

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4

дисциплина: Архитектура компьютера

Студент: Тимаков Макарий Владимирович

Группа: НПИбд 02-25

МОСКВА

2025г.

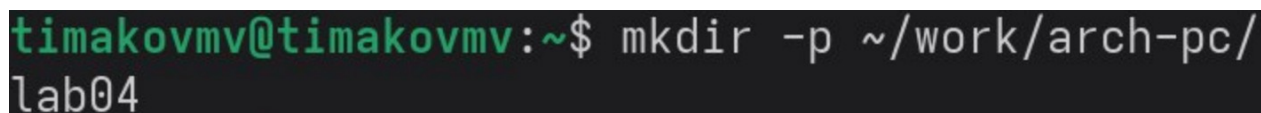
1. Цель работы

Освоить процедуры компиляции и сборки программ, познакомиться с языком ассемблера NASM.

2. Выполнение лабораторной работы

2.1 Программа Hello world!

Создаем каталог для работы с программами на языке ассемблера NASM



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

Рисунок 2.1: Создаем каталоги с помощью команды mkdir

Переходим в созданный каталог



```
timakovmv@timakovmv:~$ cd ~/work/arch-pc/lab04
```

Рисунок 2.2: Переходим в каталог с помощью команды cd

Создаем текстовый файл



```
timakovmv@timakovmv:~/work/arch-pc/lab04$ touch hello.asm
timakovmv@timakovmv:~/work/arch-pc/lab04$ ls
hello.asm
```

Рисунок 2.3: Создаем текстовый файл hello.asm

Открываем данный файл в текстовом редакторе

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

Рисунок 2.4: Открываем файл и заполняем его по пр

2.2 Транслятор NASM

Преобразуем текст программы в объектный код , проверяем созданся ли объектный файл с помощью команды ls

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

Рисунок 2.5: Используем команду nasm

2.3 Расширенный синтаксис командной строки NASM

Компилируем исходный файл

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

Рисунок 2.6: Преобразуем файл hello.asm в obj.o

Проверяем, как сработала команда

```
timakovmv@timakovmv:~/work/arch-pc/lab04$ ls  
hello.asm hello.o list.lst obj.o
```

Рисунок 2.7: Проверяем создание файла командой ls

2.4 Компоновщик LD

Передаем объектный файл на обработку компоновщику

```
timakovmv@timakovmv:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
```

Рисунок 2.8: Используем команду ld

Проверяем создался ли исполняемый файл hello

```
timakovmv@timakovmv:~/work/arch-pc/lab04$ ls  
hello hello.asm hello.o list.lst obj.o
```

Рисунок 2.9: Используем команду ls

Передаем объектный файл на обработку компоновщику

```
timakovmv@timakovmv:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
```

Рисунок 2.10: Используем команду ld, создавая файл main

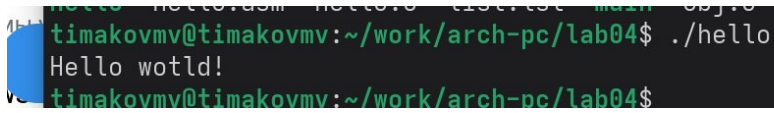
Проверяем создался ли исполняемый файл hello

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

Рисунок 2.11: Используем команду ls

2.5 Запуск исполняемого файла

Запускаем на выполнение созданный исполняемый файл




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

Рисунок 2.12: Используем команду ./hello

3. Задание для самостоятельной работы

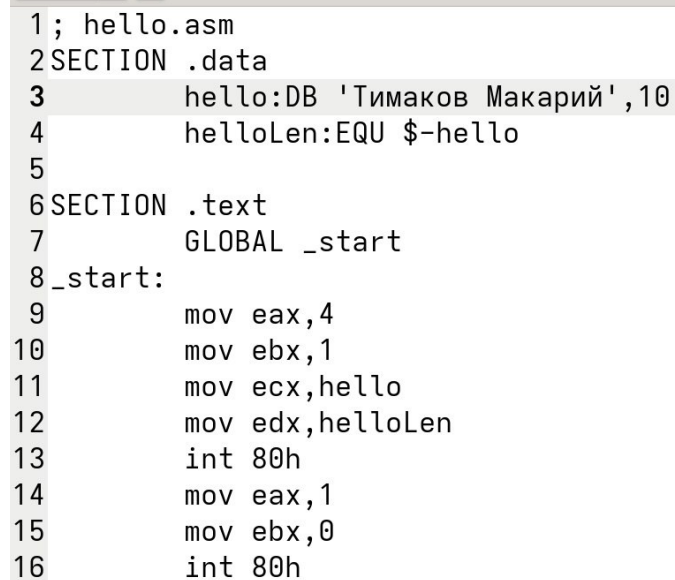
Создаем копию файла hello.asm



```
timakovmv@timakovmv:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
timakovmv@timakovmv:~/work/arch-pc/lab04$
```

Рисунок 3.1: Используем команду cp

Открываем файл и редактируем его



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

Рисунок 3.2: Редактируем файл для своего имени и фамилии

Прописываем те же команды, что и с первой программой

```
timakovmv@timakovmv:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
timakovmv@timakovmv:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst lab4.asm
timakovmv@timakovmv:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o hello
timakovmv@timakovmv:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
timakovmv@timakovmv:~/work/arch-pc/lab04$ ./hello
Тимаков Макарий
timakovmv@timakovmv:~/work/arch-pc/lab04$
```

Рисунок 3.3: Прописываем команды для работы файла и запускаем программу

Копируем файлы в локальный репозиторий

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

Рисунок 3.4: Копируем файлы в каталог с ЛР4

Переходим в каталог лабораторных работ и загружаем файлы на Github

```
timakovmv@timakovmv:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ git add .
timakovmv@timakovmv:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ git commit -am 'feat(main): add files lab4'
[master 9495ce7] feat(main): add files lab4
2 files changed, 32 insertions(+)
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/lab4.asm
timakovmv@timakovmv:~/work/study/2025-2026/Архитектура компьютера/arch-pc$ git push
```

Рисунок 3.5 Загружаем файлы

4. Выводы

В ходе лабораторной работы мы изучили основы работы с ассемблером NASM в операционной системе Linux. Мы приобрели практические навыки написания, трансляции и выполнения низкоуровневых программ на языке ассемблера.