
0.1 Front matter

title: "Отчёт по лабораторной работе №4" subtitle: "дисциплина: «Архитектура компьютеров и операционные системы»" author: "Швед Карина Дмитриевна"

0.2 Generic options

lang: ru-RU toc-title: "Содержание"

0.3 Bibliography

bibliography: bib/cite.bib csl: pandoc/csl/gost-r-7-0-5-2008-numeric.csl

0.4 Pdf output format

toc: true # Table of contents toc-depth: 2 lof: true # List of figures lot: true # List of tables
fontsize: 12pt linestretch: 1.5 papersize: a4 documentclass: scrreprt ## I18n polyglossia
polyglossia-lang: name: russian options: - spelling=modern - babelshorthands=true
polyglossia-otherlangs: name: english ## I18n babel babel-lang: russian babel-otherlangs:
english ## Fonts mainfont: IBM Plex Serif romanfont: IBM Plex Serif sansfont: IBM Plex
Sans monofont: IBM Plex Mono mathfont: STIX Two Math mainfontoptions:
Ligatures=Common,Ligatures=TeX,Scale=0.94 romanfontoptions:
Ligatures=Common,Ligatures=TeX,Scale=0.94 sansfontoptions:
Ligatures=Common,Ligatures=TeX,Scale=MatchLowercase,Scale=0.94 monofontoptions:
Scale=MatchLowercase,Scale=0.94,FakeStretch=0.9 mathfontoptions: ## Biblatex biblatex:
true biblio-style: "gost-numeric" biblatexoptions: - parenttracker=true - backend=biber -
hyperref=auto - language=auto - autolang=other* - citestyle=gost-numeric ## Pandoc-
crossref LaTeX customization figureTitle: "Рис." tableTitle: "Таблица" listingTitle:
"Листинг" lofTitle: "Список иллюстраций" lotTitle: "Список таблиц" lolTitle:
"Листинги" ## Misc options indent: true header-includes: -

keep figures where there are in the text

— # keep figures where there are in the text

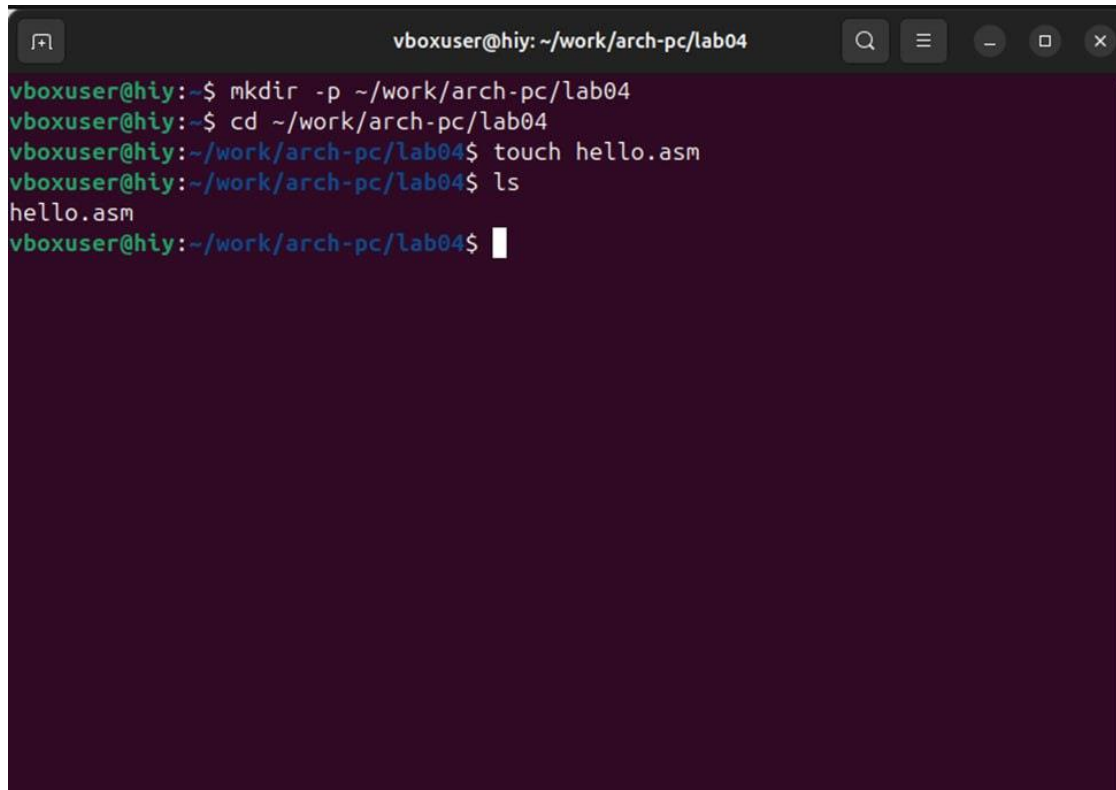
1 Цель работы

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

2 Ход работы

Я создала каталог для работы с программами на языке ассемблера NASM lab 4 с помощью команды `mkdir`, затем перешла в него с помощью команды `cd` и создала файл `hello.asm`

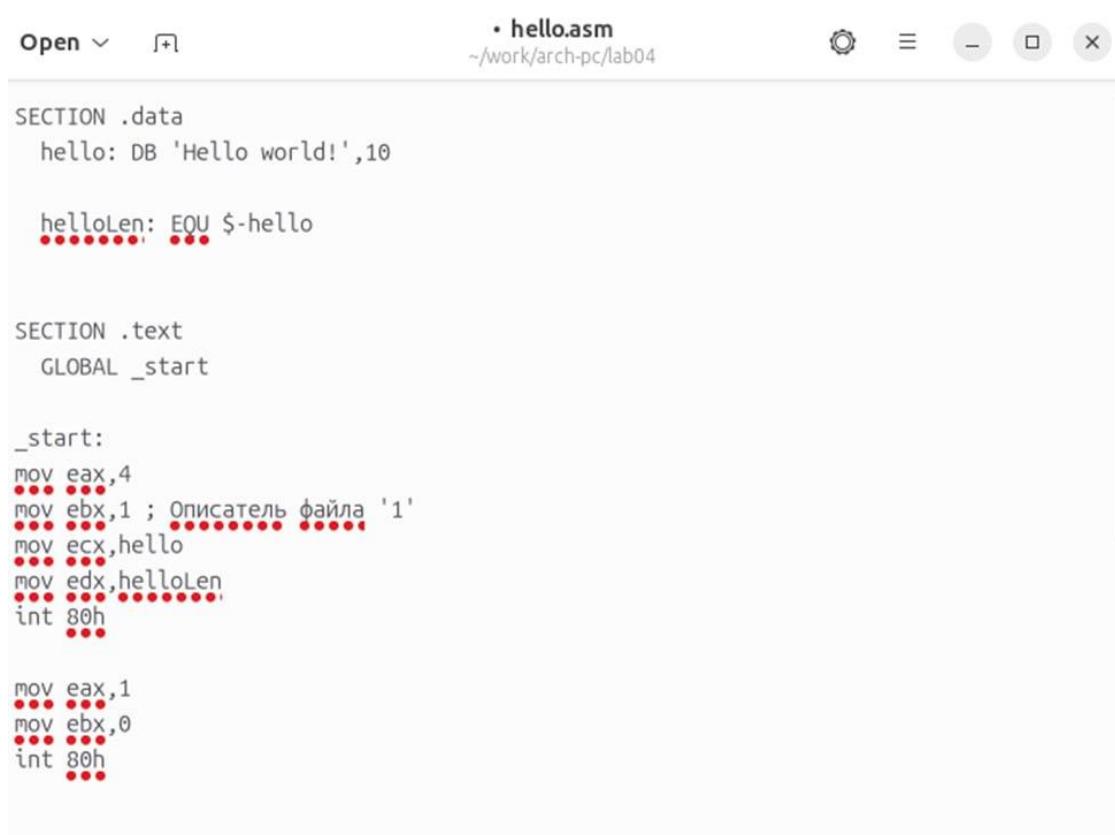
(рис. `fig:001?`).

A terminal window with a dark background and light green text. The window title is 'vboxuser@hiy: ~/work/arch-pc/lab04'. The terminal shows the following commands and output:

```
vboxuser@hiy:~$ mkdir -p ~/work/arch-pc/lab04
vboxuser@hiy:~$ cd ~/work/arch-pc/lab04
vboxuser@hiy:~/work/arch-pc/lab04$ touch hello.asm
vboxuser@hiy:~/work/arch-pc/lab04$ ls
hello.asm
vboxuser@hiy:~/work/arch-pc/lab04$
```

Далее я открыла этот файл с помощью текстового редактора и ввела следующий текст

(рис. `fig:002?`).



```
SECTION .data
    hello: DB 'Hello world!',10

    helloLen: EQU $-hello

SECTION .text
    GLOBAL _start

_start:
    mov eax,4
    mov ebx,1 ; Описатель файла '1'
    mov ecx,hello
    mov edx,helloLen
    int 80h

    mov eax,1
    mov ebx,0
    int 80h
```

Для компиляции текста программы «Hello World» я написала: `nasm -f elf hello.asm`. Транслятор преобразовал текст программы из файла `hello.asm` в объектный код, который записался в файл `hello.o`

С помощью команды `ls` я проверила, что объектный файл был создан (рис. **fig:003?**).

компиляция текста программы и проверка создания файла `hello.o`

Далее я выполнила следующую команду: `nasm -o obj.o -f elf -g -l list.lst hello.asm`

Эта команда скомпилировала исходный файл `hello.asm` в `obj.o`. При этом формат выходного файла стал `elf`, и в него стали включены символы для отладки (опция `-g`). Также был создан файл листинга `list.lst`.

С помощью команды `ls` я проверила, что файлы были создан (рис. **fig:004?**).

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

Чтобы получить исполняемую программу, я передала объектный файл на обработку компоновщику: `ld -m elf_i386 hello.o -o hello` Проверила создание файла `hello` с помощью `ls`

(рис. [fig:005?](#)).

```
Processing triggers for man-db (2.12.0-4build2) ...
vboxuser@hiy:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
vboxuser@hiy:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o list.lst obj.o
vboxuser@hiy:~/work/arch-pc/lab04$
```

Далее я выполнила еще одну линковку для объектного файла `obj.o` и получила исполняемый файл с именем `main`.

(рис. [fig:006?](#)).

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

Далее я набрала `./hello`, Запустила исполняемые файлы и проверила их работу

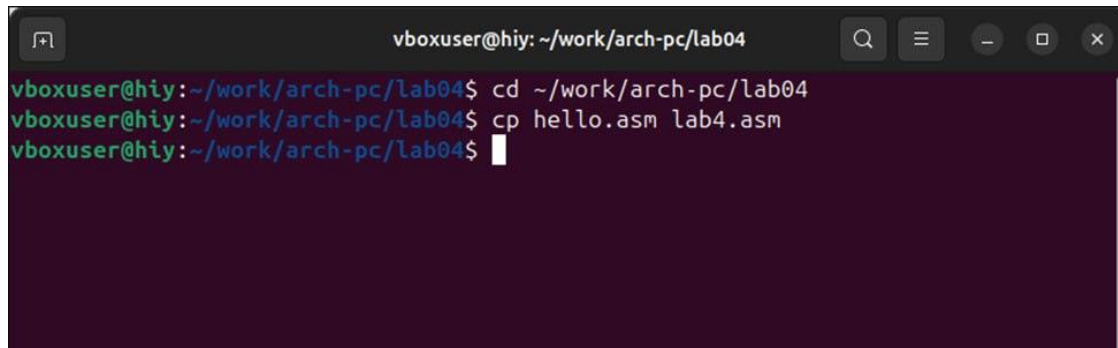
(рис. [fig:007?](#)).

```
vboxuser@hiy:~/work/arch-pc/lab04$ ./hello
Hello world!
vboxuser@hiy:~/work/arch-pc/lab04$ ./main
Hello world!
vboxuser@hiy:~/work/arch-pc/lab04$
```

3 Самостоятельная работа

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

(рис. [fig:008?](#)).



```
vboxuser@hiy: ~/work/arch-pc/lab04
vboxuser@hiy:~/work/arch-pc/lab04$ cd ~/work/arch-pc/lab04
vboxuser@hiy:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
vboxuser@hiy:~/work/arch-pc/lab04$
```

С помощью текстового редактора я внесла изменения в текст программы в файле lab4.asm так, что вместо Hello world! на экран вывелась строка с моим именем и фамилией

(рис. 009?).



```
SECTION .data
    hello: DB 'Ивед Карина',10

    helloLen: EQU $-hello

SECTION .text
    GLOBAL _start

_start:
    mov eax,4
    mov ebx,1 ; Описатель файла '1'
    mov ecx,hello
    mov edx,helloLen
    int 80h

    mov eax,1
    mov ebx,0
    int 80h
```

Далее я оттранслировала полученный текст программы lab4.asm в объектный файл. Выполнила компоновку объектного файла и запустила получившийся исполняемый файл

(рис. 010?).

```
vboxuser@hiy: ~/work/arch-pc/lab04
vboxuser@hiy:~/work/arch-pc/lab04$ cd ~/work/arch-pc/lab04
vboxuser@hiy:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
vboxuser@hiy:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
vboxuser@hiy:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab.4
vboxuser@hiy:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
vboxuser@hiy:~/work/arch-pc/lab04$ ./lab4
Швед Карина
vboxuser@hiy:~/work/arch-pc/lab04$
```

Далее я скопировала файлы hello.asm и lab4.asm в свой локальный репозиторий в каталог ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04/ и загрузила файлы на Github.

(рис. 4.011?).

```
vboxuser@hiy: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04
vboxuser@hiy:~/work/arch-pc/lab04$ cd ~/work/arch-pc/lab04
vboxuser@hiy:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
vboxuser@hiy:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
vboxuser@hiy:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab.4
vboxuser@hiy:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
vboxuser@hiy:~/work/arch-pc/lab04$ ./lab4
Швед Карина
vboxuser@hiy:~/work/arch-pc/lab04$ cp hello.asm ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04/
vboxuser@hiy:~/work/arch-pc/lab04$ cp lab4.asm ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04/
vboxuser@hiy:~/work/arch-pc/lab04$ git add .
fatal: not a git repository (or any of the parent directories): .git
vboxuser@hiy:~/work/arch-pc/lab04$ cd
vboxuser@hiy:~$ git add .
fatal: not a git repository (or any of the parent directories): .git
vboxuser@hiy:~$ cd ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04/
vboxuser@hiy:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ git commit -m "new changes"
[master 7700cfb] new changes
2 files changed, 38 insertions(+)
create mode 100644 labs/lab04/hello.asm
create mode 100644 labs/lab04/lab4.asm
vboxuser@hiy:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 6 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 690 bytes | 172.00 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:karinashved/-study_2024-2025_arhpc.git
565f33c..7700cfb master -> master
vboxuser@hiy:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$
```

4 Выводы

Я освоила 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,
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.
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,
17. — 1120 с. — (Классика Computer Science)