

Лабораторная работа №4

Глушенок Анна Александровна

Содержание

1	Цель работы	5
2	Ход выполнения работы	6
2.1	Задание 1. Программа Hello world!	6
2.2	Задание 2. Транслятор NASM	7
2.3	Задание 3. Расширенный синтаксис командной строки NASM . . .	8
2.4	Задание 4. Компоновщик LD	9
2.5	Задание 5. Запуск исполняемого файла	10
2.6	Задания для самостоятельной работы	11
3	Вывод	14

Список иллюстраций

2.1	Создание рабочего каталога	6
2.2	Работа с hello.asm в тестовом редакторе	7
2.3	Создание объектного файла	7
2.4	Компиляция файла hello.asm в obj.o	8
2.5	Просмотр дополнительной информации	9
2.6	Создание исполняемого файла	10
2.7	Просмотр дополнительной информации	10
2.8	Запуск исполняемого файла	11
2.9	Создание копии файла	11
2.10	Внесение изменений в файл	12
2.11	Запуск файла	12
2.12	Проверка	12
2.13	Копирование файлов	12
2.14	Проверка результата	13

Список таблиц

1 Цель работы

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

2 Ход выполнения работы

2.1 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.

С помощью команды `mkdir` создаем необходимый каталог, перемещаемся в него командой `cd`, и создаем текстовый файл с именем `hello.asm`, используя команду `touch`.

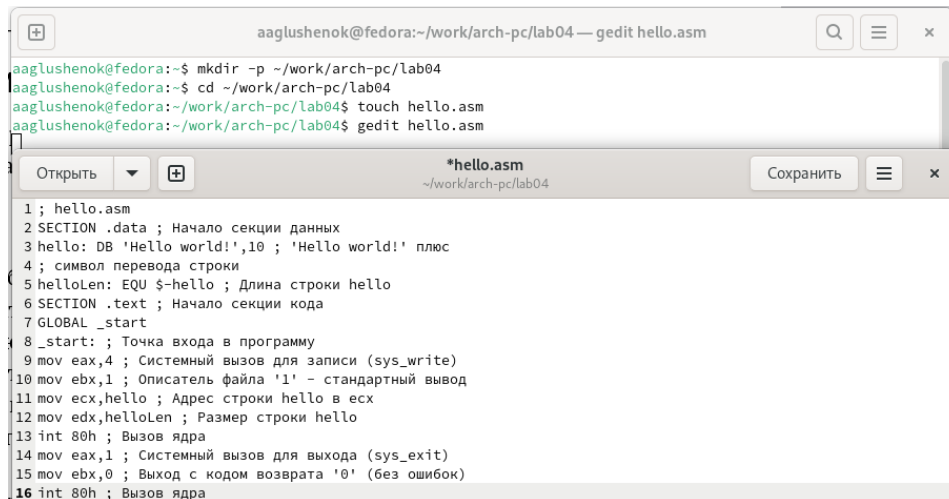
A screenshot of a terminal window. The title bar shows the user 'aaglushenok' on a 'fedora' machine, in the directory '~/work/arch-pc/lab04', editing 'hello.asm' with 'gedit'. The terminal content shows three commands being executed: 'mkdir -p ~/work/arch-pc/lab04', 'cd ~/work/arch-pc/lab04', and 'touch hello.asm'. The prompt changes from '~\$' to '~/work/arch-pc/lab04\$' after each command.

```
aaglushenok@fedora:~/work/arch-pc/lab04 — gedit hello.asm
aaglushenok@fedora:~$ mkdir -p ~/work/arch-pc/lab04
aaglushenok@fedora:~$ cd ~/work/arch-pc/lab04
aaglushenok@fedora:~/work/arch-pc/lab04$ touch hello.asm
```

Рис. 2.1: Создание рабочего каталога

Откройте этот файл с помощью любого текстового редактора, и введите в него необходимый текст.

Открываем файл в редакторе `gedit`, вводя “`gedit hello.asm`”, затем вводим в него необходимый текст.



```
aaglushmanok@fedora:~/work/arch-pc/lab04 — gedit hello.asm
aaglushmanok@fedora:~$ mkdir -p ~/work/arch-pc/lab04
aaglushmanok@fedora:~$ cd ~/work/arch-pc/lab04
aaglushmanok@fedora:~/work/arch-pc/lab04$ touch hello.asm
aaglushmanok@fedora:~/work/arch-pc/lab04$ gedit hello.asm

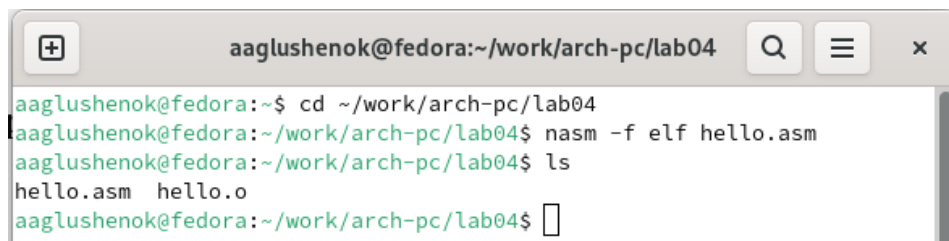
1 ; hello.asm
2 SECTION .data ; Начало секции данных
3 hello: DB 'Hello world!',10 ; 'Hello world!' плюс
4 ; символ перевода строки
5 helloLen: EQU $-hello ; Длина строки hello
6 SECTION .text ; Начало секции кода
7 GLOBAL _start
8 _start: ; Точка входа в программу
9 mov eax,4 ; Системный вызов для записи (sys_write)
10 mov ebx,1 ; Описатель файла '1' - стандартный вывод
11 mov ecx,hello ; Адрес строки hello в ecx
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
```

Рис. 2.2: Работа с hello.asm в тестовом редакторе

2.2 Задание 2. Транслятор NASM

Преобразуйте текст программы из файла hello.asm в объектный код, который запишется в файл hello.o. С помощью команды ls проверьте, что объектный файл был создан

Вводим в терминал “nasm -f elf hello.asm”, затем командой ls проверяем, что необходимый файл создан.



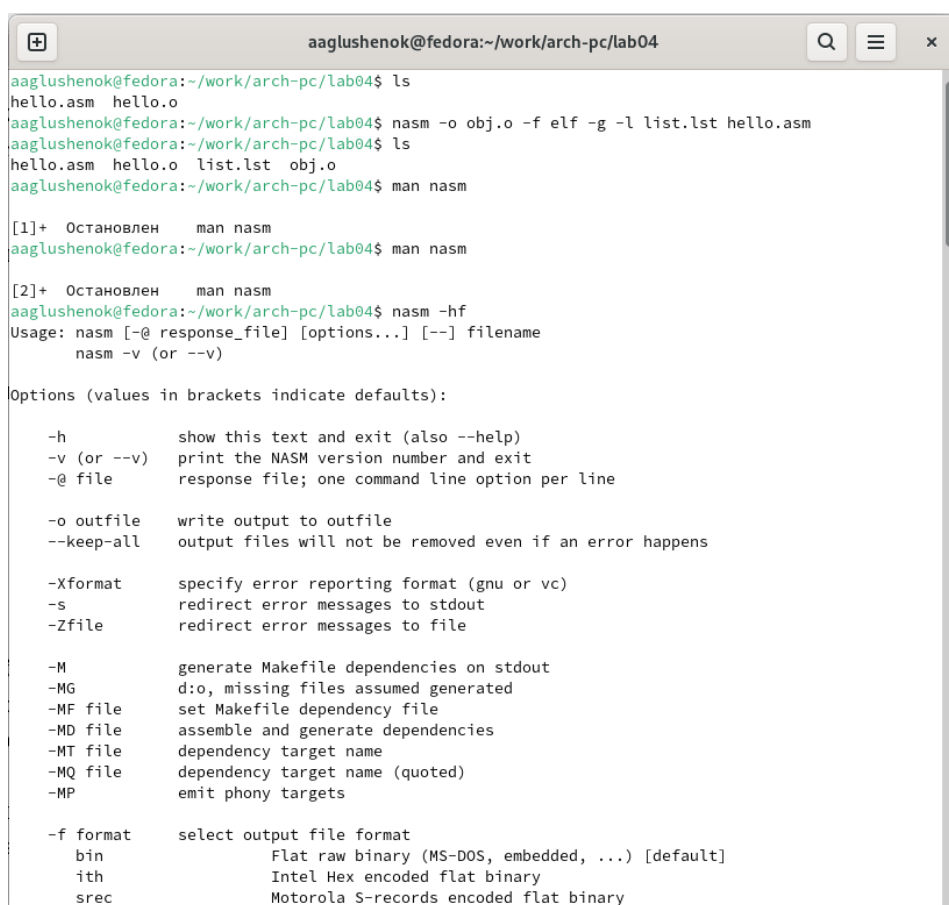
```
aaglushmanok@fedora:~/work/arch-pc/lab04
aaglushmanok@fedora:~$ cd ~/work/arch-pc/lab04
aaglushmanok@fedora:~/work/arch-pc/lab04$ nasm -f elf hello.asm
aaglushmanok@fedora:~/work/arch-pc/lab04$ ls
hello.asm hello.o
aaglushmanok@fedora:~/work/arch-pc/lab04$
```

Рис. 2.3: Создание объектного файла

2.3 Задание 3. Расширенный синтаксис командной строки NASM

Скомпилируйте исходный файл `hello.asm` в `obj.o`. С помощью команды `ls` проверьте, что файлы были созданы. Для более подробной информации см. “`man nasm`”. Для получения списка форматов объектного файла см. “`nasm -hf`”.

Для компиляции файла `hello.asm` в `obj.o` выполняем команду “`nasm -o obj.o -f elf -g -l list.lst hello.asm`”. Вводим указанные команды, получаем более подробную информацию и список форматов объектного файла.



```
aaglushmanok@fedora:~/work/arch-pc/lab04$ ls
hello.asm hello.o
aaglushmanok@fedora:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst hello.asm
aaglushmanok@fedora:~/work/arch-pc/lab04$ ls
hello.asm hello.o list.lst obj.o
aaglushmanok@fedora:~/work/arch-pc/lab04$ man nasm

[1]+  Остановлен      man nasm
aaglushmanok@fedora:~/work/arch-pc/lab04$ man nasm

[2]+  Остановлен      man nasm
aaglushmanok@fedora:~/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

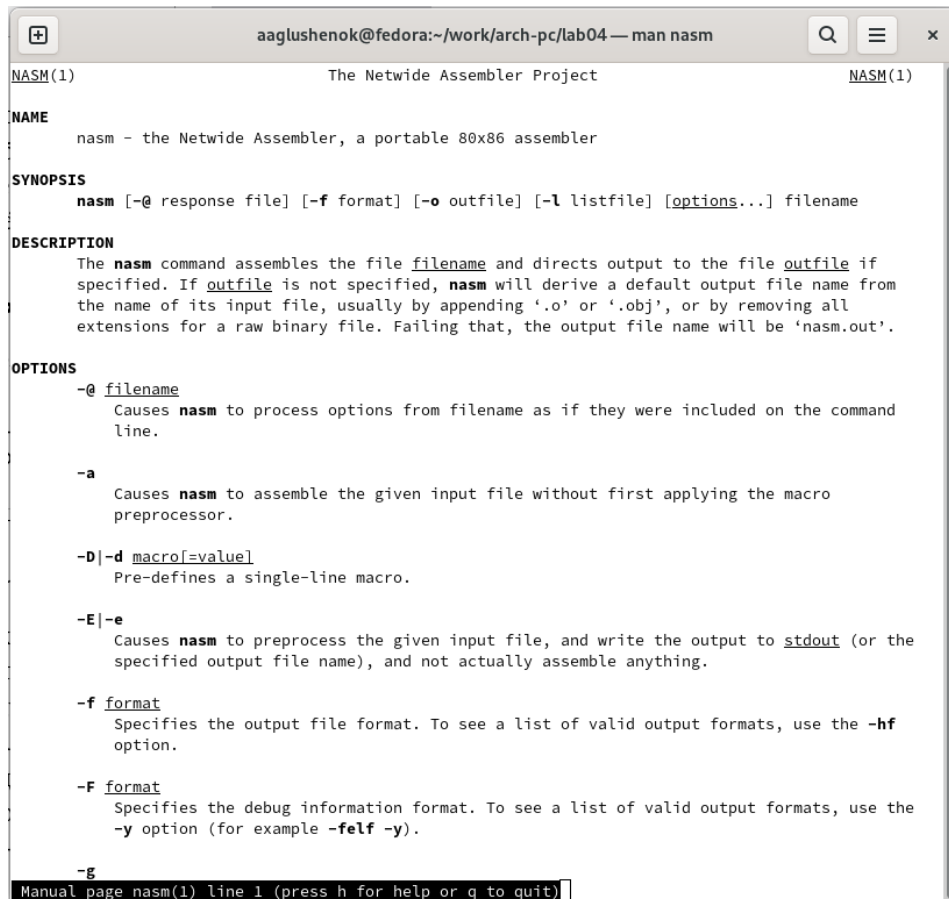
  -o outfile        write output to outfile
  --keep-all       output files will not be removed even if an error happens

  -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
```

Рис. 2.4: Компиляция файла `hello.asm` в `obj.o`



```
NASM(1)                                The Netwide Assembler Project                                NASM(1)

NAME
    nasm - the Netwide Assembler, a portable 80x86 assembler

SYNOPSIS
    nasm [-@ response file] [-f format] [-o outfile] [-l listfile] [options...] filename

DESCRIPTION
    The nasm command assembles the file filename and directs output to the file outfile if
    specified. If outfile is not specified, nasm will derive a default output file name from
    the name of its input file, usually by appending '.o' or '.obj', or by removing all
    extensions for a raw binary file. Failing that, the output file name will be 'nasm.out'.

OPTIONS
    -@ filename
        Causes nasm to process options from filename as if they were included on the command
        line.

    -a
        Causes nasm to assemble the given input file without first applying the macro
        preprocessor.

    -D|-d macro[=value]
        Pre-defines a single-line macro.

    -E|-e
        Causes nasm to preprocess the given input file, and write the output to stdout (or the
        specified output file name), and not actually assemble anything.

    -f format
        Specifies the output file format. To see a list of valid output formats, use the -hf
        option.

    -F format
        Specifies the debug information format. To see a list of valid output formats, use the
        -y option (for example -felF -y).

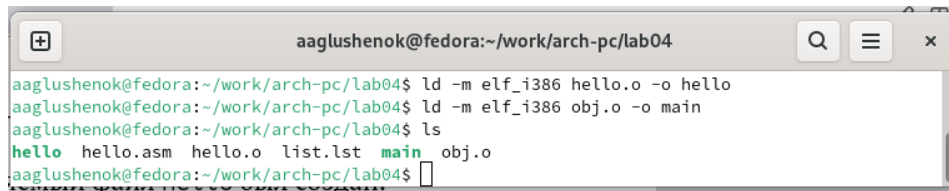
    -g
        Manual page nasm(1) line 1 (press h for help or q to quit)
```

Рис. 2.5: Просмотр дополнительной информации

2.4 Задание 4. Компоновщик LD

Получите исполняемую программу, передав объектный файл на обработку компоновщику. С помощью команды `ls` проверьте, что исполняемый файл `hello` был создан. Выполните следующую команду: `ld -m elf_i386 obj.o -o main`.

Вводим команды “`ld -m elf_i386 obj.o -o main`” и “`ld -m elf_i386 obj.o -o main`”. Проверяем командой `ls`.

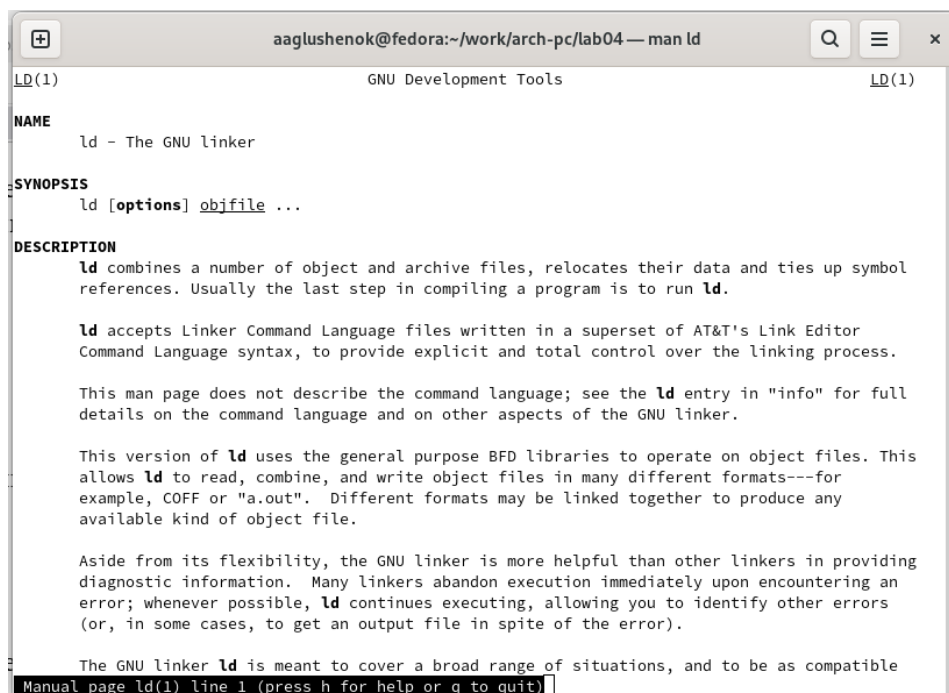


```
aaglushenok@fedora:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
aaglushenok@fedora:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
aaglushenok@fedora:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
aaglushenok@fedora:~/work/arch-pc/lab04$
```

Рис. 2.6: Создание исполняемого файла

Формат командной строки LD можно увидеть, набрав `ld --help`. Для получения более подробной информации см. `man ld`.

Вводим указанные команды для просмотра формата командной строки `ld` и получения более подробной информации.



```
aaglushenok@fedora:~/work/arch-pc/lab04 — man ld
LD(1)                                GNU Development Tools                                LD(1)

NAME
  ld - The GNU linker

SYNOPSIS
  ld [options] objfile ...

DESCRIPTION
  ld combines a number of object and archive files, relocates their data and ties up symbol references. Usually the last step in compiling a program is to run ld.

  ld accepts Linker Command Language files written in a superset of AT&T's Link Editor Command Language syntax, to provide explicit and total control over the linking process.

  This man page does not describe the command language; see the ld entry in "info" for full details on the command language and on other aspects of the GNU linker.

  This version of ld uses the general purpose BFD libraries to operate on object files. This allows ld to read, combine, and write object files in many different formats---for example, COFF or "a.out". Different formats may be linked together to produce any available kind of object file.

  Aside from its flexibility, the GNU linker is more helpful than other linkers in providing diagnostic information. Many linkers abandon execution immediately upon encountering an error; whenever possible, ld continues executing, allowing you to identify other errors (or, in some cases, to get an output file in spite of the error).

  The GNU linker ld is meant to cover a broad range of situations, and to be as compatible

Manual page ld(1) line 1 (press h for help or q to quit)
```

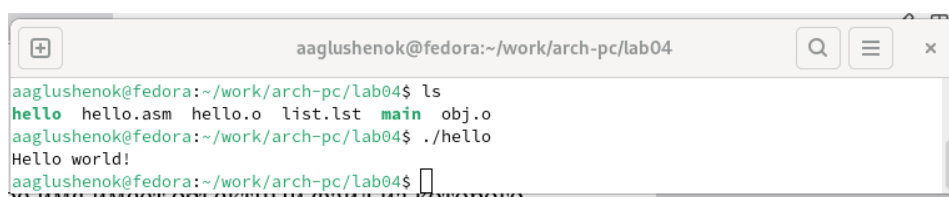
Рис. 2.7: Просмотр дополнительной информации

2.5 Задание 5. Запуск исполняемого файла

Запустите на выполнение созданный исполняемый файл, находящийся в текущем каталоге.

Для запуска исполняемого файла на выполнение переходим в нужный каталог

и вводим “./hello”/

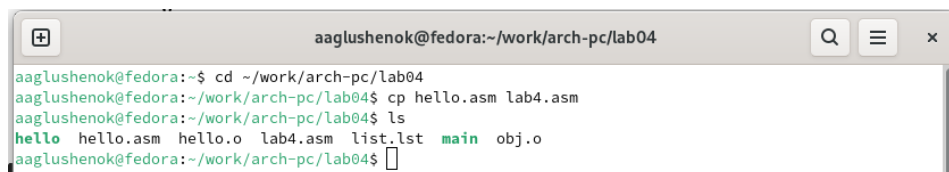


```
aaglushmanok@fedora:~/work/arch-pc/lab04
aaglushmanok@fedora:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o list.lst main obj.o
aaglushmanok@fedora:~/work/arch-pc/lab04$ ./hello
Hello world!
aaglushmanok@fedora:~/work/arch-pc/lab04$
```

Рис. 2.8: Запуск исполняемого файла

2.6 Задания для самостоятельной работы

В каталоге ~/work/arch-pc/lab04 с помощью команды `cp` создайте копию файла `hello.asm` с именем `lab4.asm`.



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

Рис. 2.9: Создание копии файла

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. Оттранслируйте полученный текст программы `lab4.asm` в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл.

```
aaglushmanok@fedora:~/work/arch-pc/lab04$ gedit lab4.asm
lab4.asm
~/work/arch-pc/lab04
1 ; hello.asm
2 SECTION .data
3 hello: DB 'Глушенок Анна',10
4 helloLen: EQU $-hello
5 SECTION .text
6 GLOBAL _start
7 _start:
8 mov eax,4
9 mov ebx,1
10 mov ecx,hello
11 mov edx,helloLen
12 int 80h
13 mov eax,1
14 mov ebx,0
15 int 80h
```

Рис. 2.10: Внесение изменений в файл

```
aaglushmanok@fedora:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
aaglushmanok@fedora:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst lab4.asm
aaglushmanok@fedora:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
aaglushmanok@fedora:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
aaglushmanok@fedora:~/work/arch-pc/lab04$ ./lab4
Глушенок Анна
aaglushmanok@fedora:~/work/arch-pc/lab04$
```

Рис. 2.11: Запуск файла

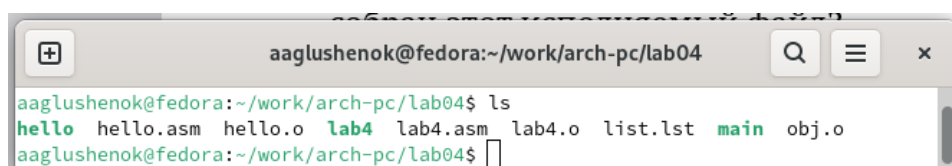
```
aaglushmanok@fedora:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4 lab4.asm lab4.o list.lst main obj.o
aaglushmanok@fedora:~/work/arch-pc/lab04$
```

Рис. 2.12: Проверка

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.

```
aaglushmanok@fedora:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ cp ~/work/arch-pc/lab04/*.asm ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04/
aaglushmanok@fedora:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ cd ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04
bash: cd: /home/aaglushmanok/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04: Нет такого файла или каталога
aaglushmanok@fedora:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ cd ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04
aaglushmanok@fedora:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ ls
hello.asm lab4.asm presentation report
aaglushmanok@fedora:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$
```

Рис. 2.13: Копирование файлов



The image shows a terminal window with a title bar that reads "aaglushenok@fedora:~/work/arch-pc/lab04". The terminal contains the following text:

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

The output of the `ls` command is displayed in a single line, with each file name separated by a space. The file names are color-coded: `hello` is green, `lab4` is red, and `main` is green. The prompt `aaglushenok@fedora:~/work/arch-pc/lab04$` is shown at the end of the line.

Рис. 2.14: Проверка результата

3 Вывод

В ходе выполнения лабораторной работы мне удалось освоить процедуры компиляции и сборки программ, написанных на ассемблере NASM.