

# **Лабораторная работа номер 4.**

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

Сорокин Кирилл

# Содержание

|          |                                       |           |
|----------|---------------------------------------|-----------|
| <b>1</b> | <b>Цель работы</b>                    | <b>4</b>  |
| <b>2</b> | <b>Задание</b>                        | <b>5</b>  |
| <b>3</b> | <b>Теоретическое введение</b>         | <b>6</b>  |
| <b>4</b> | <b>Выполнение лабораторной работы</b> | <b>7</b>  |
| <b>5</b> | <b>Выводы</b>                         | <b>10</b> |
|          | <b>Список литературы</b>              | <b>11</b> |

## Список иллюстраций

|      |  |   |
|------|--|---|
| 4.1  | Создание папки . . . . .                                 | 7 |
| 4.2  | Файл для работы с кодом . . . . .                        | 7 |
| 4.3  | Наполнение файла с кодом . . . . .                       | 7 |
| 4.4  | Объектный файл . . . . .                                 | 8 |
| 4.5  | Объектный файл с другим именем и файл листинга . . . . . | 8 |
| 4.6  | Создание исполняемых файлов . . . . .                    | 8 |
| 4.7  | Запуск программы . . . . .                               | 8 |
| 4.8  | Самостоятельная работа. Редактирование . . . . .         | 9 |
| 4.9  | Компиляция новой программы . . . . .                     | 9 |
| 4.10 | Выгрузка файлов . . . . .                                | 9 |

# 1 Цель работы

Научиться компилировать и собирать программы, написанные на ассемблере NASM.

## 2 Задание

Изучить теоретический материал, и на его основе написать простейшие программы на языке ассемблер NASM.

### 3 Теоретическое введение

Язык ассемблера (assembly language, сокращённо asm) — машинно-ориентированный язык низкого уровня. NASM — это открытый проект ассемблера, версии которого доступны под различные операционные системы и который позволяет получать объектные файлы для этих систем. В NASM используется Intel-синтаксис и поддерживаются инструкции x86-64.

## 4 Выполнение лабораторной работы

Создадим папку для выполнения работы и перейдём в неё (рис. 4.1).

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

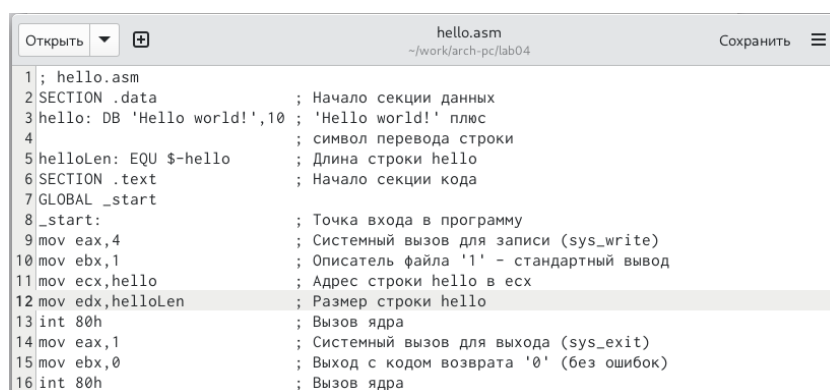
Рис. 4.1: Создание папки

Командой `touch` создадим файл `hello.asm`, где будем писать текст программы (рис. 4.2).

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

Рис. 4.2: Файл для работы с кодом

Заполним содержимое файла представленным нам текстом программы (рис. 4.3).



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

Рис. 4.3: Наполнение файла с кодом

Командой `nasm` создадим объектный файл `hello.o` (рис. 4.4).

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

Рис. 4.4: Объектный файл

С помощью параметра `-o` создадим объектный файл с заданным именем (`obj`), при этом формат выходного файла будет `elf`, и в него будут включены символы для отладки благодаря параметру `-g` и будет создан файл листинга (`list.lst`) благодаря параметру `-l`. (рис. 4.5).

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

Рис. 4.5: Объектный файл с другим именем и файл листинга

Командой `ld` создадим из объектных файлов исполняемые файлы (рис. 4.6).

```
kvsorokin@dk6n55 ~/work/arch-pc/lab04 $ ld -m elf_i386 hello.o -o hello
kvsorokin@dk6n55 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
kvsorokin@dk6n55 ~/work/arch-pc/lab04 $
```

Рис. 4.6: Создание исполняемых файлов

Запустим исполняемый файл (рис. 4.7).

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

Рис. 4.7: Запуск программы

В качестве самостоятельной работы скопируем файл `hello.asm` с именем `lab4.asm`, а затем отредактируем его как указано в задании (рис. 4.8).



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

Открыть ▾

+

lab4.asm  
~/work/arch-pc/lab04

```
1 ; hello.asm
2 SECTION .data                ; Начало секции данных
3 hello: DB 'Sorokin Kirill',10 ; 'Sorokin Kirill' плюс
```

Рис. 4.8: Самостоятельная работа. Редактирование

Создадим объектный файл и исполняемый файл из lab4.asm, после чего запустим его (рис. 4.9).

```
kvsorokin@dk6n55 ~/work/arch-pc/lab04 $ nasm -f elf lab4.asm
kvsorokin@dk6n55 ~/work/arch-pc/lab04 $ ld -m elf_i386 lab4.o -o lab4
kvsorokin@dk6n55 ~/work/arch-pc/lab04 $ ls
hello  hello.o  lab4  lab4.asm  lab4.o  list.lst  main  obj.o
kvsorokin@dk6n55 ~/work/arch-pc/lab04 $ ./lab4
Sorokin Kirill
```

Рис. 4.9: Компиляция новой программы

Скопируем файлы lab4.asm и hello.asm в рабочую папку лабораторной работы, после чего выгрузим их на Github (рис. 4.10).

```
kvsorokin@dk6n55 ~/work/arch-pc/lab04 $ cp lab4.asm '/afs/.dk.sci.pfu.edu.ru/home/k/v/kvsorokin/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04/lab4.asm'
kvsorokin@dk6n55 ~/work/arch-pc/lab04 $ cp hello.asm '/afs/.dk.sci.pfu.edu.ru/home/k/v/kvsorokin/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04/hello.asm'
kvsorokin@dk6n55 ~/work/arch-pc/lab04 $ ls '/afs/.dk.sci.pfu.edu.ru/home/k/v/kvsorokin/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04'
hello.asm  lab4.asm  presentation  report
kvsorokin@dk6n55 ~/work/arch-pc/lab04 $ cd '/afs/.dk.sci.pfu.edu.ru/home/k/v/kvsorokin/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04'
kvsorokin@dk6n55 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04 $ git add .
kvsorokin@dk6n55 ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04 $ git commit -am 'lab4'
[master 6e3354b] lab4
2 files changed, 32 insertions(+)
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/lab4.asm
kvsorokin@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), 959 байтов | 959.00 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:sorokinkirill1132236060/-study_2023-2024_arh--pc.git
a0c8dcb..6e3354b master -> master
```

Рис. 4.10: Выгрузка файлов

## 5 Выводы

Мы освоили процедуру компиляции и сборки программ, написанных на ассемблере NASM.

## Список литературы

1. GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>.
2. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>.
3. Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>.
4. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: <http://www.amazon.com/Learning-bash-Shell-Programming-Nutshell/dp/0596009658>.
6. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс, 2017.
11. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
12. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.
13. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВ Петербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
14. Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-е изд. — М. : МАКС Пресс, 2011. — URL: [http://www.stolyarov.info/books/asm\\_unix](http://www.stolyarov.info/books/asm_unix).

15. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
16. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер,
17. — 1120 с. — (Классика Computer Science).