

# **Отчёт по лабораторной работе №4**

**Дисциплина: Архитектура компьютера**

Ким Денис Вячеславович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>5</b>	<b>Выводы</b>	<b>12</b>
	<b>Список литературы</b>	<b>13</b>

# Список иллюстраций

4.1	Создание каталога . . . . .	8
4.2	Создание и открытие текстового файла . . . . .	8
4.3	Ввод текста . . . . .	8
4.4	Компиляция текста . . . . .	9
4.5	Ввод команды . . . . .	9
4.6	Передача файла на обработку . . . . .	9
4.7	Выполнение команды . . . . .	9
4.8	Запуск программы Hello . . . . .	9
4.9	Создание копии файла . . . . .	10
4.10	Внос изменений . . . . .	10
4.11	Трансляция текста . . . . .	10
4.12	Копирование файлов . . . . .	10
4.13	Загрузка файлов на Github . . . . .	11

# Список таблиц

3.1	Описание некоторых каталогов файловой системы GNU Linux . . .	7
-----	---	---

# 1 Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

## **2 Задание**

В процессе данной лабораторной работы мне предстоит познакомиться с процедурой компиляции и сборки программ, написанных на ассемблере NASM, его языком, а также процессом создания и обработки программы на этом языке.

## 3 Теоретическое введение

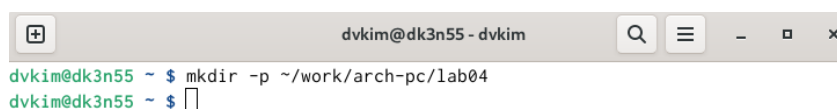
Здесь описываются теоретические аспекты, связанные с выполнением работы. Например, в табл. 3.1 приведено краткое описание стандартных каталогов Unix.

Таблица 3.1: Описание некоторых каталогов файловой системы GNU Linux	
Имя каталога	Описание каталога
/	Корневая директория, содержащая всю файловую систему
/bin	Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям
/etc	Общесистемные конфигурационные файлы и файлы конфигурации установленных программ
/home	Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя
/media	Точки монтирования для сменных носителей
/root	Домашняя директория пользователя root
/tmp	Временные файлы
/usr	Вторичная иерархия для данных пользователя

Более подробно про Unix см. в [1–4].

## 4 Выполнение лабораторной работы

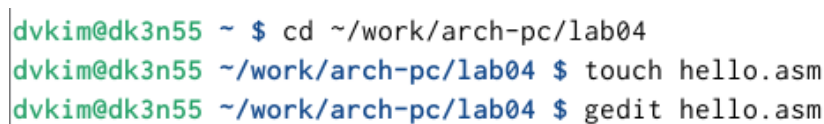
Создаём каталог для работы с программами на языке ассемблера NASM (рис. 4.1):



```
dvkim@dk3n55 ~ $ mkdir -p ~/work/arch-pc/lab04
dvkim@dk3n55 ~ $
```

Рис. 4.1: Создание каталога

Переходим в него, создаём текстовый файл hello.asm и открываем его с помощью gedit (рис. 4.2):



```
dvkim@dk3n55 ~ $ cd ~/work/arch-pc/lab04
dvkim@dk3n55 ~/work/arch-pc/lab04 $ touch hello.asm
dvkim@dk3n55 ~/work/arch-pc/lab04 $ gedit hello.asm
```

Рис. 4.2: Создание и открытие текстового файла

Вводим текст в файл (рис. 4.3):



```
1 SECTION .data
2     hello: DB 'Hello world!',10
3
4     helloLen: EQU $-hello ; Длина строки hello
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
```

Рис. 4.3: Ввод текста



Вводим команды для компиляции текста. Проверяем правильность (рис. 4.4):

```
dvkim@dk3n55 ~/work/arch-pc/lab04 $ nasm -f elf hello.asm
dvkim@dk3n55 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o
```

Рис. 4.4: Компиляция текста

Выполняем следующую команду. Проверяем правильность: (рис. 4.5):

```
dvkim@dk3n55 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
dvkim@dk3n55 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o  list.lst  obj.o
```

Рис. 4.5: Ввод команды

Передаём объектный файл на обработку компоновщику (рис. 4.6):

```
dvkim@dk3n55 ~/work/arch-pc/lab04 $ ld -m elf_i386 hello.o -o hello
dvkim@dk3n55 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  list.lst  obj.o
```

Рис. 4.6: Передача файла на обработку

Ключ -o с последующим значением задаёт в данном случае имя создаваемого исполняемого файла. Выполняем следующую команду: (рис. 4.7):

```
dvkim@dk3n55 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
dvkim@dk3n55 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
```

Рис. 4.7: Выполнение команды

Запуск программы Hello (рис. 4.8):

```
dvkim@dk3n55 ~/work/arch-pc/lab04 $ ./hello
Hello world!
```

Рис. 4.8: Запуск программы Hello

Выполняем задания для самостоятельной работы. В каталоге ~/work/arch-pc/lab04 с помощью команды cp создаем копию файла hello.asm с именем lab4.asm (рис. 4.9):

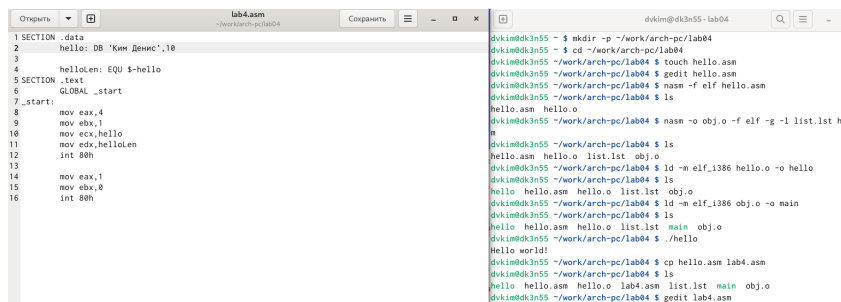
```

dvkim@dk3n55 ~/work/arch-pc/lab04 $ cp hello.asm lab4.asm
dvkim@dk3n55 ~/work/arch-pc/lab04 $ ls
hello    hello.o  lab4.asm  list.lst  main  obj.o
dvkim@dk3n55 ~/work/arch-pc/lab04 $

```

Рис. 4.9: Создание копии файла

С помощью любого текстового редактора вносим изменения в текст программы в файле lab4.asm так, чтобы вместо Hello world! на экран выводилась строка с нашими фамилией и именем (рис. 4.10):



```

1 SECTION .data
2     hello: DB 'Ким Денис',10
3
4     hellolen: EQU $-hello
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

```

```

dvkim@dk3n55 ~ $ mkdir -p ~/work/arch-pc/lab04
dvkim@dk3n55 ~ $ cd ~/work/arch-pc/lab04
dvkim@dk3n55 ~/work/arch-pc/lab04 $ touch hello.asm
dvkim@dk3n55 ~/work/arch-pc/lab04 $ gedit hello.asm
dvkim@dk3n55 ~/work/arch-pc/lab04 $ nasm -f elf hello.asm
hello.asm hello.o
dvkim@dk3n55 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst h
ello.asm
dvkim@dk3n55 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o  list.lst  obj.o
dvkim@dk3n55 ~/work/arch-pc/lab04 $ ld -m elf_i386 hello.o -o hello
dvkim@dk3n55 ~/work/arch-pc/lab04 $ ls
hello      hello.o  list.lst  main  obj.o
dvkim@dk3n55 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
dvkim@dk3n55 ~/work/arch-pc/lab04 $ ls
hello      hello.o  list.lst  main  obj.o
dvkim@dk3n55 ~/work/arch-pc/lab04 $ ./hello
Ким Денис!
dvkim@dk3n55 ~/work/arch-pc/lab04 $ cp hello.asm lab4.asm
dvkim@dk3n55 ~/work/arch-pc/lab04 $ ls
hello    hello.o  lab4.asm  list.lst  main  obj.o
dvkim@dk3n55 ~/work/arch-pc/lab04 $ gedit lab4.asm

```

Рис. 4.10: Внос изменений

Оттранслируем полученный текст программы lab4.asm в объектный файл. Выполняем компоновку объектного файла и запускаем получившийся исполняемый файл (рис. 4.11):

```

dvkim@dk3n55 ~/work/arch-pc/lab04 $ nasm -f elf lab4.asm
dvkim@dk3n55 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst lab4.asm
dvkim@dk3n55 ~/work/arch-pc/lab04 $ ld -m elf_i386 lab4.o -o lab4
dvkim@dk3n55 ~/work/arch-pc/lab04 $ ls
hello    hello.o  lab4  lab4.o  list.lst  main  obj.o
dvkim@dk3n55 ~/work/arch-pc/lab04 $ ./lab4
Ким Денис
dvkim@dk3n55 ~/work/arch-pc/lab04 $

```

Рис. 4.11: Трансляция текста

Копируем файлы hello.asm и lab4.asm в наш локальный репозиторий в каталог ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/. (рис. 4.12):

Копирование файлов

Рис. 4.12: Копирование файлов

Загружаем файлы на Github. (рис. 4.13):

```
dvkim@dk3n55 ~ $ cd ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/
dvkim@dk3n55 ~/work/study/2024-2025/Архитектура компьютера/arch-pc $ git add .
dvkim@dk3n55 ~/work/study/2024-2025/Архитектура компьютера/arch-pc $ git commit -am 'labs'
[master b7cf970] labs
14 files changed, 32 insertions(+)
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/lab4.asm
create mode 100644 labs/lab04/report/image/1.jpg
create mode 100644 labs/lab04/report/image/10.jpg
create mode 100644 labs/lab04/report/image/11.jpg
create mode 100644 labs/lab04/report/image/12.png
create mode 100644 labs/lab04/report/image/2.jpg
create mode 100644 labs/lab04/report/image/3.jpg
create mode 100644 labs/lab04/report/image/4.jpg
create mode 100644 labs/lab04/report/image/5.jpg
create mode 100644 labs/lab04/report/image/6.jpg
create mode 100644 labs/lab04/report/image/7.jpg
create mode 100644 labs/lab04/report/image/8.jpg
create mode 100644 labs/lab04/report/image/9.jpg
dvkim@dk3n55 ~/work/study/2024-2025/Архитектура компьютера/arch-pc $ git push
Перечисление объектов: 25, готово.
Подсчет объектов: 100% (25/25), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (20/20), готово.
Запись объектов: 100% (20/20), 420.32 Киб | 3.28 Миб/с, готово.
Total 20 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 3 local objects.
To github.com:dvkim2306/study_2024-2025_arh--pc.git
 5fb2f30..b7cf970 master -> master
dvkim@dk3n55 ~/work/study/2024-2025/Архитектура компьютера/arch-pc $
```

Рис. 4.13: Загрузка файлов на Github

## **5 Выводы**

В ходе данной работы я освоил процедуры компиляции и сборки программ, написанных на ассемблере NASM. Теперь я могу вводить текст, транслировать и компоновать его, а также запускать программу (выводить текст) с помощью данного ассемблера.

## Список литературы

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.
2. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.