

# **Лабораторная работа №4**

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

Налобин Михаил Дмитриевич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Ход работы</b>	<b>6</b>
<b>3</b>	<b>Выводы</b>	<b>11</b>

## Список иллюстраций

2.1	Создание и открытие файла формата .asm . . . . .	6
2.2	Заполнение файла командами . . . . .	7
2.3	Компиляция программы hello . . . . .	7
2.4	Компиляция программы полным вариантом командной строки NASM . . . . .	7
2.5	Передача компоновщику файла hello.o . . . . .	8
2.6	Передача компоновщику файла obj.o . . . . .	8
2.7	Запуск исполняемого файла . . . . .	8
2.8	Создание файла lab4.asm . . . . .	8
2.9	Редактирование файла lab4.asm . . . . .	9
2.10	Вид отредактированного файла . . . . .	9
2.11	Компиляция программы lab4 . . . . .	9
2.12	Передача компоновщику файла hello.o и просмотр результата . .	10
2.13	Загрузка на Github . . . . .	10

## Список таблиц

# 1 Цель работы

Целью данной лабораторной работы является освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

## 2 Ход работы

Перешли в каталог с лабораторной работой №4, создали текстовый файл hello.asm с помощью команды touch и открыли этот текстовый файл с использованием gedit(рис. 2.1).

```
[mdnlobin@mdnlobin ~]$ cd ~/work/study/2023-2024/Архитектура\ компьютера/arch-pc/labs/lab04
[mdnlobin@mdnlobin lab04]$ touch hello.asm
[mdnlobin@mdnlobin lab04]$ gedit hello.asm
```

Рис. 2.1: Создание и открытие файла формата .asm

В текстовом редакторе gedit ввели необходимые команды (рис. 2.2).

```

SECTION .data
    hello:      DB 'Hello World!',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. 2.2: Заполнение файла командами

Превратили текст программы в объективный код с использованием NASM и проверили корректность выполнения (рис. 2.3).

```

[mdnalobin@mdnalobin lab04]$ nasm -f elf hello.asm
[mdnalobin@mdnalobin lab04]$ ls
hello.asm hello.o presentation report

```

Рис. 2.3: Компиляция программы hello

Путем применения расширенного синтаксиса командной строки NASM выполнили команду компиляции файла hello.o в obj.o, после чего убедились в том, что все файлы были созданы (рис. 2.4).

```

[mdnalobin@mdnalobin lab04]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
[mdnalobin@mdnalobin lab04]$ ls
hello.asm hello.o list.lst obj.o presentation report

```

Рис. 2.4: Компиляция программы полным вариантом командной строки NASM

Передали объектный файл hello.o на обработку компоновщику ld и проверили выполнение командой ls (рис. 2.5).

```
[mdnalobin@mdnalobin lab04]$ ld -m elf_i386 hello.o -o hello
[mdnalobin@mdnalobin lab04]$ ls
hello  hello.asm  hello.o  list.lst  obj.o  presentation  report
```

Рис. 2.5: Передача компоновщику файла hello.o

Передали объектный файл obj.o в ходе получим исполняемый файл с именем main, далее снова проверяем выполнение обработки компоновщика (рис. 2.6).

```
[mdnalobin@mdnalobin lab04]$ ld -m elf_i386 obj.o -o main
[mdnalobin@mdnalobin lab04]$ ls
hello  hello.asm  hello.o  list.lst  main  obj.o  presentation  report
```

Рис. 2.6: Передача компоновщику файла obj.o

Запустили на выполнение созданный исполняемый файл, набрав в командой строке ./hello (рис. 2.7).

```
[mdnalobin@mdnalobin lab04]$ ./hello
Hello World!
```

Рис. 2.7: Запуск исполняемого файла

### ##Самостоятельная работа

Создали копию файла hello.asm с именем lab4.asm с помощью команды cp и проверили на наличие (рис. 2.8).

```
[mdnalobin@mdnalobin lab04]$ cp hello.asm lab4.asm
[mdnalobin@mdnalobin lab04]$ ls
hello  hello.o  list.lst  obj.o  report
hello.asm  lab4.asm  main  presentation
```

Рис. 2.8: Создание файла lab4.asm



Пользуясь текстовым редактором gedit, внесли изменения в текст программы, в результате чего должны будем получить при выводе мои имя и фамилию (рис. 2.9 и рис. ??).

```
[mdnalobin@mdnalobin lab04]$ gedit lab4.asm
```

Рис. 2.9: Редактирование файла lab4.asm

```
SECTION .data
    name:          DB 'Налобин Михаил',10
    nameLen:       EQU $-name

SECTION .text
    GLOBAL _start

_start:
    mov eax,4
    mov ebx,1
    mov ecx,name
    mov edx,nameLen
    int 80h

    mov eax,1
    mov ebx,0
    int 80h
```

Рис. 2.10: Вид отредактированного файла

Оттранслировали полученный текст в объектный файл (рис. 2.11).

```
[mdnalobin@mdnalobin lab04]$ nasm -f elf lab4.asm
[mdnalobin@mdnalobin lab04]$ ls
hello      hello.o   lab4.o    main      presentation
hello.asm  lab4.asm  list.lst  obj.o     report
```

Рис. 2.11: Компиляция программы lab4

Выполнили компоновку объектного файла и запустили получившийся исполняемый файл (рис. 2.12).

```
[mdnalobin@mdnalobin lab04]$ ld -m elf_i386 lab4.o -o lab4
[mdnalobin@mdnalobin lab04]$ ./lab4
Налобин Михаил
```

Рис. 2.12: Передача компоновщику файла hello.o и просмотр результата

Загрузили все файлы на github (рис. 2.13).

```
[mdnalobin@mdnalobin lab04]$ git add .
[mdnalobin@mdnalobin lab04]$ git commit -am 'Лабораторная работа№4'
[master a418915] Лабораторная работа№4
9 files changed, 52 insertions(+)
create mode 100755 labs/lab04/hello
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/hello.o
create mode 100755 labs/lab04/lab4
create mode 100644 labs/lab04/lab4.ams
create mode 100644 labs/lab04/lab4.o
create mode 100644 labs/lab04/list.lst
create mode 100755 labs/lab04/main
create mode 100644 labs/lab04/obj.o
[mdnalobin@mdnalobin lab04]$ git push
Перечисление объектов: 16, готово.
Подсчет объектов: 100% (16/16), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (13/13), готово.
Запись объектов: 100% (13/13), 3.39 КиБ | 173.00 КиБ/с, готово.
Всего 13 (изменений 4), повторно использовано 0 (изменений 0), повторно использован
remote: Resolving deltas: 100% (4/4), completed with 2 local objects.
To github.com:xenxock/study_2023-2024_arh--pc.git
 3065ffd..a418915 master -> master
```

Рис. 2.13: Загрузка на Github

## 3 Выводы

В ходе данной лабораторной работы освоили процедуры компиляции и сборки программ, написанные на ассемблере NASM, на примере вывода сообщения ‘Hello World!’ и фамилией с именем.

...