

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра физико-математических и естественных наук**

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4**

дисциплина: Архитектура компьютера

Студент: Попова Алиса Владимировна

Группа: НПИ-03-25

**МОСКВА**

2025г.

## Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM

# Теоретическая часть

Процесс создания ассемблерной программы:



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.

## Задание 1

### Программа Hello world!

- Создайте каталог для работы с программами на языке ассемблера NASM:
- Перейдите в созданный каталог
- Создайте текстовый файл с именем hello.asm
- Откройте этот файл с помощью любого текстового редактора, например, gedit
- Введите текст файла

```
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41: ~/wor...
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~$ mkdir -p ~/work/arch-pc/lab04
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~$ cd ~/work/arch-pc/lab04
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab04$ touch hello.asm
```

```
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41: ~/wor...
hello.o
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab04$ cat > hello.asm << 'EOF'
; hello.asm
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
EOF
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/arch-pc/lab04$ cat hello.asm
```

Создан каталог для работы на языке ассамблера ASM, совершён переход в созданный каталог, создан текстовый файл с именем hello.asm. Но в ходе выполнения задания возникли проблемы с открытием файла при помощи текстового редактора, было принято решение вводить задание через терминал.

## Задание 2

### Транслятор NASM

- для компиляции приведённого выше текста программы «Hello World» необходимо написать:

```
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/study/2025-2026/
Архитектура компьютера/arch-pc/labs/lab04$ nasm -f elf hello.asm
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/study/2025-2026/
Архитектура компьютера/arch-pc/labs/lab04$ ls
hello.asm  hello.o  lab4.asm
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/study/2025-2026/
Архитектура компьютера/arch-pc/labs/lab04$
```

запущена команда **nasm -f elf hello.asm**, превращающая программу в объектный код. После этого проведена проверка на наличие в каталоге нового объектного файла hello.o.

## Задание 3 и Задание 4

### Компоновщик LD + Запуск исполняемого файла

- Чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику.
- Запустить на выполнение созданный исполняемый файл

```
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/study/2025-2026/
Архитектура компьютера/arch-pc/labs/lab04$ ld -m elf_i386 hello.o -o hello
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/study/2025-2026/
Архитектура компьютера/arch-pc/labs/lab04$ ./hello
Hello world!
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/study/2025-2026/
```

Файл был передан на обработку компоновщику и запущен.

## Задание для самостоятельной работы

1. В каталоге `~/work/arch-pc/lab04` с помощью команды `cp` создайте копию файла `hello.asm` с именем `lab4.asm`
2. С помощью любого текстового редактора внесите изменения в текст программы в файле `lab4.asm` так, чтобы вместо `Hello world!` на экран выводилась строка с вашими фамилией и именем.
3. Оттранслируйте полученный текст программы `lab4.asm` в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл.
4. Скопируйте файлы `hello.asm` и `lab4.asm` в Ваш локальный репозиторий в каталог `~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab04/`. Загрузите файлы на Github.

```
Архитектура компьютера/arch-pc/labs/lab04$ nasm -f elf lab4.asm
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/study/2025-2026/
Архитектура компьютера/arch-pc/labs/lab04$ ls
hello  hello.asm  hello.o  lab4.asm  lab4.o
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/study/2025-2026/
Архитектура компьютера/arch-pc/labs/lab04$ ld -m elf_i386 lab4.o -o lab4
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/study/2025-2026/
Архитектура компьютера/arch-pc/labs/lab04$ ./lab4
Попова Алиса
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/study/2025-2026/
Архитектура компьютера/arch-pc/labs/lab04$
```

```
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41: ~/work/study...
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/study/2025-2026/
Архитектура компьютера/arch-pc/labs/lab04$ git add .
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/study/2025-2026/
Архитектура компьютера/arch-pc/labs/lab04$ git commit -m "lab4: add report files
"
[master 4c990e3] lab4: add report files
6 files changed, 36 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.asm
create mode 100644 labs/lab04/lab4.o
alisa-vladimirovna-p@alisa-vladimirovna-p-Nitro-ANV15-41:~/work/study/2025-2026/
Архитектура компьютера/arch-pc/labs/lab04$ git push
Username for 'https://github.com': sirokchek
Password for 'https://sirokchek@github.com':
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 12 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (10/10), 1.60 KiB | 821.00 KiB/s, done.
Total 10 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 2 local objects.
To https://github.com/sirokchek/study_2025-2026_arh-pc.git
631c93e..4c990e3 master -> master
```

## Выводы

В ходе работы были освоены основные этапы создания и обработки программ на языке ассемблера NASM: написание кода, компиляция, компоновка и запуск. Были изучены базовые команды NASM и LD, а также структура простой ассемблерной программы.



# Список литературы

## Архитектура ЭВМ

### 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, 2015. — 1120 с. — (Классика Computer Science).