

Отчёт по лабораторной работе №4

Ярослав Антонович Меркулов

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Выполнение самостоятельной работы	9
4	Выводы	12

Список иллюстраций

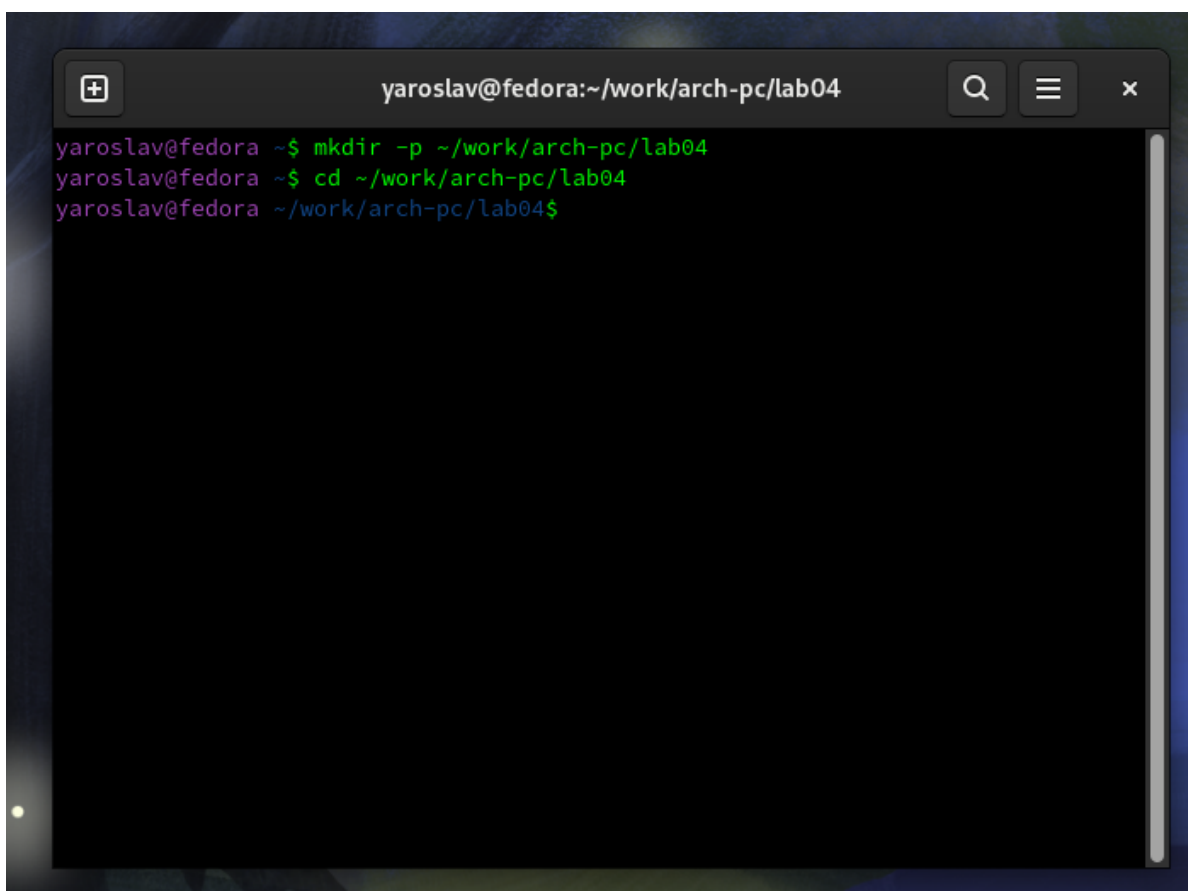
2.1	Создание каталога для работы	5
2.2	Создание файла с помощью <i>touch</i>	6
2.3	Открытый файл <i>hello.asm</i>	6
2.4	Готовый текстовый файл	6
2.5	Объектный файл	7
2.6	Объектный файл и файл листинга	7
2.7	Готовый исполняемый файл	7
2.8	Другой исполняемый файл	7
2.9	<i>./hello</i> и вывод	8
3.1	Копирование <i>hello.asm</i>	9
3.2	Файл <i>lab04.asm</i>	9
3.3	Трансляция и компоновка	10
3.4	Работа программы	10
3.5	Копирование файлов	10
3.6	Проверка наличия файлов	10
3.7	Загрузка на <i>Git</i>	11

1 Цель работы

Научится компилировать и собирать программы, написанные на ассемблере
NASM

2 Выполнение лабораторной работы

1. Создаём каталог для работы с программами на языке ассемблера NASM.

A screenshot of a terminal window with a dark background. The window title bar shows 'yaroslav@fedora:~/work/arch-pc/lab04'. The terminal displays three lines of commands and their outputs: 'mkdir -p ~/work/arch-pc/lab04', 'cd ~/work/arch-pc/lab04', and the prompt changes to '~/work/arch-pc/lab04\$'.

```
yaroslav@fedora ~$ mkdir -p ~/work/arch-pc/lab04
yaroslav@fedora ~$ cd ~/work/arch-pc/lab04
yaroslav@fedora ~/work/arch-pc/lab04$
```

Рис. 2.1: Создание каталога для работы

2. Создаём текстовый файл с именем *hello.asm*.

```
yaroslav@fedora ~/work/arch-pc/lab04$ touch hello.asm
yaroslav@fedora ~/work/arch-pc/lab04$
```

Рис. 2.2: Создание файла с помощью *touch*

3. Открываем созданный файл с помощью *gedit*.

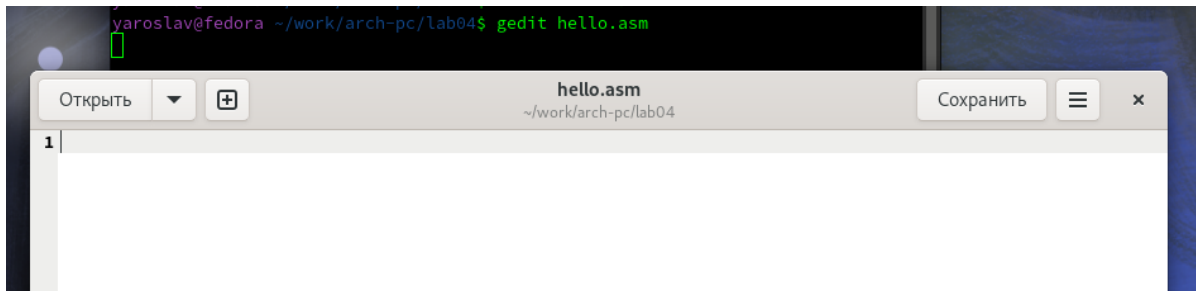


Рис. 2.3: Открытый файл *hello.asm*

4. Вводим заданный текст.

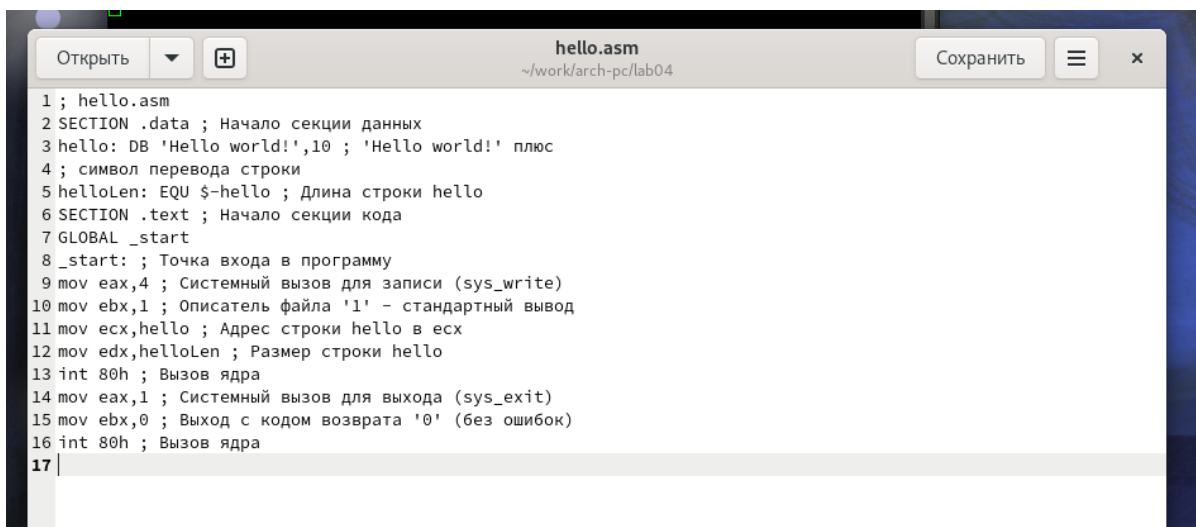


Рис. 2.4: Готовый текстовый файл

5. Выполняем трансляцию с помощью *nasm*.

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

Рис. 2.5: Объектный файл

6. Снова выполняем трансляцию, но с другими параметрами.

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

Рис. 2.6: Объектный файл и файл листинга

7. Переходим к компоновке. Используем *ld*.

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

Рис. 2.7: Готовый исполняемый файл

8. Используем компоновку с другими параметрами. Получаем исполняемый файл с другим названием.

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

Рис. 2.8: Другой исполняемый файл

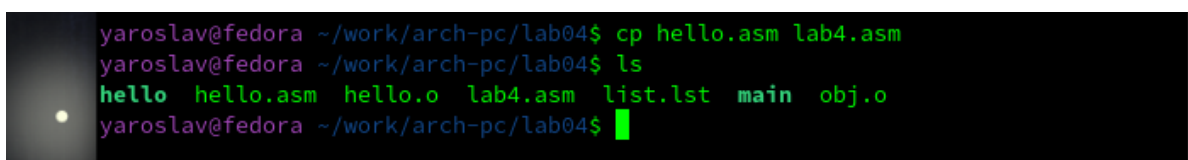
9. Запускаем файл.

```
yaroslav@fedora ~/work/arch-pc/lab04$ ./hello
Hello world!
yaroslav@fedora ~/work/arch-pc/lab04$
```

Рис. 2.9: ./hello и вывод

3 Выполнение самостоятельной работы

1. Копируем *hello.asm* и переименовываем в *lab04.asm*.



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

Рис. 3.1: Копирование *hello.asm*

2. Открываем с помощью *gedit* файл *lab04.asm*. Меняем *Hello world!* на *Меркулов Ярослав*.



```
yaroslav@fedora ~/work/arch-pc/lab04$ gedit lab4.asm

Открыть ▼ + lab4.asm ~/work/arch-pc/lab04 Сохранить ≡

1 ; hello.asm
2 SECTION .data ; Начало секции данных
3 hello: DB 'Меркулов Ярослав',10 ; 'Hello world!' плюс
4 ; символ перевода строки
5 helloLen: EQU $-hello ; Длина строки hello
6 SECTION .text ; Начало секции кода
7 GLOBAL _start
8 _start: ; Точка входа в программу
9 mov eax,4 ; Системный вызов для записи (sys_write)
10 mov ebx,1 ; Описатель файла '1' - стандартный вывод
11 mov ecx,hello ; Адрес строки hello в ecx
12 mov edx,helloLen ; Размер строки hello
13 int 80h ; Вызов ядра
14 mov eax,1 ; Системный вызов для выхода (sys_exit)
15 mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
16 int 80h ; Вызов ядра
17
```

Рис. 3.2: Файл *lab04.asm*

3. Оттранслируем и скомпилируем файл.

```
yaroslav@fedora ~/work/arch-pc/lab04$ nasm -f elf lab4.asm
yaroslav@fedora ~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4.asm lab4.o list.lst main obj.o
yaroslav@fedora ~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
yaroslav@fedora ~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4 lab4.asm lab4.o list.lst main obj.o
```

Рис. 3.3: Трансляция и компоновка

4. Запускаем получившийся файл.

```
yaroslav@fedora ~/work/arch-pc/lab04$ ./lab4
Меркулов Ярослав
yaroslav@fedora ~/work/arch-pc/lab04$
```

Рис. 3.4: Работа программы

5. Копируем текстовые файлы в локальный репозиторий.

```
yaroslav@fedora ~/work/arch-pc/lab04$ cp lab4.asm hello.asm ~/work/study/2024-2025/Архитектура\ компьютера/arch-pc/labs/lab04
```

Рис. 3.5: Копирование файлов

6. Проверяем наличие файлов.

```
yaroslav@fedora ~/work/arch-pc/lab04$ cd ~/work/study/2024-2025/Архитектура\ компьютера/arch-pc/labs/lab04
yaroslav@fedora ~/work/study/2024-2025/Архитектура\ компьютера/arch-pc/labs/lab04$ ls
hello.asm lab4.asm presentation report
```

Рис. 3.6: Проверка наличия файлов

7. Загружаем всё на *Github*.

```

yaroslav@fedora ~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$ cd ~/work/study/2024-2025/Архитекту
yaroslav@fedora ~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git add .
yaroslav@fedora ~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git commit -am 'lab4'
[master ff3fc6a] lab4
19 files changed, 127 insertions(+)
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/lab4.asm
create mode 100644 labs/lab04/report/image/pic1.png
create mode 100644 labs/lab04/report/image/pic10.png
create mode 100644 labs/lab04/report/image/pic11.png
create mode 100644 labs/lab04/report/image/pic12.png
create mode 100644 labs/lab04/report/image/pic13.png
create mode 100644 labs/lab04/report/image/pic14.png
create mode 100644 labs/lab04/report/image/pic15.png
create mode 100644 labs/lab04/report/image/pic2.png
create mode 100644 labs/lab04/report/image/pic3.png
create mode 100644 labs/lab04/report/image/pic4.png
create mode 100644 labs/lab04/report/image/pic5.png
create mode 100644 labs/lab04/report/image/pic6.png
create mode 100644 labs/lab04/report/image/pic7.png
create mode 100644 labs/lab04/report/image/pic8.png
create mode 100644 labs/lab04/report/image/pic9.png
delete mode 100644 labs/lab04/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab04/report/report4.md
yaroslav@fedora ~/work/study/2024-2025/Архитектура компьютера/arch-pc$ git push
Перечисление объектов: 29, готово.
Подсчет объектов: 100% (29/29), готово.
При сжатии изменений используется до 8 потоков.

```

Рис. 3.7: Загрузка на *Git*

4 Выводы

Были получены знания и умения по работе с NASM (компиляции и сборки).