описание результатов выполнения лабораторной работы

### 3.1 2.1 Программа Hello world!

Создание рабочего каталога для лабораторной работы №4 (рис. Рисунок 3.1).

```
clanj1@clanj1:~/work$ mkdir -p ~/work/arch-pc/lab04  
clanj1@clanj1:~/work$ cd ~/work/arch-pc/lab04
```

Рисунок 3.1: Создание каталога lab04

---

Создание файла hello.asm и открытие его для ввода исходного кода (рис. Рисунок 3.2).

```
clanj1@clanj1:~/work/arch-pc/lab04$ touch hello.asm
clanj1@clanj1:~/work/arch-pc/lab04$ gedit hello.asm
```

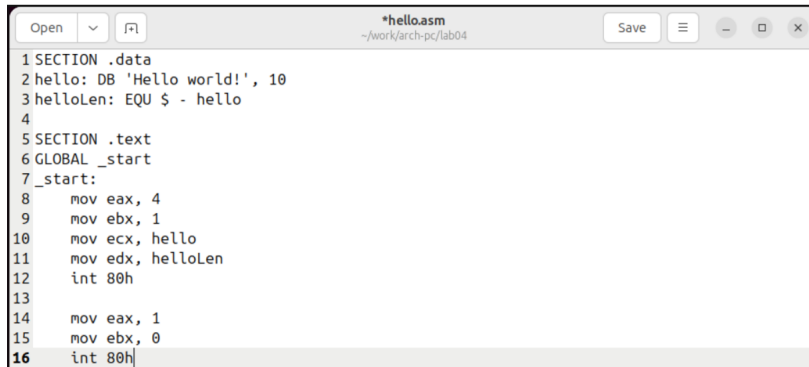


Рисунок 3.2: Создание и редактирование hello.asm

Этап демонстрирует корректное создание исходного файла и начало формирования рабочей программы, что обеспечивает возможность перехода к трансляции.

## 3.2 2.2 . Транслятор NASM

Трансляция файла hello.asm и проверка появления объектного модуля hello.o (рис. Рисунок 3.3).

```
clanj1@clanj1:~/work/arch-pc/lab04$ nasm -f elf hello.asm
clanj1@clanj1:~/work/arch-pc/lab04$ ls
hello.asm  hello.o
```

Рисунок 3.3: Результат трансляции hello.asm

Наличие файла hello.o подтверждает корректность трансляции и готовность к дальнейшей компоновке программы.

## 3.3 2.3. Расширенный синтаксис командной строки NASM

Трансляция `hello.asm` с дополнительными параметрами NASM для получения `obj.o` и `list.lst` (рис. Рисунок 3.4).

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

Рисунок 3.4: Расширенная трансляция NASM

Вывод `ls` показывает появление `obj.o` и `list.lst`, что означает успешное формирование объектного модуля и листинга, позволяющего проверить структуру и адресацию команд.

## 3.4 2.4. Компоновщик LD

Компоновка файлов `hello.o` и `obj.o` в исполняемые программы `hello` и `main` (рис. Рисунок 3.5).

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

Рисунок 3.5: Компоновка `hello` и `main`

Вывод `ls` фиксирует появление файлов `hello` и `main`, что подтверждает успешное создание двух исполняемых программ из разных объектных модулей.

## 3.5 2.5 Запуск исполняемого файла

Запуск исполняемой программы `hello` после компоновки (рис. Рисунок 3.6).

A terminal window with a dark background. The prompt is 'clanj1@clanj1:~/work/arch-pc/lab04\$' in green and blue. The command './hello' is entered in blue. The output 'Hello world!' is displayed in white on the next line.

```
clanj1@clanj1:~/work/arch-pc/lab04$ ./hello
Hello world!
```

Рисунок 3.6: Запуск программы hello

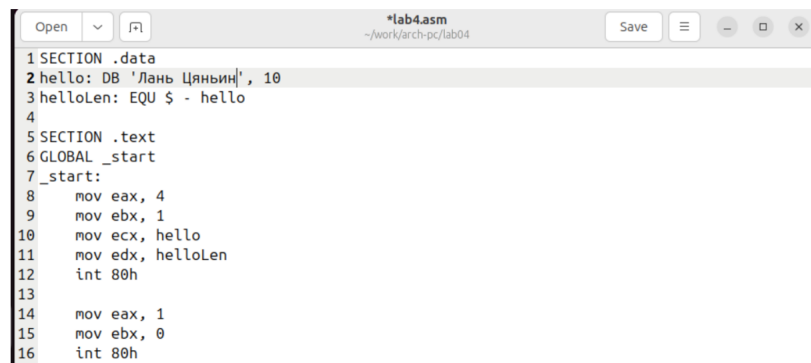
Вывод «Hello World!» подтверждает корректное выполнение программы и правильную работу системы ввода-вывода.

## 4 3. Задание для самостоятельной работы

### 4.1 3.1 Создание и изменение lab4.asm

Копирование файла hello.asm в lab4.asm и последующее открытие lab4.asm в редакторе для просмотра и редактирования (рис. Рисунок 4.1).

```
clanj1@clanj1:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
clanj1@clanj1:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4.asm list.lst main obj.o
clanj1@clanj1:~/work/arch-pc/lab04$ gedit lab4.asm
```



```
1 SECTION .data
2 hello: DB 'Лань Цяньин', 10
3 helloLen: EQU $ - hello
4
5 SECTION .text
6 GLOBAL _start
7 _start:
8     mov eax, 4
9     mov ebx, 1
10    mov ecx, hello
11    mov edx, helloLen
12    int 80h
13
14    mov eax, 1
15    mov ebx, 0
16    int 80h
```

Рисунок 4.1: Копирование и открытие lab4.asm

Вывод ls подтверждает успешное создание lab4.asm, а отображение исходного кода в редакторе показывает, что файл корректно скопирован и готов для дальнейших изменений в рамках задания.

## 4.2 3.2 Трансляция и компоновка lab4.asm

Трансляция файла lab4.asm с помощью команды `nasm -f elf lab4.asm` (рис. Рисунок 4.2).

```
clanj1@clanj1:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
clanj1@clanj1:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  lab4.asm  lab4.o  list.lst  main  obj.o
```

Рисунок 4.2: Трансляция lab4.asm

Вывод `ls` показывает появление файла lab4.o, что подтверждает успешную трансляцию lab4.asm и готовность файла к этапу компоновки.

—

Компоновка объектного файла lab4.o в исполняемую программу lab4 и последующий запуск для проверки вывода (рис. Рисунок 4.3).

```
clanj1@clanj1:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
clanj1@clanj1:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  lab4  lab4.asm  lab4.o  list.lst  main  obj.o
clanj1@clanj1:~/work/arch-pc/lab04$ ./lab4
Лань Цяньин
```

Рисунок 4.3: Компоновка и запуск lab4

Вывод «Лань Цяньин» подтверждает

## 4.3 3.3 Копирование файлов в каталог курса и загрузка на GitHub

Копирование файлов hello.asm и lab4.asm в итоговый каталог лабораторной работы курса (рис. Рисунок 4.4).

```
clanj1@clanj1:~/work/arch-pc/lab04$ cp hello.asm lab4.asm ~/work/study/2025-2026
/Архитектура\ компьютера/arch-pc/labs/lab04/
```

Рисунок 4.4: Копирование файлов в каталог курса

Проверка содержимого итогового каталога лабораторной работы после копирования файлов (рис. Рисунок 4.5).

```
clanji1@clanji1:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs$ cd ~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab04
clanji1@clanji1:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab04$ ls
hello.asm  lab4.asm
```

Рисунок 4.5: Проверка каталога lab04

Вывод ls показывает наличие hello.asm и lab4.asm

Отправка файлов лабораторной работы в репозиторий GitHub (рис. Рисунок 4.6).

```
clanji1@clanji1:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ git add .
clanji1@clanji1:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ git commit
--amend
[master bffbf35] feat(main): add hello.asm and lab4.asm for lab04
Date: Thu Dec 11 15:02:29 2025 +0300
6 files changed, 17 deletions(-)
delete mode 100755 labs/lab04/hello
delete mode 100644 labs/lab04/hello.o
delete mode 100755 labs/lab04/lab4
delete mode 100644 labs/lab04/lab4.o
delete mode 100644 labs/lab04/list.lst
delete mode 100644 labs/lab04/obj.o
clanji1@clanji1:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 2 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 414 bytes | 414.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:lanqianying/study_2025-2026_arch-pc.git
7a2ff88..bffbf35 master -> master
```

Рисунок 4.6: Коммит и push на GitHub

Вывод git показывает успешное создание коммита и сообщение «master -> master», что подтверждает корректную отправку файлов в GitHub.

## 5 4. Выводы

В ходе лабораторной работы были освоены основные этапы создания программ на языке ассемблера NASM: трансляция, компоновка и запуск. Были получены практические навыки работы с утилитами `nasm` и `ld`, а также выполнена модификация программы и размещение файлов в репозитории GitHub.