

# **Отчёта по лабораторной работе №4**

**Создание и процесс обработки программ на языке ассемблера NASM**

Жозе Рамос Домингуш

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
3.1	Программа Hello world! . . . . .	6
3.2	Транслятор NASM . . . . .	6
3.3	Расширенный синтаксис командной строки NASM . . . . .	7
3.4	Компоновщик LD . . . . .	7
3.5	Запуск исполняемого файла . . . . .	8
3.6	Задание для самостоятельной работы . . . . .	8
<b>4</b>	<b>Выводы</b>	<b>10</b>

## Список иллюстраций

3.1	Создаем каталоги с помощью команды <code>mkdir</code> . . . . .	6
3.2	Переходим в каталог с помощью команды <code>cd</code> . . . . .	6
3.3	Создаем текстовый файл <code>hello.asm</code> . . . . .	6
3.4	Открываем файл и заполняем его по примеру . . . . .	6
3.5	Используем команду <code>nasm</code> . . . . .	7
3.6	Проверяем работу команды . . . . .	7
3.7	Преобразуем файл <code>hello.asm</code> в <code>obj.o</code> . . . . .	7
3.8	Проверяем создание файла командой <code>ls</code> . . . . .	7
3.9	Используем команду <code>ld</code> . . . . .	7
3.10	Используем команду <code>ls</code> . . . . .	7
3.11	Используем команду <code>ld</code> , создавая файл <code>main</code> . . . . .	8
3.12	Используем команду <code>ls</code> . . . . .	8
3.13	Используем команду <code>./hello</code> . . . . .	8
3.14	Используем команду <code>cp</code> . . . . .	8
3.15	Открываем файл в текстовом редакторе . . . . .	8
3.16	Редактируем файл для своего имени и фамилии . . . . .	8
3.17	Прописываем команды для работы файла и запускаем программу	8
3.18	Копируем файлы в каталог с ЛР4 . . . . .	9
3.19	Загружаем файлы . . . . .	9

# 1 Цель работы

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

## 2 Задание

Написать 2 программы(Hello world, lab4(Имя Фамилия))

## 3 Выполнение лабораторной работы

### 3.1 Программа Hello world!

Создаем каталог для работы с программами на языке ассемблера NASM (рис. fig. 3.1).

Создаем каталоги с помощью команды `mkdir`

Рис. 3.1: Создаем каталоги с помощью команды `mkdir`

Переходим в созданный каталог (рис. fig. 3.2).

Переходим в каталог с помощью команды `cd`

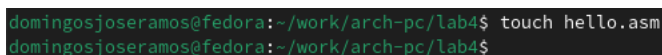
Рис. 3.2: Переходим в каталог с помощью команды `cd`

Создаем текстовый файл (рис. fig. 3.3).

Создаем текстовый файл `hello.asm`

Рис. 3.3: Создаем текстовый файл `hello.asm`

Открываем данный файл в текстовом редакторе (рис. fig. 3.4).



```
domingosjoseramos@fedora:~/work/arch-pc/lab4$ touch hello.asm
domingosjoseramos@fedora:~/work/arch-pc/lab4$
```

Рис. 3.4: Открываем файл и заполняем его по примеру

### 3.2 Транслятор NASM

Преобразуем текст программы в объектный код (рис. fig. 3.5).

Используем команду `nasm`

Рис. 3.5: Используем команду `nasm`

Проверяем создан ли объектный файл с помощью команды `ls` (рис. fig. 3.6).

Проверяем работу команды

Рис. 3.6: Проверяем работу команды

### 3.3 Расширенный синтаксис командной строки NASM

Компилируем исходный файл (рис. fig. 3.7).

Преобразуем файл `hello.asm` в `obj.o`

Рис. 3.7: Преобразуем файл `hello.asm` в `obj.o`

Проверяем, как сработала команда (рис. fig. 3.8).

Проверяем создание файла командой `ls`

Рис. 3.8: Проверяем создание файла командой `ls`

### 3.4 компоновщик LD

Передаем объектный файл на обработку компоновщику (рис. fig. 3.9).

Используем команду `ld`

Рис. 3.9: Используем команду `ld`

Проверяем создан ли исполняемый файл `hello` (рис. fig. 3.10).

Используем команду `ls`

Рис. 3.10: Используем команду `ls`

Передаем объектный файл на обработку компоновщику (рис. fig. 3.11).

Используем команду `ld`, создавая файл `main`

Рис. 3.11: Используем команду `ld`, создавая файл `main`

Проверяем создан ли исполняемый файл `hello` (рис. fig. 3.12).

Используем команду `ls`

Рис. 3.12: Используем команду `ls`

### 3.5 Запуск исполняемого файла

Запускаем на выполнение созданный исполняемый файл (рис. fig. 3.13).

Используем команду `./hello`

Рис. 3.13: Используем команду `./hello`

### 3.6 Задание для самостоятельной работы

Создаем копию файла `hello.asm` (рис. fig. 3.14).

Используем команду `cp`

Рис. 3.14: Используем команду `cp`

Открываем файл и редактируем его (рис. fig. 3.15).

Открываем файл в текстовом редакторе

Рис. 3.15: Открываем файл в текстовом редакторе

Редактируем файл для своего имени и фамилии

Рис. 3.16: Редактируем файл для своего имени и фамилии

Прописываем те же команды, что и с первой программой (рис. fig. 3.17).

Прописываем команды для работы файла и запускаем программу

Рис. 3.17: Прописываем команды для работы файла и запускаем программу



Копируем файлы в локальный репозиторий (рис. fig. 3.18).

Копируем файлы в каталог с ЛР4

Рис. 3.18: Копируем файлы в каталог с ЛР4

Переходим в каталог лабораторных работ и загружаем файлы на Github (рис. fig. 3.19).

Загружаем файлы

Рис. 3.19: Загружаем файлы

## 4 Выводы

Мы познакомились с языком ассемблера NASM и создали две работающих программы.