

# **Отчёта по лабораторной работе №4**

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

Камбунду Паулине

# 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 . . . . .	7
3.3	Расширенный синтаксис командной строки NASM . . . . .	7
3.4	Компоновщик LD . . . . .	8
3.5	Запуск исполняемого файла . . . . .	8
3.6	Задание для самостоятельной работы . . . . .	9
<b>4</b>	<b>Выводы</b>	<b>11</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 . . . . .	7
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> . . . . .	8
3.9	Используем команду <code>ld</code> . . . . .	8
3.10	Используем команду <code>ls</code> . . . . .	8
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> . . . . .	9
3.14	Используем команду <code>cp</code> . . . . .	9
3.15	Открываем файл в текстовом редакторе . . . . .	9
3.16	Редактируем файл для своего имени и фамилии . . . . .	9
3.17	Прописываем команды для работы файла и запускаем программу	10
3.18	Копируем файлы в каталог с ЛР4 . . . . .	10
3.19	Загружаем файлы . . . . .	10

# 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).

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

Рис. 3.1: Создаем каталоги с помощью команды mkdir

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

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

Рис. 3.2: Переходим в каталог с помощью команды cd

Создаем текстовый файл (рис. fig. 3.3).

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

Рис. 3.3: Создаем текстовый файл hello.asm

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



Рис. 3.4: Открываем файл и заполняем его по примеру

## 3.2 Транслятор NASM

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

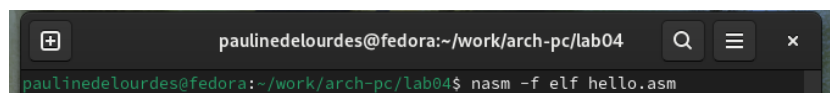


Рис. 3.5: Используем команду nasm

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

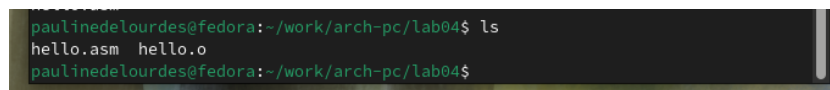


Рис. 3.6: Проверяем работу команды

## 3.3 Расширенный синтаксис командной строки NASM

Компилируем исходный файл (рис. fig. 3.7).

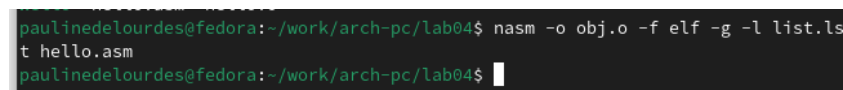


Рис. 3.7: Преобразуем файл hello.asm в obj.o

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

```

paulinedelourdes@fedora:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o
paulinedelourdes@fedora:~/work/arch-pc/lab04$

```

Рис. 3.8: Проверяем создание файла командой ls

### 3.4 Компоновщик LD

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

```

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

```

Рис. 3.9: Используем команду ld

Проверяем создался ли исполняемый файл hello (рис. fig. 3.10).

```

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

```

Рис. 3.10: Используем команду ls

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

```

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

```

Рис. 3.11: Используем команду ld, создавая файл main

Проверяем создался ли исполняемый файл hello (рис. fig. 3.12).

```

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

```

Рис. 3.12: Используем команду ls

### 3.5 Запуск исполняемого файла

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



```

hello: hello.oasm hello.o lab4.oasm lab4.o elf.oasm obj.o
paulinedelourdes@fedora:~/work/arch-pc/lab04$ ./hello
Hello world!

```

Рис. 3.13: Используем команду ./hello

## 3.6 Задание для самостоятельной работы

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

```

hello world:
paulinedelourdes@fedora:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
paulinedelourdes@fedora:~/work/arch-pc/lab04$ gedit lab4.asm

```

Рис. 3.14: Используем команду cp

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

```

paulinedelourdes@fedora:~/work/arch-pc/lab04$ gedit lab4.asm
paulinedelourdes@fedora:~/work/arch-pc/lab04$ nasm -f elf lab4.asm

```

Рис. 3.15: Открываем файл в текстовом редакторе

```

1 ; hello.asm
2 SECTION .data ; Начало секции данных
3     hello: DB 'Hello world!',10 ; 'Камбунду Паулине!' 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
15    mov eax,1 ; Системный вызов для выхода (sys_exit)
16    mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
17    int 80h ; Вызов ядра

```

Рис. 3.16: Редактируем файл для своего имени и фамилии

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

```

paulinedelourdes@fedora:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
paulinedelourdes@fedora:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.ls
lab4.asm
paulinedelourdes@fedora:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o hello
paulinedelourdes@fedora:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
ld: unrecognized emulation mode: elf_i386
Supported emulations: elf_x86_64 elf32_x86_64 elf_i386 elf_iamcu i386pep i386pe
elf64bpf
paulinedelourdes@fedora:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
paulinedelourdes@fedora:~/work/arch-pc/lab04$ ./hello
Hello world!

```

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

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

```

cp: destino 'Компьютера/arch-pc/labs/lab04' : File exists or pasta inexistence
paulinedelourdes@fedora:~/work/arch-pc/lab04$ cp hello.asm ~/work/study/2024-202
5/'Архитектура Компьютера'/study_2024-2025_arh-pc/labs/lab04
paulinedelourdes@fedora:~/work/arch-pc/lab04$ cp lab4.asm ~/work/study/2024-202
5/'Архитектура Компьютера'/arh-pc/labs/lab04/

```

Рис. 3.18: Копируем файлы в каталог с ЛР4

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

```

-2025_arh-pc$ git add .
paulinedelourdes@fedora:~/work/study/2024-2025/Архитектура Компьютера/study_2024
-2025_arh-pc$ git commit -am 'feat(main): add files lab-4'
[master e7b65e3] feat(main): add files lab-4
11 files changed, 87 insertions(+)
create mode 100755 labs/lab04/lab04/hello
create mode 100644 labs/lab04/lab04/hello.asm
create mode 100644 labs/lab04/lab04/hello.o
create mode 100644 labs/lab04/lab04/lab4.asm
create mode 100644 labs/lab04/lab04/lab4.o
create mode 100644 labs/lab04/lab04/list.ls
create mode 100644 labs/lab04/lab04/list.lst
create mode 100755 labs/lab04/lab04/main
create mode 100644 labs/lab04/lab04/obj.o
create mode 100644 labs/lab04/lab04/obj.o
create mode 100644 labs/lab04/lab04/lab4.asm
paulinedelourdes@fedora:~/work/study/2024-2025/Архитектура Компьютера/study_2024
-2025_arh-pc$

```

Рис. 3.19: Загружаем файлы

## 4 Выводы

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