

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №4

дисциплина: Архитектура компьютера

Студент: Володина Алиса Алексеевна

Группа: НКАбд-01-25

МОСКВА

2025 г.

Содержание	
Цель работы	5
Задания	6
Теоретическое введение	7
Выполнение лабораторной работы	9
Вывод	13
Список литературы	14

Список иллюстраций

Рисунок 0.6.....	9
Рисунок 0.7.....	9
Рисунок 0.8.....	9
Рисунок 0.9.....	9
Рисунок 0.10.....	9
Рисунок 0.11.....	10
Рисунок 0.12.....	10
Рисунок 0.13.....	10
Рисунок 0.14.....	10
Рисунок 0.15.....	10
Рисунок 0.16.....	11
Рисунок 0.17.....	11
Рисунок 0.18.....	11
Рисунок 0.19.....	11
Рисунок 0.20.....	11
Рисунок 0.21.....	12
Рисунок 0.22.....	12
Рисунок 0.23.....	12
Рисунок 0.24.....	12
Рисунок 0.25.....	12
Рисунок 0.26.....	13

Список таблиц

Рисунок 0.1	7
Рисунок 0.2	7
Рисунок 0.3	7
Рисунок 0.4	8
Рисунок 0.5	8

1)Цель работы

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

2)Задания

Написание программы для вывода “Hello world!”, работа с NASM, работа с компоновщиком LD, запуск исполняемого файла, отправка файлов на Github.

3)Теоретическое введение



Рисунок 0.1



Рисунок 0.2

Параметр	Действие
+число	переход к указанной числом строке
-b	черно-белая цветовая гамма
-c	цветовой режим ANSI для терминалов без поддержки цвета
-d	отключить поддержку мыши
-v	вывести версию программы

Рисунок 0.3

Клавиши	Действие
Ctrl + Ins	копировать
Shift + Ins	вставить
Shift + Del	вырезать
Ctrl + Del	удалить выделенный текст
F3	начать выделение текста (повторное нажатие F3 закончит выделение)
Shift + F3	начать выделение блока текста (повторное нажатие F3 закончит выделение)
F5	скопировать выделенный текст
F6	переместить выделенный текст

Рисунок 0.4

Клавиши	Действие
F8	удалить выделенный текст
Meta + l	переход к строке по её номеру
Meta + q	вставка литерала (непечатного символа)
Meta + t	сортировка строк выделенного текста
Meta + u	выполнить внешнюю команду и вставить в позицию под курсором её вывод
Ctrl + f	занести выделенный фрагмент во внутренний буфер обмена mc (записать во внешний файл)
Ctrl + k	удалить часть строки до конца строки
Ctrl + n	создать новый файл
Ctrl + s	включить или выключить подсветку синтаксиса
Ctrl + t	выбрать кодировку текста
Ctrl + u	отменить действия
Ctrl + x	перейти в конец следующего слова
Ctrl + y	удалить строку
Ctrl + z	перейти на начало предыдущего слова
Shift + F5	вставка текста из внутреннего буфера обмена mc (прочитать внешний файл)
Meta + Enter	диалог перехода к определению функции
Meta + -	возврат после перехода к определению функции
Meta + +	переход вперёд к определению функции
Meta + n	включение/отключение отображения номеров строк
Tab	отодвигает вправо выделенный текст, если выключена опция «Постоянные блоки»
Meta + Tab	отодвигает влево выделенный текст, если выключена опция «Постоянные блоки»
Shift + Стрелки	выделение текста
Meta + Стрелки	выделение вертикального блока
Meta + Shift + -	переключение режима отображения табуляций и пробелов
Meta + Shift + +	переключение режима «Автовывравнивание возвратом каретки»

Рисунок 0.5

4)Выполнение лабораторной работы

Создадим каталог для работы с программами на языке ассемблера NASM (рисунок 0.6):

```
aavolodina1@dk5n06 ~ $ mkdir -p ~/work/arch-pc/lab04
```

Рисунок 0.6

Перейдем в созданный каталог (рисунок 0.7):

```
aavolodina1@dk5n06 ~ $ cd ~/work/arch-pc/lab04
```

Рисунок 0.7

Создадим текстовый файл с именем hello.asm (рисунок 0.8):

```
aavolodina1@dk5n06 ~/work/arch-pc/lab04 $ touch hello.asm
```

Рисунок 0.8

Откроем этот файл с помощью gedit (рисунок 0.9):

```
aavolodina1@dk5n06 ~/work/arch-pc/lab04 $ gedit hello.asm
```

Рисунок 0.9

Введем в него следующий текст (рисунок 0.10):



Рисунок 0.10

NASM превращает текст программы в объектный код, напишем (рисунок 0.11):

```
aavolodina1@dk5n06 ~/work/arch-pc/lab04 $ nasm -f elf hello.asm
aavolodina1@dk5n06 ~/work/arch-pc/lab04 $
```

Рисунок 0.11

С помощью команды `ls` проверим, что объектный файл был создан (рисунок 0.12):

```
aavolodina1@dk5n07 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o
aavolodina1@dk5n07 ~/work/arch-pc/lab04 $
```

Рисунок 0.12

Выполните следующую команду: `nasm -o obj.o -f elf -g -l list.lst hello.asm` (рисунок 0.13):

```
aavolodina1@dk5n06 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
aavolodina1@dk5n06 ~/work/arch-pc/lab04 $
```

Рисунок 0.13

С помощью команды `ls` проверим, что файлы были созданы (рисунок 0.14):

```
aavolodina1@dk5n07 ~/work/arch-pc/lab04 $ ls
hello.asm  hello.o  list.lst  obj.o
aavolodina1@dk5n07 ~/work/arch-pc/lab04 $
```

Рисунок 0.14

Для более подробной информации используем `man nasm`. Для получения списка форматов объектного файла используем `nasm-hf` (рисунок 0.15-0.16):

```
aavolodina1@dk5n07 ~/work/arch-pc/lab04 $ man nasm
aavolodina1@dk5n07 ~/work/arch-pc/lab04 $
```

Рисунок 0.15

```

aavolodina1@dk5n07 ~/work/arch-pc/lab04 $ nasm -hf
Usage: nasm [-@ response_file] [options...] [--] filename
       nasm -v (or --v)

Options (values in brackets indicate defaults):

  -h          show this text and exit (also --help)
  -v (or --v) print the NASM version number and exit
  -@ file     response file; one command line option per line

  -o outfile  write output to outfile
  --keep-all  output files will not be removed even if an error happens

  -Xformat    specify error reporting format (gnu or vc)
  -s          redirect error messages to stdout
  -Zfile      redirect error messages to file

  -M          generate Makefile dependencies on stdout
  -MG         d.o, missing files assumed generated
  -MF file    set Makefile dependency file
  -MD file    assemble and generate dependencies
  -MT file    dependency target name
  -MQ file    dependency target name (quoted)
  -MP         emit phony targets

  -f format   select output file format
    bin       Flat raw binary (MS-DOS, embedded, ...) [default]
    ith       Intel Hex encoded flat binary
    srec      Motorola S-records encoded flat binary
    aout      Linux a.out
    aoutb     NetBSD/FreeBSD a.out
    coff      COFF (i386) (DJGPP, some Unix variants)
    elf32     ELF32 (i386) (Linux, most Unix variants)
    elf64     ELF64 (x86-64) (Linux, most Unix variants)
    elfx32    ELFX32 (ELF32 for x86-64) (Linux)
    as86      as86 (bin86/dev86 toolchain)
    obj       Intel/Microsoft OMF (MS-DOS, OS/2, Win16)
    win32     Microsoft extended COFF for Win32 (i386)
    win64     Microsoft extended COFF for Win64 (x86-64)

```

Рисунок 0.16

Выполним следующую команду: `ld -m elf_i386 obj.o -o main` (рисунок 0.17):

```

aavolodina1@dk5n06 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
aavolodina1@dk5n06 ~/work/arch-pc/lab04 $

```

Рисунок 0.17

При помощи команды `ls` проверим, какое имя будет иметь файл (рисунок 0.18):

```

aavolodina1@dk5n07 ~/work/arch-pc/lab04 $ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
aavolodina1@dk5n07 ~/work/arch-pc/lab04 $

```

Рисунок 0.18

Запустим созданный файл (рисунок 0.19):

```

aavolodina1@dk5n06 ~/work/arch-pc/lab04 $ ./hello
Hello world!

```

Рисунок 0.19

В каталоге `~/work/arch-pc/lab04` с помощью команды `cp` создадим копию файла `hello.asm` с именем `lab4.asm` (рисунок 0.20):

```

aavolodina1@dk5n06 ~/work/arch-pc/lab04 $ cp hello.asm lab4.asm
aavolodina1@dk5n06 ~/work/arch-pc/lab04 $

```

Рисунок 0.20

С помощью текстового редактора внесем изменения в текст программы в файле `lab4.asm` так, чтобы вместо “Hello world!” на экран выводилась строка с фамилией и именем (рисунок 0.21-0.22):

```
aavolodina1@dk5n06 ~/work/arch-pc/lab04 $ gedit lab4.asm
```

Рисунок 0.21



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

1 SECTION .data
2     hello:      db "Alisa Volodina",0xa
3               helloLen:  equ $ - hello
4 SECTION .text
5     global _start
6
7 _start:
8     mov eax, 4
9     mov ebx, 1
10    mov ecx, hello
11    mov edx, helloLen
12    int 0x80
13
14    mov eax, 1
15    mov ebx, 0
16    int 0x80
```

Рисунок 0.22

Оттранслируем полученный текст программы lab4.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл (рисунок 0.23-0.25):

```
ра лаборато aavolodina1@dk5n06 ~/work/arch-pc/lab04 $ nasm -f elf lab4.asm
aavolodina1@dk5n06 ~/work/arch-pc/lab04 $
```

Рисунок 0.23

```
ра лаборато aavolodina1@dk5n06 ~/work/arch-pc/lab04 $ ld -m elf_i386 lab4.o -o lab4
aavolodina1@dk5n06 ~/work/arch-pc/lab04 $
```

Рисунок 0.24

```
aavolodina1@dk5n06 ~/work/arch-pc/lab04 $ ./lab4
Alisa Volodina
```

Рисунок 0.25

Скопируем файлы hello.asm и lab4.asm в локальный репозиторий в каталог ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab04/. Загрузим файлы на Github (рисунок 0.26):

```
aavolodina1@dk5n06 ~/work/study/2025-2026/Архитектура компьютеров/arch-pc $ git add .
aavolodina1@dk5n06 ~/work/study/2025-2026/Архитектура компьютеров/arch-pc $ git commit -am 'feat(main): add hello.asm & lab4.asm'
[master 7a40207] feat(main): add hello.asm & lab4.asm
2 files changed, 32 insertions(+)
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/lab4.asm
aavolodina1@dk5n06 ~/work/study/2025-2026/Архитектура компьютеров/arch-pc $ 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), 676 байтов | 676.00 КиБ/с, готово.
Total 6 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To github.com:AlisaVolodina/study_2025-2026_arh-pc.git
 a5df035..7a40207 master -> master
aavolodina1@dk5n06 ~/work/study/2025-2026/Архитектура компьютеров/arch-pc $
```

Рисунок 0.26

5)Вывод

Мы освоили процедуры 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 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.
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)