

# Лабораторная работа №4

## Создание и процесс обработки программ на языке ассемблера NASM

Мочалкина Софья Васильевна

### Содержание

Содержание .....	1
Цель работы .....	1
Задание .....	1
Теоретическое введение .....	2
Выполнение лабораторной работы .....	2
Выводы.....	4
Список литературы.....	5

### Содержание

1. Цель работы 1
2. Задание 1
3. Теоретическое введение 2
4. Выполнение лабораторной работы 2
5. Выводы 4
6. Список литературы 5

### Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

### Задание

1. Программа Hello world! 1.1 создать каталог для работы с программами на языке ассемблера NASM 1.2 перейти в созданный каталог 1.3 создать текстовый файл с именем hello.asm 1.4 открыть этот файл 1.5 ввести в него указанный текст
2. Транслятор Nasm 2.1 выполнить компиляцию в объектный код 3.Расширенный синтаксис 3.1 выполнить компиляцию исходного файла
3. Компоновщик LD 4.1 передать объектный файл на обработку компоновщику
4. Запустить исполняемый файл

5. Задания для самостоятельной работы 6.1 создать копию файла `hello.asm` с именем `lab4.asm` 6.2 изменить скопированный файл, чтобы выводилась строка с именем и фамилией
6. Задания для самостоятельной работы 6.3 оттранслировать полученный текст программы `lab4.asm` в объектный файл 6.4 скопировать файлы `hello.asm` и `lab4.asm` в локальный репозиторий

## Теоретическое введение

В процессе создания ассемблерной программы можно выделить четыре шага: • Набор текста программы в текстовом редакторе и сохранение её в отдельном файле. Каждый файл имеет свой тип (или расширение), который определяет назначение файла. Файлы с исходным текстом программ на языке ассемблера имеют тип `asm`.

- Трансляция — преобразование с помощью транслятора, например `nasm`, текста программы в машинный код, называемый объектным. На данном этапе также может быть получен листинг программы, содержащий кроме текста программы различную дополнительную информацию, созданную транслятором. Тип объектного файла — `o`, файла листинга — `lst`.

- Компоновка или линковка — этап обработки объектного кода компоновщиком (`ld`), который принимает на вход объектные файлы и собирает по ним исполняемый файл. Исполняемый файл обычно не имеет расширения. Кроме того, можно получить файл карты загрузки программы в ОЗУ, имеющий расширение `map`.

- Запуск программы. Конечной целью является работоспособный исполняемый файл.

Ошибки на предыдущих этапах могут привести к некорректной работе программы, поэтому может присутствовать этап отладки программы при помощи специальной программы — отладчика. При нахождении ошибки необходимо провести коррекцию программы, начиная с первого шага.

## Выполнение лабораторной работы

1. Создаю каталог для работы с программами на языке ассемблера `NASM`(см.рис.1)
2. Перехожу в созданный каталог(см.рис.1)
3. Создаю текстовый файл с именем `hello.asm`.(см.рис.1)
4. Открываю этот файл с помощью текстового редактора.(см.рис.1)

```
svmochalkina@dk3n38 ~ $ mkdir -p ~/work/arch-pc/lab04
svmochalkina@dk3n38 ~ $ cd ~/work/arch-pc/lab04
svmochalkina@dk3n38 ~/work/arch-pc/lab04 $ touch hello.asm
svmochalkina@dk3n38 ~/work/arch-pc/lab04 $ gedit hello.asm
```

Создание файла *hello.asm*(рис.1)

5. Ввожу в файл *hello.asm* текст из файла лабораторной работы(см.рис.2)

```
1 ; hello.asm
2 SECTION .data ; Начало секции данных
3 hello: DB 'Hello world!',10 ; 'Hello world!' 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 ехх
12 mov edx,helloLen ; Размер строки hello
13 int 80h ; Вызов ядра
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
```

Вводим текст в открывшийся файл (рис.2)

6. Компилирую текст программы “Hello World”(см.рис.3)

```
svmochalkina@dk3n38 ~/work/arch-pc/lab04 $ nasm -f elf hello.asm
svmochalkina@dk3n38 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o
svmochalkina@dk3n38 ~/work/arch-pc/lab04 $
```

Компиляция *Hello world*(рис.3)

7. Выполняю полный вариант командной строки *nasm*(см.рис.4)

```
svmochalkina@dk3n38 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
svmochalkina@dk3n38 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o  list.lst  obj.o
svmochalkina@dk3n38 ~/work/arch-pc/lab04 $
```

Полный вариант строки *nasm*(рис.4)

8. Передаю объектный файл на обработку компоновщику. С помощью команды *ls* проверяю, что исполняемый файл *hello* был создан(см.рис.5)

```
svmochalkina@dk3n38 ~/work/arch-pc/lab04 $ ld -m elf_i386 hello.o -o hello
svmochalkina@dk3n38 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  list.lst  obj.o
svmochalkina@dk3n38 ~/work/arch-pc/lab04 $
```

Передача компоновщику(рис.5)

9. Выполняю команду `d -m elf_i386 obj.o -o main` и смотрю на результат.(см.рис.6)

```
svmochalkina@dk3n38 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
svmochalkina@dk3n38 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
svmochalkina@dk3n38 ~/work/arch-pc/lab04 $
```

*Вводим команду(рис.6)*

10. Запустить на выполнение созданный исполняемый файл, находящийся в текущем каталоге, можно, набрав в командной строке. (см.рис.7)

```
svmochalkina@dk3n38 ~/work/arch-pc/lab04 $ ./hello
Hello world!
svmochalkina@dk3n38 ~/work/arch-pc/lab04 $
```

*Запуск Hello world(рис.7)*

11. Задание для самостоятельной работы. (см.рис.8)

1. В каталоге `~/work/arch-pc/lab04` с помощью команды `cp` создаю копию файла `hello.asm` с именем `lab4.asm`
2. С помощью текстового редактора вношу изменения в текст программы в файле `lab4.asm` так, чтобы вместо `Hello world!` на экран выводилась строка с фамилией и именем.
3. Оттранспирую полученный текст программы `lab4.asm` в объектный файл. Выполняю компоновку объектного файла и запускаю получившийся исполняемый файл.
4. Копирую файлы `hello.asm` и `lab4.asm` в локальный репозиторий в каталоге `~/work/study/2024-2025/“Архитектура компьютера”/arch-pc/labs/lab04/`. Загрузите файлы на Github.

```
svmochalkina@dk3n38 ~/work/arch-pc/lab04 $ cd ~/work/arch-pc/lab04
svmochalkina@dk3n38 ~/work/arch-pc/lab04 $ cp hello.asm lab4.asm
svmochalkina@dk3n38 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  lab4.asm  list.lst  main  obj.o
svmochalkina@dk3n38 ~/work/arch-pc/lab04 $ nasm -f elf lab4.asm
svmochalkina@dk3n38 ~/work/arch-pc/lab04 $ ld -m elf_i386 lab4.o -o name
svmochalkina@dk3n38 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main_name
svmochalkina@dk3n38 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  lab4.asm  lab4.o  list.lst  main  main_name  name  obj.o
svmochalkina@dk3n38 ~/work/arch-pc/lab04 $ ./name
Mochalkina Sofia
```

*Выполнение самостоятельной работы(рис.8)*

## Выводы

Я ознакомилась с созданием и процессом обработки программ на языке ассемблера 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,
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).