

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

дисциплина: Архитектура компьютера

Студент: Мартынова Милана Александровна

Группа: НКАбд-04-25

МОСКВА

2025 г.

Содержание

1. Цель работы.....	4
2. Порядок выполнения лабораторной работы.....	4
2.1 Программа Hello, world!	4
2.2 Транслятор NASM.....	4
2.3 Расширенный синтаксис командной строки NASM(GNU Assembler).....	5
2.4 Компоновщик LD.....	5
2.5 Запуск исполняемого файла.....	5
2.6 Задание для самостоятельной работы.....	5
3. Вывод.....	7

Список иллюстраций

1.1 Создание рабочей директории.....	4
1.2 Переход в созданный каталог.....	4
1.3 Создание .asm файла.....	4
1.4 Открытие файла с помощью текстового редактора.....	4
1.5 Написание программы.....	4
2.1 Компиляция программы.....	5
3.1 Возможности синтаксиса GAS.....	5
4.1 Отправка файла компоновщику.....	5
4.2 Создание исполняемого файла.....	5
5.1 Запуск программы.....	5
6.1 Создание копии.....	6
6.2 Редактирование копии.....	6
6.3 Трансляция и компоновка копии файла.....	6
6.4 Запуск копии файла.....	6
6.5 Отправка файлов в локальный репозиторий.....	6
6.6 Загрузка изменений.....	6

1. Цель работы

Изучение процесса трансляции и линковки программ, разработанных на ассемблере NASM.

2. Порядок выполнения лабораторной работы

2.1 Программа Hello, world!

В домашней директории создаю каталог, в котором буду хранить файлы для текущей лабораторной работы:

```
mamartynova@ubuntu:~$ mkdir -p ~/work/arch-pc/lab04
```

1.1 Создание рабочей директории

Далее перехожу в созданный каталог:

```
mamartynova@ubuntu:~$ cd ~/work/arch-pc/lab04  
mamartynova@ubuntu:~/work/arch-pc/lab04$
```

1.2 Переход в созданный каталог

Создаю в нём текстовый файл с именем `hello.asm`, в котором буду писать программу на языке ассемблера:

```
mamartynova@ubuntu:~/work/arch-pc/lab04$ touch hello.asm
```

1.3 Создание .asm файла

Далее открываю этот файл с помощью текстового редактора и пишу в нём программу:

```
mamartynova@ubuntu:~/work/arch-pc/lab04$ gedit hello.asm
```

1.4 Открытие файла с помощью текстового редактора



1.5 Написание программы

Изменяю код программы под архитектуру компьютера ARM64.

2.2 Транслятор NASM

Для архитектуры ARM64 использование NASM имеет ограничения, поскольку NASM изначально разработан для x86/x64 архитектур. Поэтому используем GNU Assembler (gas).

Компилирую с помощью GNU Assembler:

```
mamartynova@ubuntu:~/work/arch-pc/lab04$ as -o hello.o hello.asm
mamartynova@ubuntu:~/work/arch-pc/lab04$ ls
hello.asm  hello.o
```

2.1 Компиляция программы

2.3 Расширенный синтаксис командной строки NASM(GNU Assembler)

Выполняю команду, указанную на рис 3.1, она скомпилировала исходный файл hello.asm в obj.o, расширение .o говорит о том, что файл – объектный, помимо него флаги -g -al подготовят файл отладки и листинга соответственно.

```
mamartynova@ubuntu:~/work/arch-pc/lab04$ as -o obj.o -g -al=list.lst hello.asm
mamartynova@ubuntu:~/work/arch-pc/lab04$ ls
hello.asm  hello.o  list.lst  obj.o
```

3.1 Возможности синтаксиса GAS

2.4 Компоновщик LD

Затем мне необходимо передать объектный файл компоновщику, делаю это с помощью команды ld:

```
mamartynova@ubuntu:~/work/arch-pc/lab04$ ld -m aarch64elf -o hello hello.o
mamartynova@ubuntu:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  obj.o
```

4.1 Отправка файла компоновщику

Выполняю следующую команду, результатом исполнения которой будет созданный файл main, скомпонованный из объектного файла obj.o:

```
mamartynova@ubuntu:~/work/arch-pc/lab04$ ld -m aarch64linux -o main obj.o
mamartynova@ubuntu:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
```

4.2 Создание исполняемого файла

2.5 Запуск исполняемого файла

Запускаю исполняемый файл из текущего каталога:

```
mamartynova@ubuntu:~$ ./hello
Hello, world!
```

5.1 Запуск программы

2.6 Задание для самостоятельной работы

Создаю копию файла для последующей работы с ним:

```
mamartynova@ubuntu:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
mamartynova@ubuntu:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4.asm list.lst main obj.o
```

6.1 Создание копии

Редактирую копию файла, заменив 'Hello, world' на свое имя и фамилию:



```
*lab4.asm
/home/mamartynova/work/arch-pc/lab04
Сохранить x

1.global _start
2.section .data
3.msg:
4    .ascii "Martynova Milana\n"
5    len = . - msg
6
7.section .text
8_start:
9    mov x0, #1
10   ldr x1, =msg
11   ldr x2, =len
12   mov x8, #64
13   svc #0
14
15   mov x0, #0
16   mov x8, #93
17   svc #0
```

6.2 Редактирование копии

Транслирую копию файла в объектный файл, компоную и запускаю:

```
mamartynova@ubuntu:~/work/arch-pc/lab04$ as -o lab4.o lab4.asm
mamartynova@ubuntu:~/work/arch-pc/lab04$ ld -m aarch64elf -o lab4 lab4.o
mamartynova@ubuntu:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4 lab4.asm lab4.o list.lst main obj.o
```

6.3 Трансляция и компоновка копии файла

```
mamartynova@ubuntu:~/work/arch-pc/lab04$ ./lab4
Martynova Milana
```

6.4 Запуск копии файла

Убедившись в корректности работы программы, копирую рабочие файлы в свой локальный репозиторий:

```
mamartynova@ubuntu:~/work/arch-pc/lab04$ cp hello.asm lab4.asm ~/work/study/2025-2026/"Архитектура компьютера"/arch-pc/labs/lab04
mamartynova@ubuntu:~/work/arch-pc/lab04$ cd ~/work/study/2025-2026/"Архитектура компьютера"/arch-pc/labs/lab04/
mamartynova@ubuntu:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab04$ ls
hello.asm lab4.asm presentation report
```

6.5 Отправка файлов в локальный репозиторий

Далее загружаю изменения на свой удаленный репозиторий на GitHub:

```
mamartynova@ubuntu:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab04$ git add
mamartynova@ubuntu:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab04$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   hello.asm
        new file:   lab4.asm

mamartynova@ubuntu:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab04$ git commit -m "feat(main): upload 4 lab work"
[master 2b5fcd] feat(main): upload 4 lab work
2 files changed, 34 insertions(+)
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/lab4.asm
mamartynova@ubuntu:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab04$ git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 7 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 697 bytes | 697.00 KiB/s, done.
Total 6 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To github.com:Lanawx/study_2025-2026_arch-pc.git
7d3d55f..2b5fcd master -> master
```

6.6 Загрузка изменений

3. Вывод

В ходе выполнения лабораторной работы была успешно освоена процедура компиляции и сборки программ, написанных на языке ассемблера NASM(GNU Assembler). Были получены практические навыки работы с инструментарием низкоуровневого программирования и изучен полный цикл создания исполняемых файлов из исходного кода на ассемблере.

В процессе работы были выполнены следующие этапы:

1. Создан исходный код программы "Hello World" на языке ассемблера с использованием синтаксиса NASM(GNU Assembler)
2. Освоена процедура трансляции исходного кода в объектный файл с помощью компилятора NASM(GNU Assembler)
3. Изучен процесс компоновки объектного файла в исполняемый модуль с использованием компоновщика LD
4. Проведен успешный запуск полученной программы и проверена ее работоспособность
5. Выполнено задание для самостоятельной работы по модификации программы с выводом персональных данных

В результате работы были закреплены теоретические знания об архитектуре ЭВМ, системе команд процессора и принципах взаимодействия программного обеспечения с аппаратными ресурсами компьютера. Особое внимание было уделено изучению регистровой структуры процессора и механизму системных вызовов для организации ввода-вывода.

Работа подтвердила, что программирование на языке ассемблера обеспечивает максимальную близость к аппаратным возможностям компьютера и позволяет глубоко понимать принципы работы вычислительных систем на низком уровне. Полученные навыки компиляции и сборки программ являются фундаментальными для дальнейшего изучения архитектуры компьютеров и системного программирования.

Список литературы

1. Пример выполнения лабораторной работы
2. Курс на ТУИС
3. Лабораторная работа №4