

# Отчёт по лабораторной работе 4

язык ассемблера NASM

Чуева З.С.

## Содержание

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

1. Цель работы
2. Задание
3. Теоретическое введение
4. Выполнение лабораторной работы
5. Вывод

## Цель работы

Освоить процедуру компиляции и сборки программ, написанных на ассемблере NASM.

## Задание

1. Написать текст программы Hello world!
2. Выполнить компиляцию в объектный код
3. Выполнить компиляцию исходного файла
4. Выполнить компоновку объектного файла
5. Запуститьисполняемый файл
6. Выполнить задания для самостоятельной работы

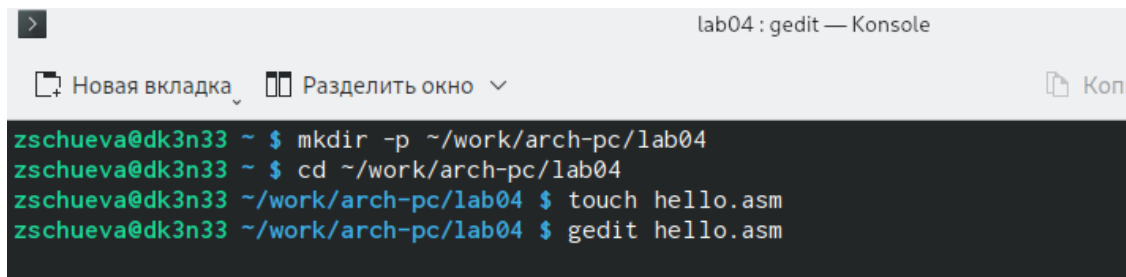
# Теоретическое введение

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.

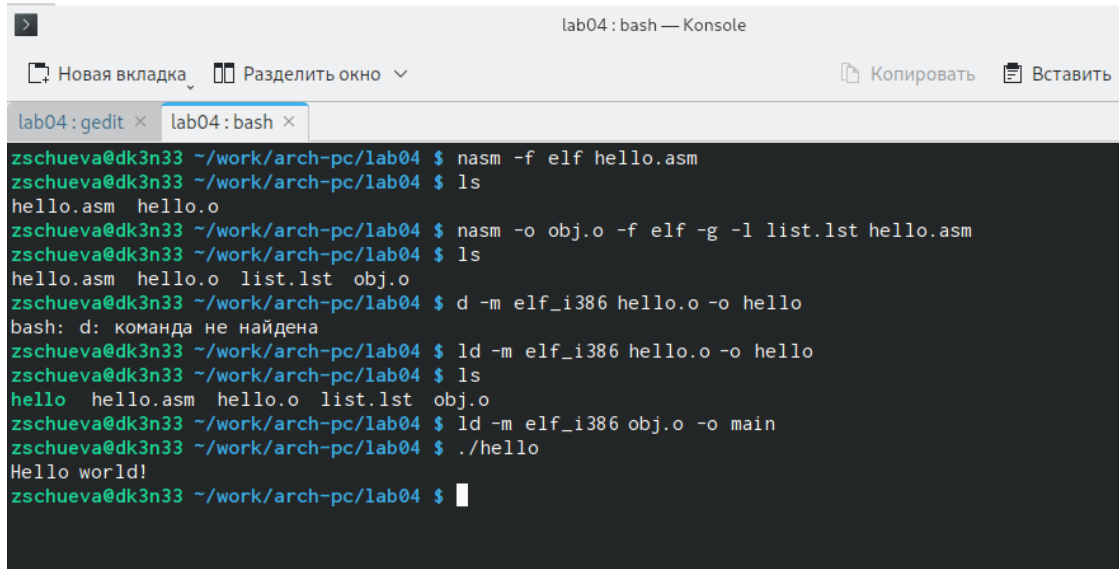
## Выполнение лабораторной работы

1. Программа Hello world!
  - 1.1 Создать каталог для работы с программой на языке NASM.
  - 1.2 Перйти в сохданный каталог.
  - 1.3 Создать текстовый файл с именем `hello.asm`.
  - 1.4 Открыть этот файл с помощью текстового редактора.
  - 1.5 Ввести в него код программы.  
смотреть Рис.1 и Рис.2
2. Транслятор NASM 2.1 Выполним компиляцию в объектный код.  
смотреть Рис.2
3. Расширенный синтаксис командной строки NASM 3.1 Выполним компиляцию исходного файла.  
смотреть Рис.2
4. Кмпоновщик LD 4.1 Передадим объектный файл на обработку компоновщику.  
смотреть Рис.2
5. Запустим исполняемый файл  
смотреть Рис.2



```
lab04 : gedit — Konsole
Новая вкладка  Разделить окно  Коп
zschueva@dk3n33 ~ $ mkdir -p ~/work/arch-pc/lab04
zschueva@dk3n33 ~ $ cd ~/work/arch-pc/lab04
zschueva@dk3n33 ~/work/arch-pc/lab04 $ touch hello.asm
zschueva@dk3n33 ~/work/arch-pc/lab04 $ gedit hello.asm
```

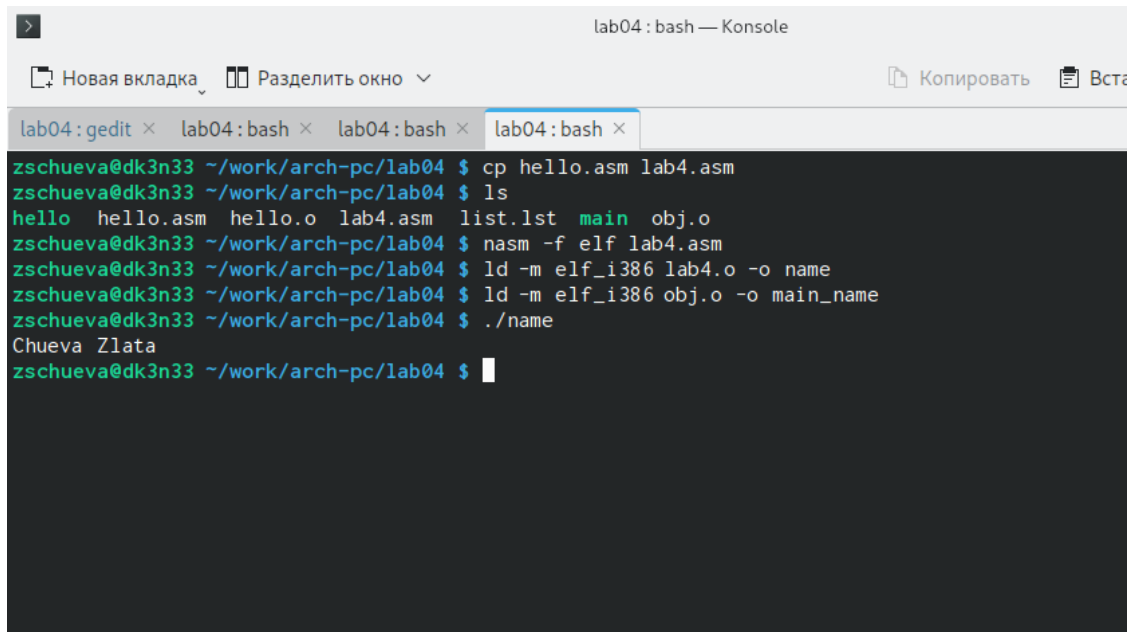
Рис.1



```
lab04 : bash — Konsole
Новая вкладка  Разделить окно  Копировать  Вставить
lab04 : gedit x  lab04 : bash x
zschueva@dk3n33 ~/work/arch-pc/lab04 $ nasm -f elf hello.asm
zschueva@dk3n33 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o
zschueva@dk3n33 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
zschueva@dk3n33 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o  list.lst  obj.o
zschueva@dk3n33 ~/work/arch-pc/lab04 $ d -m elf_i386 hello.o -o hello
bash: d: команда не найдена
zschueva@dk3n33 ~/work/arch-pc/lab04 $ ld -m elf_i386 hello.o -o hello
zschueva@dk3n33 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  list.lst  obj.o
zschueva@dk3n33 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
zschueva@dk3n33 ~/work/arch-pc/lab04 $ ./hello
Hello world!
zschueva@dk3n33 ~/work/arch-pc/lab04 $
```

Рис.2

6. Задания для самостоятельной работы 6.1 В каталоге ~/work/arch-pc/lab04 с помощью команды `cp` создадим копию файла `hello.asm` с именем `lab4.asm`.
  - 6.2 С помощью текстового редактора изменим текст программы так, чтобы вместо `hello world` вывелись имя и фамилия.
  - 6.3 Оттранслируем полученный текст программы `lab4.asm` в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл.
  - 6.4 Скопируем файлы `hello.asm` и `lab4.asm` в Ваш локальный репозиторий в каталог ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/. Загрузим файлы на Github.
- смотреть Рис.3



```
lab04: bash — Konsole
Новая вкладка  Разделить окно  Копировать  Вставить
lab04:gedit × lab04: bash × lab04: bash × lab04: bash ×
zschueva@dk3n33 ~/work/arch-pc/lab04 $ cp hello.asm lab4.asm
zschueva@dk3n33 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  lab4.asm  list.lst  main  obj.o
zschueva@dk3n33 ~/work/arch-pc/lab04 $ nasm -f elf lab4.asm
zschueva@dk3n33 ~/work/arch-pc/lab04 $ ld -m elf_i386 lab4.o -o name
zschueva@dk3n33 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main_name
zschueva@dk3n33 ~/work/arch-pc/lab04 $ ./name
Chueva Zlata
zschueva@dk3n33 ~/work/arch-pc/lab04 $
```

Рис.3

## Выводы

Освоила 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,  
::: {#refs} :::