
Front matter

title: " Лабораторная работа No4." subtitle: "Создание и процесс обработки программ на языке ассемблера NASM" author: "Арсоева Залина НБИБд-01-21"

Generic options

lang: ru-RU toc-title: "Содержание"

Bibliography

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

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

Internationalization

polyglossia-lang: name: russian options: - spelling=modern - babelshorthands=true polyglossia-otherlangs:
name: english

Babel

babel-lang: russian babel-otherlangs: english

Fonts

mainfont: PT Serif romanfont: PT Serif sansfont: PT Sans monofont: PT Mono mainfontoptions:
Ligatures=TeX romanfontoptions: Ligatures=TeX sansfontoptions: Ligatures=TeX,Scale=MatchLowercase
monofontoptions: Scale=MatchLowercase,Scale=0.9

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:

- \usepackage{indentfirst}
- \usepackage{float} # keep figures where there are in the text
- \floatplacement{figure}{H} # keep figures where there are in the text

Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM

Задание

1. Создайте каталог для работы с программами на языке ассемблера NASM:

mkdir ~/work/arch-pc/lab05

2. Перейдите в созданный каталог

cd ~/work/arch-pc/lab05

3. Создайте текстовый файл с именем hello.asm

touch hello.asm

5. Откройте этот файл с помощью любого текстового редактора, например, gedit и введите в него следующий текст:

```
; hello.asm SECTION .data ; Начало секции данных hello: DB 'Hello world!',10 ; 'Hello world!' плюс; символ перевода строки helloLen: EQU $-hello ; Длина строки hello SECTION .text ; Начало секции кода GLOBAL _start _start: ; Точка входа в программу mov eax,4 ; Системный вызов для записи (sys_write) mov ebx,1 ;  
Описатель файла '1' - стандартный вывод mov ecx,hello ; Адрес строки hello в ecx mov edx,helloLen ;  
Размер строки hello int 80h ; Вызов ядра
```

```
mov eax,1 ; Системный вызов для выхода (sys_exit) mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)  
int 80h ; Вызов ядра
```

6. Для компиляции приведённого выше текста программы «Hello World» необходимо написать:

nasm -f elf hello.asm

7. С помощью команды ls проверьте, что объектный файл был создан.
8. Выполните следующую команду:

nasm -o obj.o -f elf -g -l list.lst hello.asm

9. Чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику:

ld -m elf_i386 hello.o -o hello

10. С помощью команды `ls` проверьте, что исполняемый файл `hello` был создан
11. Выполните следующую команду:

ld -m elf_i386 obj.o -o main

12. Запустить на выполнение созданный исполняемый файл, находящийся в текущем каталоге, можно, набрав в командной строке:

./hello

Задание для самостоятельной работы

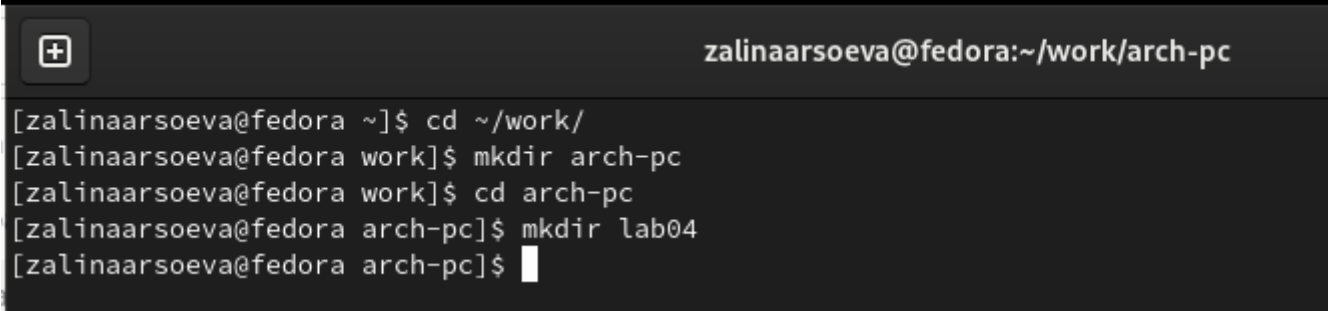
1. В каталоге `~/work/arch-pc/lab05` с помощью команды `cp` создайте копию файла `hello.asm` с именем `lab5.asm`
2. С помощью любого текстового редактора внесите изменения в текст программы в файле `lab5.asm` так, чтобы вместо `Hello world!` на экран вывела строка с вашими фамилией и именем.
3. Оттранслируйте полученный текст программы `lab5.asm` в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл.
4. Скопируйте файлы `hello.asm` и `lab5.asm` в Ваш локальный репозиторий в каталог `~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/labs/lab05/`. Загрузите файлы на Github

Выполнение лабораторной работы

1. Создаю каталог для работы с программами на языке ассемблера NASM:

mkdir ~/work/arch-pc/lab05

(рис. [-@fig:001])



```
zalinaarsoeva@fedora:~/work/arch-pc
[zalinaarsoeva@fedora ~]$ cd ~/work/
[zalinaarsoeva@fedora work]$ mkdir arch-pc
[zalinaarsoeva@fedora work]$ cd arch-pc
[zalinaarsoeva@fedora arch-pc]$ mkdir lab04
[zalinaarsoeva@fedora arch-pc]$
```

#fig:001 width=70% }

2. Перехожу в созданный каталог

cd ~/work/arch-pc/lab05

(рис. [-@fig:002])

```
[zalinaarsoeva@fedora ~]$ cd ~/work/  
[zalinaarsoeva@fedora work]$ mkdir arch-pc  
[zalinaarsoeva@fedora work]$ cd arch-pc  
[zalinaarsoeva@fedora arch-pc]$ mkdir lab04  
[zalinaarsoeva@fedora arch-pc]$ cd ~/work/arch-pc/lab04  
[zalinaarsoeva@fedora lab04]$
```

#fig:002 width=70% }

3. Создаю текстовый файл с именем hello.asm

touch hello.asm

(рис. [-@fig:003])

```
[zalinaarsoeva@fedora ~]$ cd ~/work/  
[zalinaarsoeva@fedora work]$ mkdir arch-pc  
[zalinaarsoeva@fedora work]$ cd arch-pc  
[zalinaarsoeva@fedora arch-pc]$ mkdir lab04  
[zalinaarsoeva@fedora arch-pc]$ cd ~/work/arch-pc/lab04  
[zalinaarsoeva@fedora lab04]$ touch hello.asm  
[zalinaarsoeva@fedora lab04]$
```

#fig:003 width=70% }

4. Открываю этот файл с помощью VS Code и ввожу в него следующий текст: (рис. [-@fig:004])

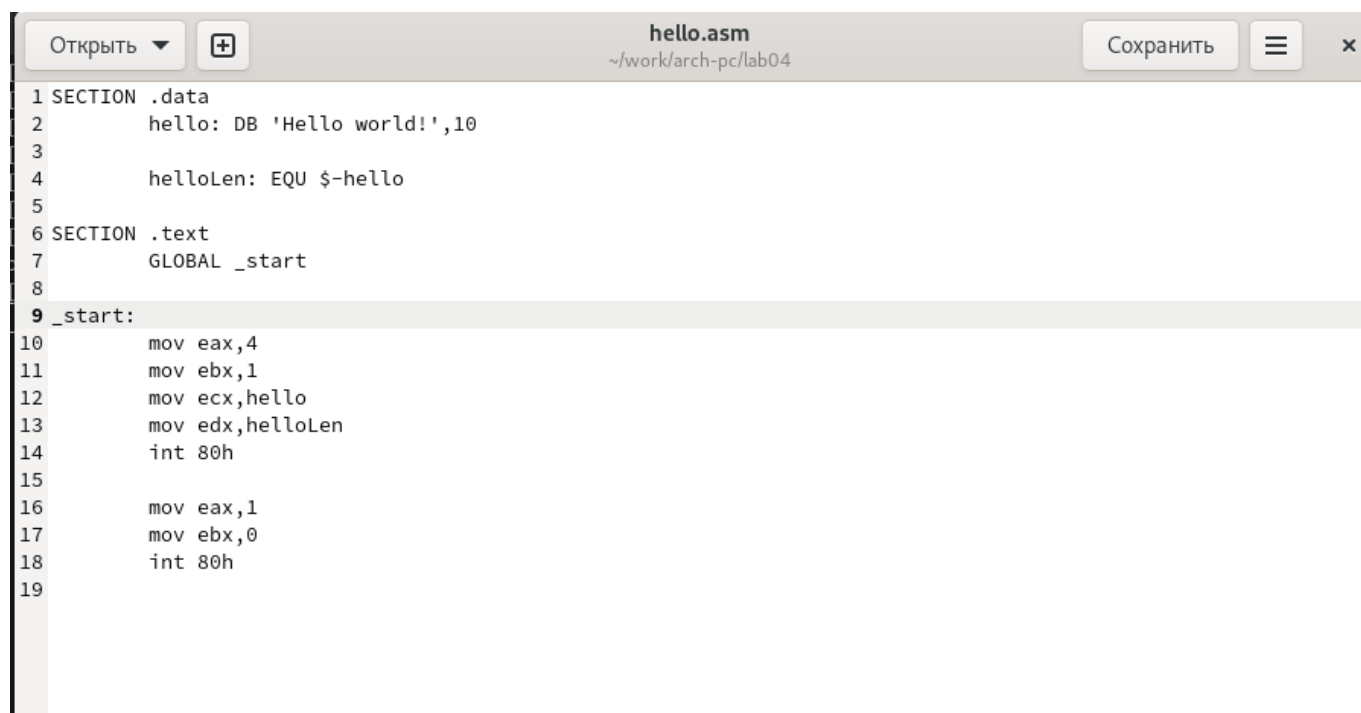


{ #fig:004 width=70% }

5. Для компиляции приведённого выше текста программы «Hello World» ввожу:

nasm -f elf hello.asm

(рис. [-@fig:005])



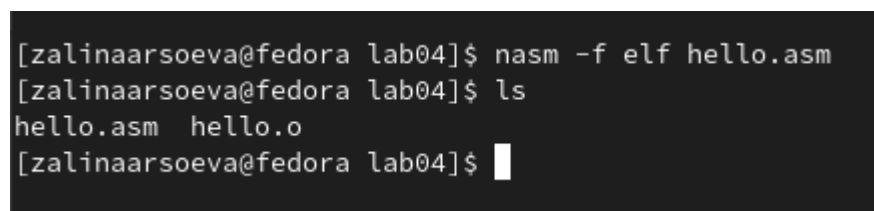
```
hello.asm
~/work/arch-pc/lab04

1 SECTION .data
2     hello: DB 'Hello world!',10
3
4     helloLen: EQU $-hello
5
6 SECTION .text
7     GLOBAL _start
8
9 _start:
10    mov eax,4
11    mov ebx,1
12    mov ecx,hello
13    mov edx,helloLen
14    int 80h
15
16    mov eax,1
17    mov ebx,0
18    int 80h
19
```

{ #fig:005 width=70% }

6. С помощью команды ls проверяю, что объектный файл был создан.

(рис. [-@fig:006])



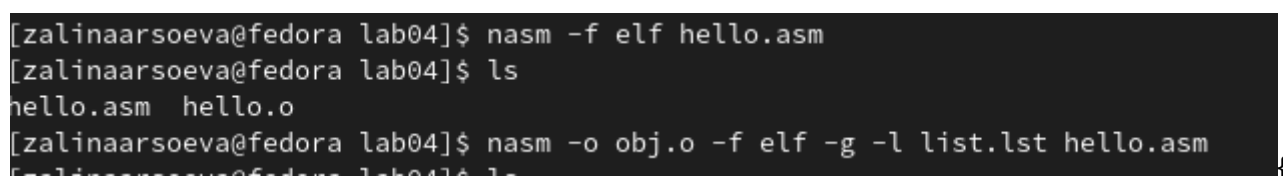
```
[zalinaarsoeva@fedora lab04]$ nasm -f elf hello.asm
[zalinaarsoeva@fedora lab04]$ ls
hello.asm  hello.o
[zalinaarsoeva@fedora lab04]$
```

{ #fig:006 width=70% }

7. Выполняю следующую команду:

nasm -o obj.o -f elf -g -l list.lst hello.asm

(рис. [-@fig:007])



```
[zalinaarsoeva@fedora lab04]$ nasm -f elf hello.asm
[zalinaarsoeva@fedora lab04]$ ls
hello.asm  hello.o
[zalinaarsoeva@fedora lab04]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
[zalinaarsoeva@fedora lab04]$ ls
hello.asm  hello.o  obj.o
```

#fig:007 width=70% }

8. Чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику:

ld -m elf_i386 hello.o -o hello

(рис. [-@fig:008])

```
[zalinaarsoeva@fedora lab04]$ ld -m elf_i386 hello.o -o hello
[zalinaarsoeva@fedora lab04]$ ls
hello hello.asm hello.o list.lst obj.o
[zalinaarsoeva@fedora lab04]$
```

{ #fig:008 width=70% }

9. С помощью команды ls проверяю, что исполняемый файл hello был создан

(рис. [-@fig:009])

```
[zalinaarsoeva@fedora lab04]$ ld -m elf_i386 hello.o -o hello
[zalinaarsoeva@fedora lab04]$ ls
hello hello.asm hello.o list.lst obj.o
[zalinaarsoeva@fedora lab04]$ ld -m elf_i386 obj.o -o main
```

#fig:009 width=70% }

10. Выполняю следующую команду:

ld -m elf_i386 obj.o -o main

(рис. [-@fig:010])

```
hello hello.asm hello.o list.lst obj.o
[zalinaarsoeva@fedora lab04]$ ld -m elf_i386 obj.o -o main
[zalinaarsoeva@fedora lab04]$ ls
hello hello.asm hello.o list.lst main obj.o
[zalinaarsoeva@fedora lab04]$
```

{ #fig:010 width=70% }

11. Запускаю на выполнение созданный исполняемый файл, находящийся в текущем каталоге, можно, набрав в командной строке:

./hello

(рис. [-@fig:011])

```
hello hello.asm hello.o list.lst main obj.o
[zalinaarsoeva@fedora lab04]$ ./hello
Hello world!
[zalinaarsoeva@fedora lab04]$
```

{ #fig:011 width=70% }

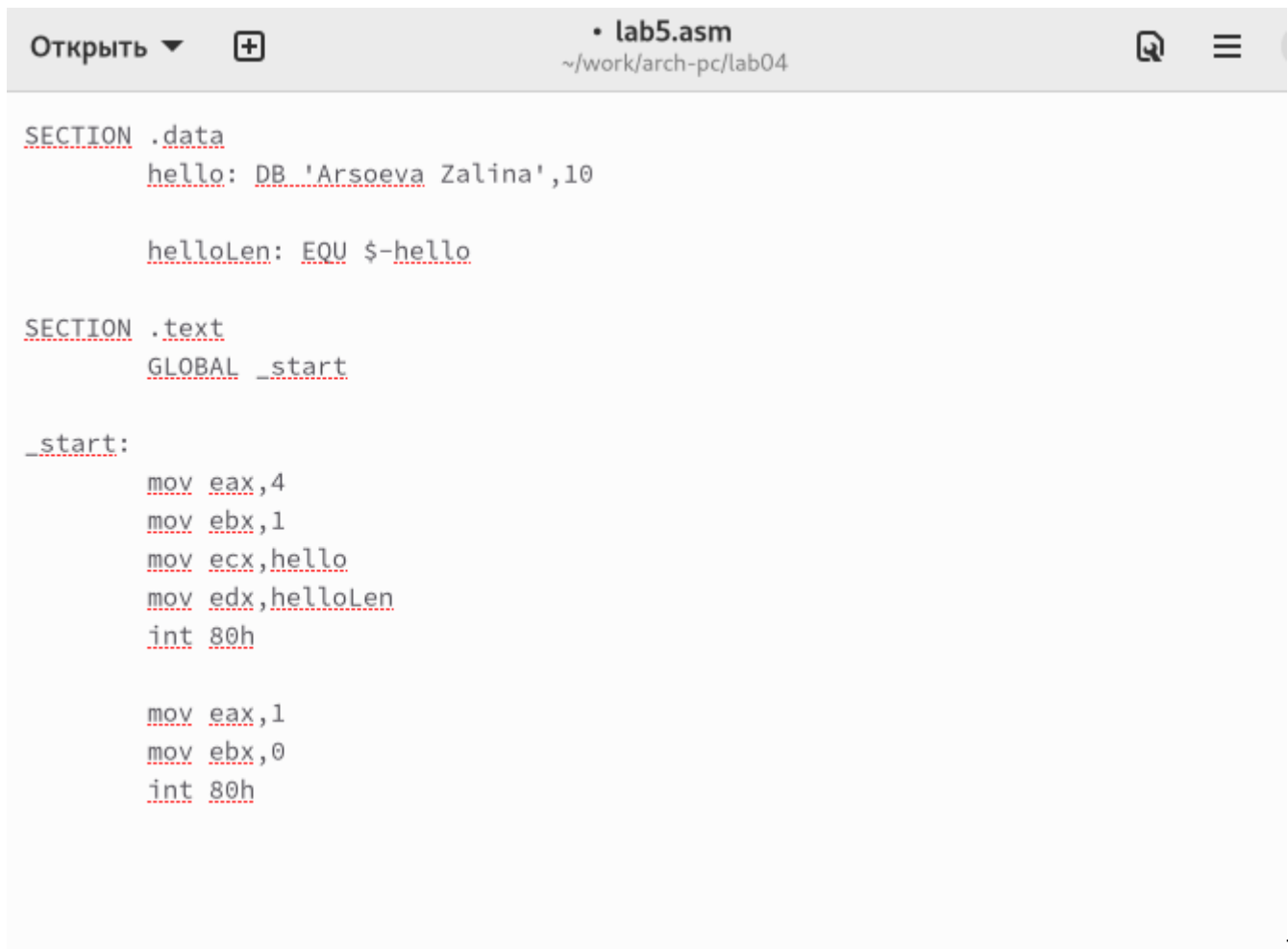
Задание для самостоятельной работы

1. В каталоге ~/work/arch-pc/lab05 с помощью команды cp создаю копию файла hello.asm с именем lab5.asm (рис. [-@fig:012])

```
Hello world!
[zalinaarsoeva@fedora lab04]$ cp hello.asm lab5.asm
[zalinaarsoeva@fedora lab04]$ ls
hello hello.asm hello.o lab5.asm list.lst main obj.o
[zalinaarsoeva@fedora lab04]$
```

{ #fig:012 width=70% }

2. С помощью любого текстового редактора вношу изменения в текст программы в файле lab5.asm так, чтобы вместо Hello world! на экран вывodiлась строка с моими фамилией и именем. (рис. [-@fig:013])



```

SECTION .data
    hello: DB 'Arsoeva Zalina',10

    helloLen: EQU $-hello

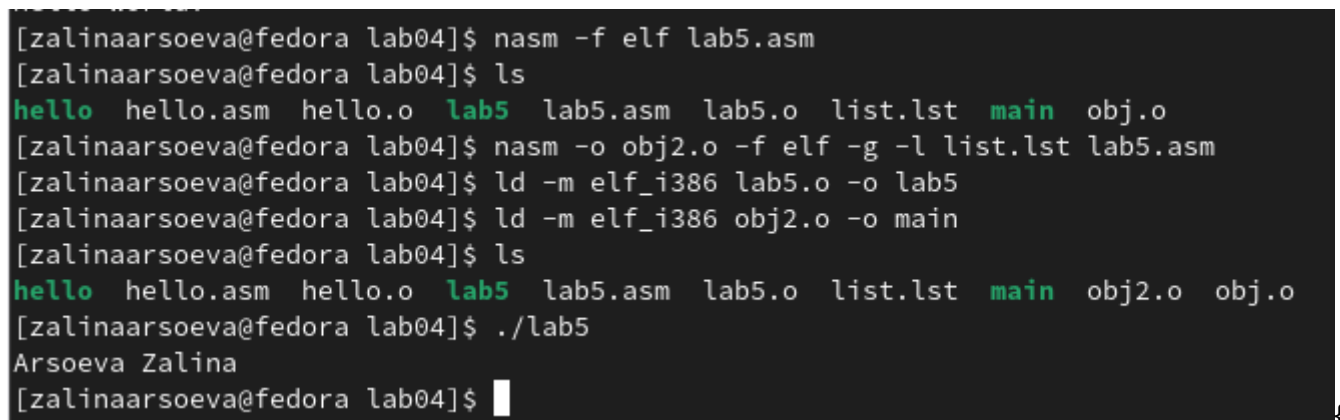
SECTION .text
    GLOBAL _start

_start:
    mov eax,4
    mov ebx,1
    mov ecx,hello
    mov edx,helloLen
    int 80h

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

#fig:013 width=70% }

3. Оттранслирую полученный текст программы lab5.asm в объектный файл. Выполняю компоновку объектного файла и запускаю получившийся исполняемый файл. (рис. [-@fig:014])



```

[zalinaarsoeva@fedora lab04]$ nasm -f elf lab5.asm
[zalinaarsoeva@fedora lab04]$ ls
hello hello.asm hello.o lab5 lab5.asm lab5.o list.lst main obj.o
[zalinaarsoeva@fedora lab04]$ nasm -o obj2.o -f elf -g -l list.lst lab5.asm
[zalinaarsoeva@fedora lab04]$ ld -m elf_i386 lab5.o -o lab5
[zalinaarsoeva@fedora lab04]$ ld -m elf_i386 obj2.o -o main
[zalinaarsoeva@fedora lab04]$ ls
hello hello.asm hello.o lab5 lab5.asm lab5.o list.lst main obj2.o obj.o
[zalinaarsoeva@fedora lab04]$ ./lab5
Arsoeva Zalina
[zalinaarsoeva@fedora lab04]$
  
```

#fig:014 width=70% }

4. Копирую файлы hello.asm и lab5.asm в локальный репозиторий в каталог ~/work/study/2022-2023/"Архитектура компьютера"/arch- pc/labs/lab05/. Загружаю файлы на Github (рис. [-@fig:015])

```
[zalinaarsoeva@fedora lab04]$ cp hello.asm lab5.asm ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/labs/lab04/  
[zalinaarsoeva@fedora lab04]$ cd ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/labs/lab04  
[zalinaarsoeva@fedora lab04]$ ls  
hello.asm lab5.asm presentation report  
[zalinaarsoeva@fedora lab04]$ git add .  
[zalinaarsoeva@fedora lab04]$ git commit -am 'feat(main): add files - lab04'  
[master 18ec0d2] feat(main): add files - lab04  
3 files changed, 38 insertions(+), 114 deletions(-)  
delete mode 100644 labs/lab03/report/report.docs  
create mode 100644 labs/lab04/hello.asm  
create mode 100644 labs/lab04/lab5.asm  
[zalinaarsoeva@fedora lab04]$ git push  
Перечисление объектов: 13, готово.  
Подсчет объектов: 100% (13/13), готово.  
Сжатие объектов: 100% (8/8), готово.  
Запись объектов: 100% (8/8), 808 байтов | 808.00 КиБ/с, готово.  
Всего 8 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов 0  
remote: Resolving deltas: 100% (4/4), completed with 3 local objects.  
To github.com:ArsoevaZalina/study_2022-2023_arh-pc-.git  
c3f6284..18ec0d2 master -> master  
[zalinaarsoeva@fedora lab04]$
```

{ #fig:015 width=70% }

Выводы

Я освоила процедуры компиляции и сборки программ, написанных на ассем- блере NASM.

Список литературы{.unnumbered}
