

# **Лабораторная работа №4**

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

Приспешкин Андрей Андреевич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>4</b>	<b>Выводы</b>	<b>13</b>
	<b>Список литературы</b>	<b>14</b>

## Список иллюстраций

3.1	Рис. 1 Создание нового подкаталога . . . . .	7
3.2	Рис. 2 Переход в созданный подкаталог . . . . .	7
3.3	Рис. 3 Создание файла hello.asm . . . . .	7
3.4	Рис. 4 Откроем файл hello.asm в текстовом редакторе neovim . . .	7
3.5	Рис. 5 Пример кода в текстовом редакторе . . . . .	8
3.6	Рис. 6 Работа транслятора . . . . .	8
3.7	Рис.7 Компиляция программы с расширенным синтаксисом . . .	8
3.8	Рис.8 Проверка работы транслятора . . . . .	9
3.9	Рис. 9 Работа компоновщика ld . . . . .	9
3.10	Рис.10 Проверка работы компоновщика . . . . .	9
3.11	Рис.11 Передача файла obj.o компоновщику и проверка его работы	9
3.12	Рис. 12 Запуск программы hello . . . . .	9
3.13	Рис.13 Копирование файла hello.asm . . . . .	10
3.14	Рис.14 Использование команды nvim для открытие файла lab4.asm	10
3.15	Рис.15 Код с внесёнными изменениями . . . . .	11
3.16	Рис.16 Компиляция программы lab4.asm . . . . .	11
3.17	Рис.17 Создание файла lab . . . . .	11
3.18	Рис.18 Результат работы программы lab . . . . .	12
3.19	Использование утилиты sr для копирования программ в рабочий каталог . . . . .	12

## Список таблиц

# 1 Цель работы

Цель данной лабораторной работы – базовая ознакомление с языком NASM, а также освоить процедуры компиляции и сборки программ написанных на этом языке.

## 2 Задание

1. Создание программы Hello world на языке NASM
2. Работа с транслятором NASM
3. Работа с компоновщиком LD
4. Запуск исполняемого файла
5. Задания для самостоятельной работы

### 3 Выполнение лабораторной работы

Создадим в рабочем каталоге work подкаталог в котором будет выполняться лабораторная работа(Рис.1).

```
aaprispeshkin:[aaprispeshkin]:~$ mkdir -p ~/work/arch-pc/lab04
```

Рис. 3.1: Рис. 1 Создание нового подкаталога

Перейдём в созданный нами подкаталог(Рис.2).

```
aaprispeshkin:[aaprispeshkin]:~$ cd ~/work/arch-pc/lab04
```

Рис. 3.2: Рис. 2 Переход в созданный подкаталог

Создадим файл hello.asm утилитой touch(Рис.3).

```
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab04$ touch hello.asm
```

Рис. 3.3: Рис. 3 Создание файла hello.asm

Редактировать файл будем в текстовом редакторе neovim(Рис.4).

```
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab04$ nvim hello.asm
```

Рис. 3.4: Рис. 4 Откроем файл hello.asm в текстовом редакторе neovim

Вставим пример кода из лабораторной работы в текстовый редактор(Рис.5).

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

Рис. 3.5: Рис. 5 Пример кода в текстовом редакторе

Переведём текст программы hello.asm в объектный код с помощью транслятора(Рис.6).

```
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab04$ nasm -f elf hello.asm
```

Рис. 3.6: Рис. 6 Работа транслятора

Введём команду которая скомпилирует файл hello.asm в файл obj.o, так же воспользуемся ключом -g и с помощью ключа -l создадим файл листинга list.lst(Рис.7).

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

Рис. 3.7: Рис.7 Компиляция программы с расширенным синтаксисом



Проверим правильность выполнения работы утилитой ls(Рис.8).

```
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o list.lst obj.o
```

Рис. 3.8: Рис.8 Проверка работы транслятора

Передадим файл hello.o компоновщику ld, зададим имя создаваемого файла с помощью ключа -o(Рис.9).

```
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
```

Рис. 3.9: Рис. 9 Работа компоновщика ld

Проверим правильность работы компоновщика утилитой ls(Рис. 10).

```
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o list.lst obj.o
```

Рис. 3.10: Рис.10 Проверка работы компоновщика

Передадим файл obj.o компоновщику. Файл будем иметь имя main, так как оно было указано после ключа -o(Рис.11).

```
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o list.lst main obj.o
```

Рис. 3.11: Рис.11 Передача файла obj.o компоновщику и проверка его работы

Запустим сделанную нами программу hello(Рис.12).

```
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab04$ ./hello
Hello world!
```

Рис. 3.12: Рис. 12 Запуск программы hello

Утилитой `cp` создадим копию `hello.asm` и назовём её `lab4.asm`. Проверим правильность копирования утилитой `ls`(Рис.13).

```
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  lab4.asm  list.lst  main  obj.o
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab04$
```

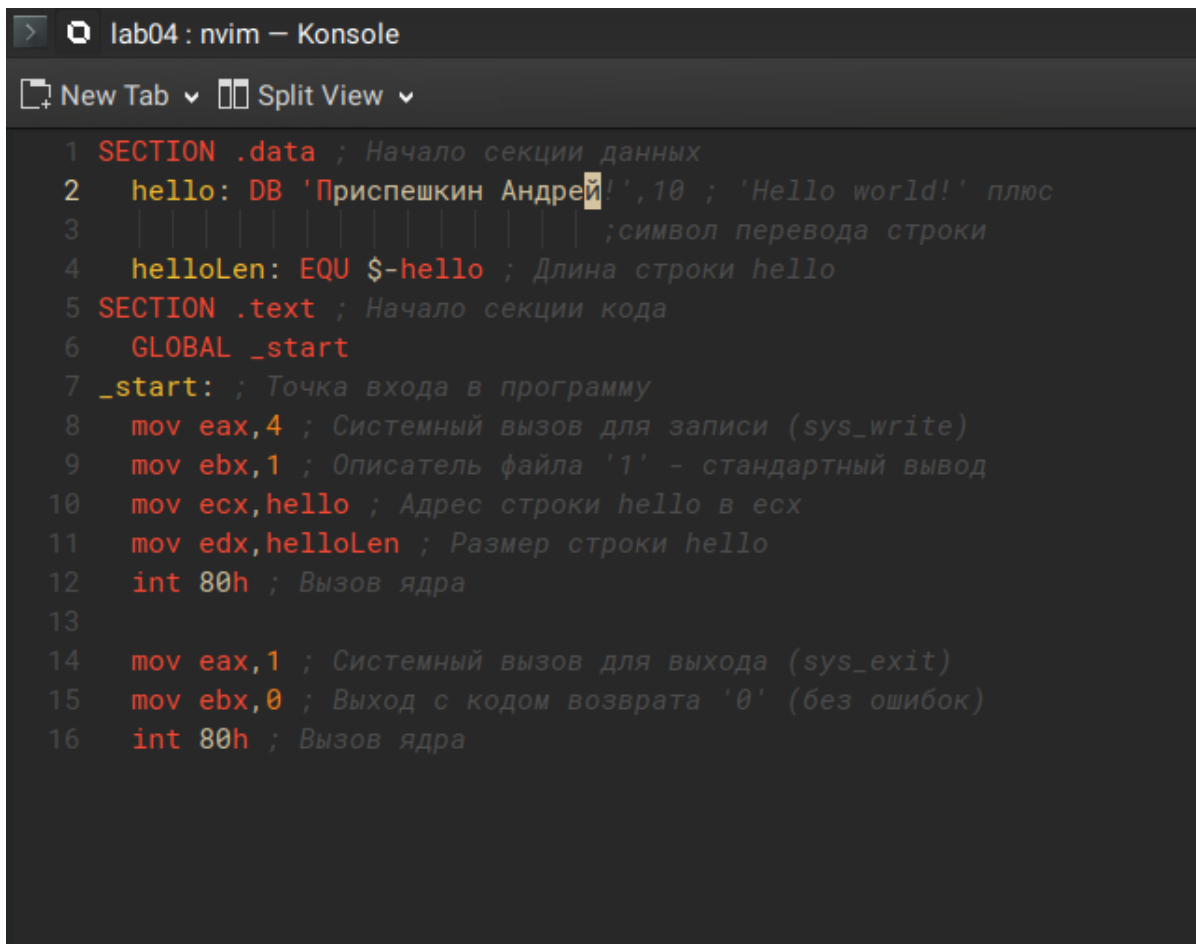
Рис. 3.13: Рис.13 Копирование файла `hello.asm`

Откроем файл `lab4.asm` в текстовом редакторе `neovim`(Рис.14).

```
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab04$ nvim lab4.asm
```

Рис. 3.14: Рис.14 Использование команды `nvim` для открытие файла `lab4.asm`

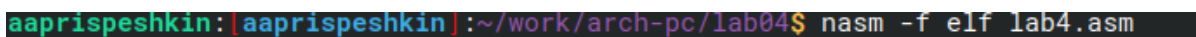
Внеёсем изменения в код так, чтобы программа `lab4.asm` выводила мои имя и фамилию(Рис.15).

A screenshot of a terminal window titled 'lab04 : nvim — Konsole'. The window shows a text editor with assembly code. The code is as follows:

```
1 SECTION .data ; Начало секции данных
2 hello: DB 'Приспешкин Андрей!',10 ; 'Hello world!' плюс
3 | | | | | | | | | | | | ;символ перевода строки
4 helloLen: EQU $-hello ; Длина строки hello
5 SECTION .text ; Начало секции кода
6 GLOBAL _start
7 _start: ; Точка входа в программу
8 mov eax,4 ; Системный вызов для записи (sys_write)
9 mov ebx,1 ; Описатель файла '1' - стандартный вывод
10 mov ecx,hello ; Адрес строки hello в ecx
11 mov edx,helloLen ; Размер строки hello
12 int 80h ; Вызов ядра
13
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
```

Рис. 3.15: Рис.15 Код с внесёнными изменениями

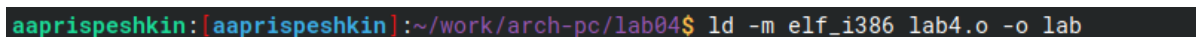
Скомпилируем получившийся код в объектный файл lab4.o(Рис.16).

A terminal screenshot showing the command to compile the assembly file into an object file:

```
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
```

Рис. 3.16: Рис.16 Компиляция программы lab4.asm

Отправим объектный код компоновщику для создания файла lab(Рис.17).

A terminal screenshot showing the command to link the object file into an executable:

```
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab
```

Рис. 3.17: Рис.17 Создание файла lab

Запустим для проверки программу lab(Рис.18).

```

aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab04$ ./lab
Приспешкин Андрей!
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab04$

```

Рис. 3.18: Рис.18 Результат работы программы lab

Скопируем программы созданные в ходе этой лабораторной работы в рабочий каталог(Рис.19).

```

aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab04$ cp hello.asm ~/work/study/2023-2024/Архитектура\ Компьютера/study_2023-2024_arh--pc/labs/lab04
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab04$ cp lab4.asm ~/work/study/2023-2024/Архитектура\ Компьютера/study_2023-2024_arh--pc/labs/lab04
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab04$

```

Рис. 3.19: Использование утилиты cp для копирования программ в рабочий каталог

## 4 Выводы

В ходе данной лабораторной работы я познакомился с языком NASM, а также научился работать с компилятором и компоновщиком для превращения программ в объектный код и запускаемые файлы соответственно.

# Список литературы

Лабораторная работа №4. Создание и процесс обработки программ на языке ассемблера NASM