

# **Лабораторная работа No4. Создание и процесс обработки программ на языке ассемблера NASM**

**Дисциплина: Архитектура компьютера**

Сущенко Алина Николаевна

# 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>Выполнение заданий для самостоятельной работы</b>	<b>10</b>
<b>5</b>	<b>Выводы</b>	<b>13</b>

## Список иллюстраций

3.1	Создание каталога с помощью 'mkdir' . . . . .	7
3.2	Переход в каталог и создание текстового файла . . . . .	7
3.3	Наличие файла . . . . .	7
3.4	Команда gedit . . . . .	7
3.5	Команда gedit . . . . .	8
3.6	Транслируем текст и видим, что всё на месте . . . . .	8
3.7	Компилировка файла и их наличие . . . . .	8
3.8	Выполняем компоновку и проверяем . . . . .	9
3.9	Выполняем компоновку и проверяем . . . . .	9
3.10	Работа команды './hello' . . . . .	9
4.1	Создание нового файла . . . . .	10
4.2	Открытие нового файла . . . . .	10
4.3	Транслируем текст программы . . . . .	11
4.4	Компиляция и проверка . . . . .	11
4.5	Выполнение компоновки и проверка . . . . .	11
4.6	Выполнение компоновки и проверка . . . . .	11
4.7	Выполнение компоновки и проверка . . . . .	11
4.8	Выполнение компоновки и проверка . . . . .	12
4.9	Переход и проверка . . . . .	12
4.10	Загрузка файлов . . . . .	12
4.11	Загрузка файлов . . . . .	12

## Список таблиц

# 1 Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM

## 2 Задание

1. Создать каталог для работы с заданиями на языке NASM.
2. Создать текстовый файл 'hello.asm' и открыть этот файл с помощью текстового редактора.
3. Ввести текст в созданный файл.
4. Транслировать текст в объектный файл.
5. Выполнить компоновку объектного файла и запустить файл.
6. Создать копию созданного файла, переименовать его в 'lab4.asm' и продолжить те же действия с файлом 'hello.asm'
7. Скопировать файлы в репозиторий.
8. Загрузить файлы на Github.

## 3 Выполнение лабораторной работы

### 1. Программа Hello world!

Создаём каталог для работы на NASM командой 'mkdir -p' (рис [3.1])

```
ansuthenko@dk6n55 ~ $ mkdir -p ~/work/arch-pc/lab04
```

Рис. 3.1: Создание каталога с помощью 'mkdir'

Переход в каталог

Переходим в каталог и создаём текстовый файл 'hello.asm' (рис [3.2])

```
ansuthenko@dk6n55 ~ $ cd ~/work/arch-pc/lab04  
ansuthenko@dk6n55 ~/work/arch-pc/lab04 $ touch hello.asm
```

Рис. 3.2: Переход в каталог и создание текстового файла

Проверяем наличие файла.(рис [3.3])

```
ansuthenko@dk6n55 ~/work/arch-pc/lab04 $ ls  
hello.asm
```

Рис. 3.3: Наличие файла

Открываем файл с помощью gedit (рис [3.4])

```
ansuthenko@dk6n55 ~/work/arch-pc/lab04 $ gedit hello.asm
```

Рис. 3.4: Команда gedit

Вводим открытый файл текст 'Hello world!' (рис [afig:005])

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

Рис. 3.5: Команда gedit

## 2. Транслятор NASM

С помощью команды 'nasm -f elf' транслируем текст программы в объектный файл и проверим наличие ([3.6])

```
ansuthenko@dk6n55 ~/work/arch-pc/lab04 $ nasm -f elf hello.asm
ansuthenko@dk6n55 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o
```

Рис. 3.6: Транслируем текст и видим, что всё на месте

## 3. Синтаксис командной строки NASM

С помощью команд компилируем исходный файл 'hello.asm' в 'obj.o' при помощи команд 'nasm -o', '-f elf -g -l' и проверяем их наличие (рис [3.7])

```
ansuthenko@dk6n55 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
ansuthenko@dk6n55 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o  list.lst  obj.o
```

Рис. 3.7: Компилировка файла и их наличие



#### 4. Компановщик LD

Передаём файл 'hello.asm' на обработку с помощью команды 'ld -m elf\_i386 .. -o' и проверяем (рис [3.8])

```
ansuthenko@dk6n55 ~/work/arch-pc/lab04 $ ld -m elf_i386 hello.o -o hello
ansuthenko@dk6n55 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  list.lst  obj.o
```

Рис. 3.8: Выполняем компоновку и проверяем

```
ansuthenko@dk6n55 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
ansuthenko@dk6n55 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
```

Рис. 3.9: Выполняем компоновку и проверяем

1)Имя файла 'main' 2)Имя объектного файла : 'obj.o'

#### 5. Запуск файла.

С помощью команды './hello' запускаем файл (рис [3.10])

```
ansuthenko@dk6n55 ~/work/arch-pc/lab04 $ ./hello
Hello world!
```

Рис. 3.10: Работа команды './hello'

## 4 Выполнение заданий для самостоятельной работы

1. С помощью команды 'ср' создаём копию файла и называем его 'lab4.asm' и проверяем его наличие (рис [4.1])

```
ansuthenko@dk6n55 ~/work/arch-pc/lab04 $ cp hello.asm lab4.asm
ansuthenko@dk6n55 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  lab4.asm  list.lst  main  obj.o
```

Рис. 4.1: Создание нового файла

С помощью gedit открываем файл 'lab4.asm' и вносим своё имя и фамилию (рис [4.2])

```
SECTION .data
    hello:  DB 'Сущенко Алина' ,10
    helloLen: EQU $-hello
SECTION .text
    GLOBAL _start
_start:
    mov eax,4
    mov ebx,1
    mov ecx,hello
    mov edx,helloLen
    int 80h
    mov eax,1
    mov ebx,0
    int 80h
```

Рис. 4.2: Открытие нового файла

С помощью 'nasm -f elf' транслируем текст программы в файл и проверяем (рис [4.3])

```

ansuthenko@dk6n55 ~/work/arch-pc/lab04 $ nasm -f elf lab4.asm
ansuthenko@dk6n55 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  lab4.asm  lab4.o  list4.lst  list.lst  main  obj4.o  obj.o

```

Рис. 4.3: Транслируем текст программы

Компилируем исходный файл с помощью 'nasm -o' и '-f elf -g -l' и проверяем (рис

```

ansuthenko@dk6n55 ~/work/arch-pc/lab04 $ nasm -o obj4.o -f elf -g -l list4.lst lab4.asm
ansuthenko@dk6n55 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  lab4.asm  lab4.o  list4.lst  list.lst  main  obj4.o  obj.o

```

Рис. 4.4: Компиляция и проверка

Передаём объектный файл на обработку и проверяем (рис [afig:015])

```

ansuthenko@dk6n55 ~/work/arch-pc/lab04 $ ld -m elf_i386 lab4.o -o lab4
ansuthenko@dk6n55 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  lab4  lab4.asm  lab4.o  list4.lst  list.lst  main  obj4.o  obj.o

```

Рис. 4.5: Выполнение компоновки и проверка

По тому же принципу передаём файл 'obj4.o' (рис [afig:016])

```

ansuthenko@dk6n55 ~/work/arch-pc/lab04 $ ld -m elf_i386 lab4.o -o main4
ansuthenko@dk6n55 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  lab4  lab4.asm  lab4.o  list4.lst  list.lst  main  main4  obj4.o  obj.o

```

Рис. 4.6: Выполнение компоновки и проверка

С помощью './lab4' запускаем файл.(рис [afig:017])

```

ansuthenko@dk6n55 ~/work/arch-pc/lab04 $ ./lab4 Сущенко Алина
Сущенко Алина
ansuthenko@dk6n55 ~/work/arch-pc/lab04 $

```

Рис. 4.7: Выполнение компоновки и проверка

Копируем файлы в лок.репозиторий в нужный нам каталог.(рис [afig:018])

```
ansuthenko@dk6n55 ~/work/arch-pc/lab04 $ cp ~/work/arch-pc/lab04/hello.asm ~/work/study/2023-2024/архитектура компьютера/arch-pc/labs/lab04/
ansuthenko@dk6n55 ~/work/arch-pc/lab04 $ cp ~/work/arch-pc/lab04/lab4.asm ~/work/study/2023-2024/архитектура компьютера/arch-pc/labs/lab04/
```

Рис. 4.8: Выполнение компановки и проверка

Проверяем (рис [afig:019])

```
ansuthenko@dk6n55 ~/work/arch-pc/lab04 $ cd ~/work/study/2023-2024/архитектура компьютера/arch-pc/labs/lab04/
ansuthenko@dk6n55 ~/work/study/2023-2024/архитектура компьютера/arch-pc/labs/lab04 $ ls
hello.asm lab4.asm presentation report
```

Рис. 4.9: Переход и проверка

Загрузка файлов на github (рис [afig:020])

```
ansuthenko@dk6n55 ~/work/study/2023-2024/архитектура компьютера/arch-pc/labs/lab04 $ git add .
ansuthenko@dk6n55 ~/work/study/2023-2024/архитектура компьютера/arch-pc/labs/lab04 $ git commit -am 'add asm'
[master 412a5f4] add asm
2 files changed, 28 insertions(+)
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/lab4.asm
ansuthenko@dk6n55 ~/work/study/2023-2024/архитектура компьютера/arch-pc/labs/lab04 $ git push
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 665 байтов | 665.00 КиБ/с, готово.
Всего 6 (изменений 3), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To github.com:ALINASUSHCHENKO/study_2023-2024_arh--pc.git
bid45f9..412a5f4 master -> master
```

Рис. 4.10: Загрузка файлов

Проверка (рис [afig:021])

ALINASUSHCHENKO add asm	
Name	Last commit message
..	
presentation	feat(main): make course structure
report	feat(main): make course structure
hello.asm	add asm
lab4.asm	add asm

Рис. 4.11: Загрузка файлов

## 5 Выводы

В ходе выполнения работы мы освоили компиляцию и сборку программ на ассемблере NASM.