

Шаблон отчёта по лабораторной работе

Простейший вариант

Алади Принц Чисом; НКАБД-05-22

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	17
	Список литературы	18

Список иллюстраций

Список таблиц

1 Цель работы

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

2 Задание

1. Программа Hello world!
2. Транслятор NASM, Расширенный синтаксис командой строки NASM и Компоновщик LD.
3. Запуск исполняемого файла
4. Задание для самостоятельной работы
5. В каталоге `~/work/arch-рс/lab05` с помощью команды `ср` создайте копию файла `hello.asm` с именем `lab5.asm`
6. С помощью любого текстового редактора внесите изменения в текст программы в файле `lab5.asm` так, чтобы вместо `Hello world!` на экран выводилась строка с вашими фамилией и именем.
7. Оттранслируйте полученный текст программы `lab5.asm` в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл.
8. Скопируйте файлы `hello.asm` и `lab5.asm` в Ваш локальный репозиторий в каталог `~/work/study/2022-2023/“Архитектура компьютера”/arch-рс/labs/lab05/`. Загрузите файлы на Github.

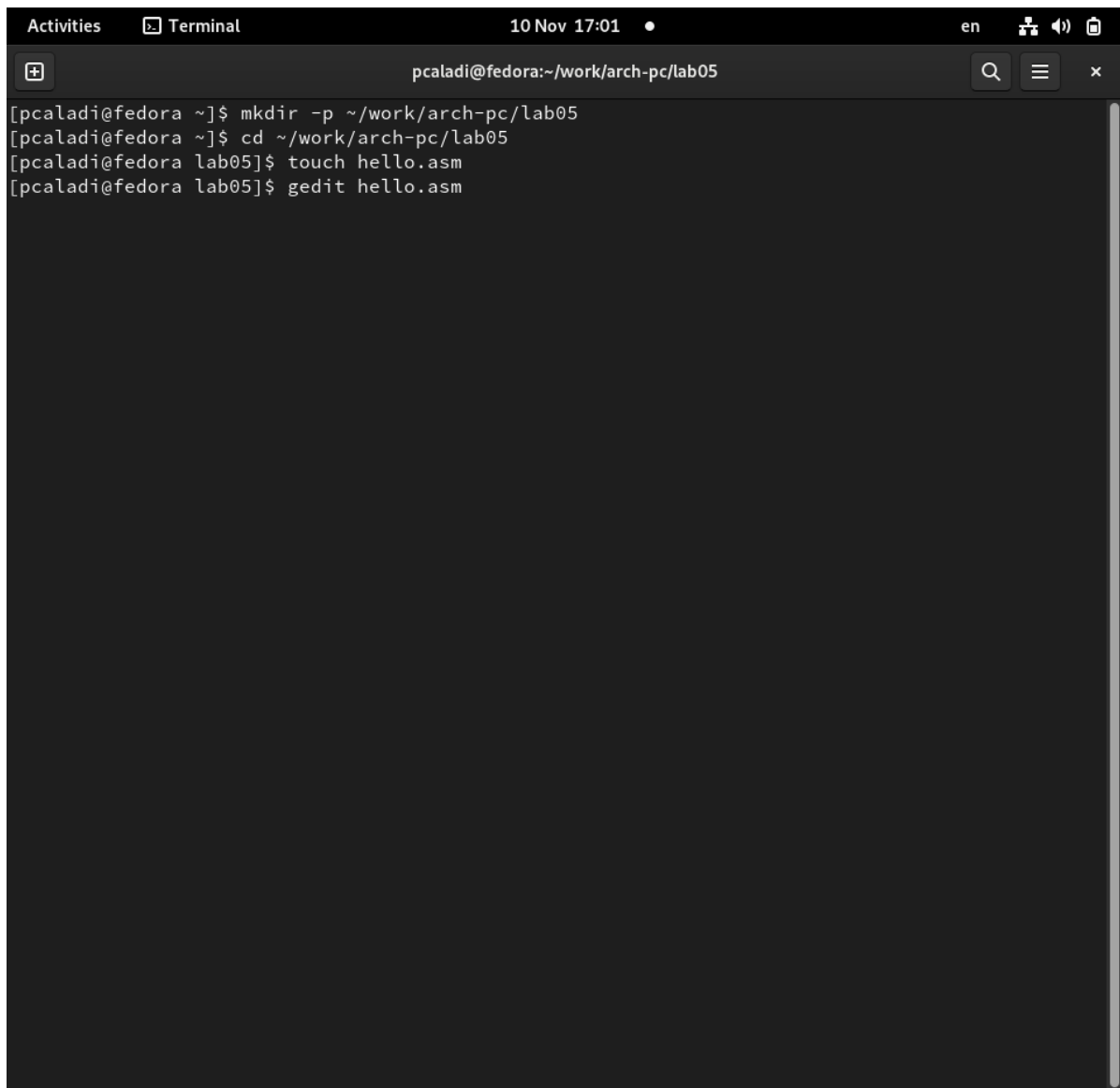
3 Теоретическое введение

Основными функциональными элементами любой электронно-вычислительной машины (ЭВМ) являются центральный процессор, память и периферийные устройства. Взаимодействие этих устройств осуществляется через общую шину, к которой они подключены. Физически шина представляет собой большое количество проводников, соединяющих устройства друг с другом. В современных компьютерах проводники выполнены в виде электропроводящих дорожек на материнской (системной) плате.

Основной задачей процессора является обработка информации, а также организация координации всех узлов компьютера. В состав центрального процессора (ЦП) входят следующие устройства: • арифметико-логическое устройство (АЛУ) — выполняет логические и арифметические действия, необходимые для обработки информации, хранящейся в памяти; • устройство управления (УУ) — обеспечивает управление и контроль всех устройств компьютера; • регистры — сверхбыстрая оперативная память небольшого объёма, входящая в состав процессора, для временного хранения промежуточных результатов выполнения инструкций; регистры процессора делятся на два типа: регистры общего назначения и специальные регистры.

4 Выполнение лабораторной работы

1. Программа Hello world! Я создал каталог для работы с программами на ассемблере NASM и открыл файл в текстовом редакторе. После этого я ввел текст, показанный в описании изображения ниже. (рис.??, ??)

A terminal window titled "Terminal" with a dark background. The window shows a series of commands and their outputs. The user "pcaladi" is at a Fedora machine. The commands executed are: "mkdir -p ~/work/arch-pc/lab05", "cd ~/work/arch-pc/lab05", "touch hello.asm", and "gedit hello.asm". The terminal window has a title bar with "Activities", "Terminal", and the date/time "10 Nov 17:01". The window also has a search icon, a menu icon, and a close icon. The terminal output shows the prompt changing from "~" to "lab05" after the "cd" command.

```
pcaladi@fedora:~/work/arch-pc/lab05
[pcaladi@fedora ~]$ mkdir -p ~/work/arch-pc/lab05
[pcaladi@fedora ~]$ cd ~/work/arch-pc/lab05
[pcaladi@fedora lab05]$ touch hello.asm
[pcaladi@fedora lab05]$ gedit hello.asm
```

#fig:001

width=70%}



```
; 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

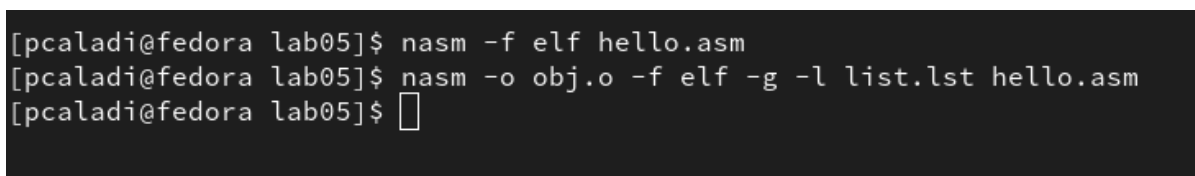
_start:                      ; Точка входа в программу
    mov eax,4                ; Системный вызов для записи (sys_write)
    mov ebx,1                ; Описатель файла '1' - стандартный вывод
    mov ecx,hello            ; Адрес строки hello в ecx
    mov edx,helloLen         ; Размер строки hello
    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
```

#fig:002

width=70%}

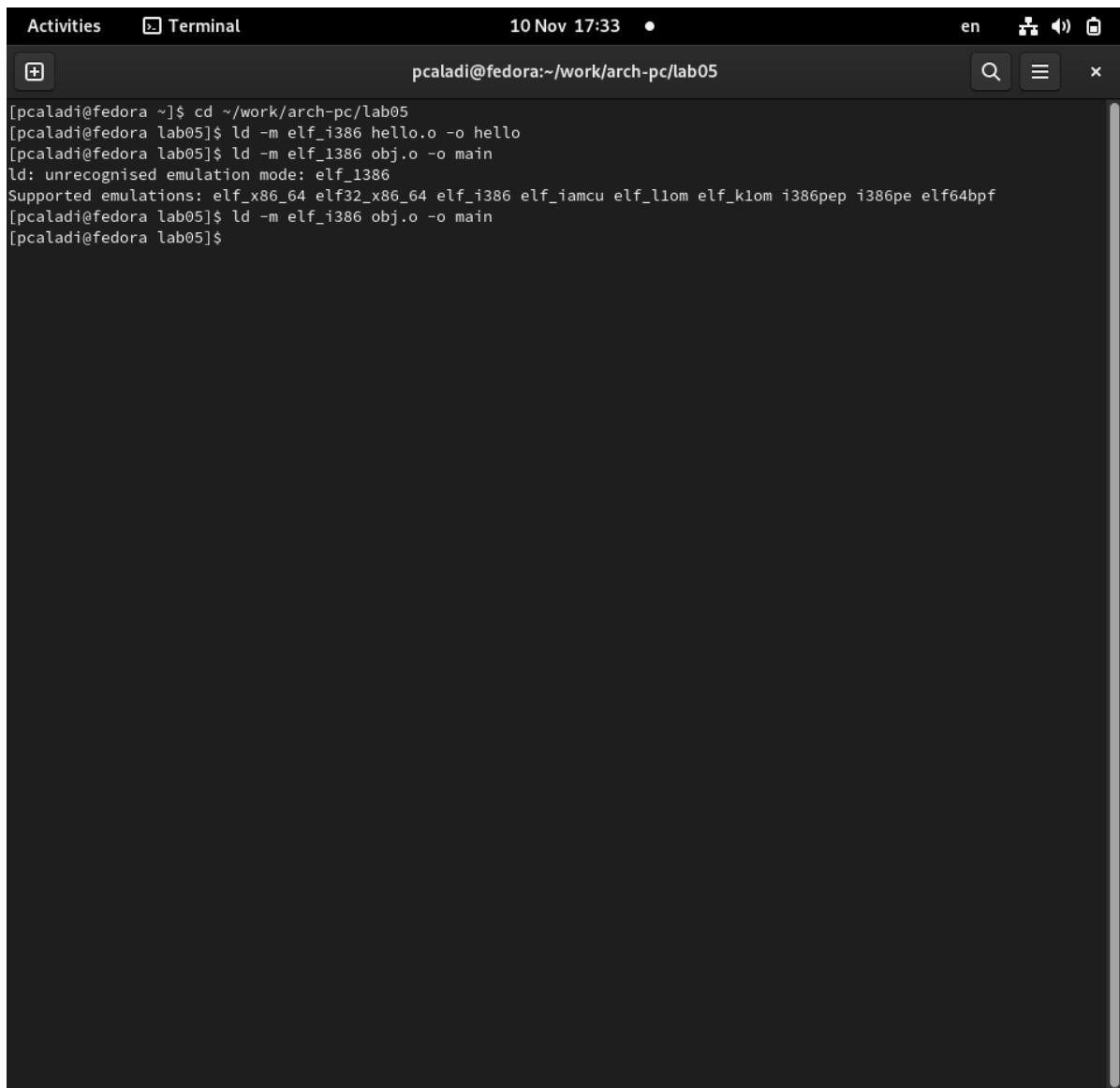
2. Транслятор NASM, Расширенный синтаксис командой строки NASM и Компоновщик LD. Я скомпилировал программу «Hello World», используя NASM, расширенный синтаксис командной строки NASM и компоновщик LD, как показано на рисунке ниже. (рис.??, ??)



```
[pcaladi@fedora lab05]$ nasm -f elf hello.asm
[pcaladi@fedora lab05]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
[pcaladi@fedora lab05]$
```

#fig:003

width=70%}

A terminal window titled "Terminal" with a dark background. The window shows a series of commands and their outputs. The user navigates to a directory and attempts to link two object files using the linker (ld) with the -m elf_i386 flag. The linker reports an "unrecognised emulation mode: elf_i386" and lists supported emulations. The user then repeats the command, but the error persists.

```
Activities Terminal 10 Nov 17:33 en
pcaladi@fedora:~/work/arch-pc/lab05
[pcaladi@fedora ~]$ cd ~/work/arch-pc/lab05
[pcaladi@fedora lab05]$ ld -m elf_i386 hello.o -o hello
[pcaladi@fedora lab05]$ ld -m elf_i386 obj.o -o main
ld: unrecognised emulation mode: elf_i386
Supported emulations: elf_x86_64 elf32_x86_64 elf_i386 elf_iamcu elf_llom elf_k1om i386pep i386pe elf64bpf
[pcaladi@fedora lab05]$ ld -m elf_i386 obj.o -o main
[pcaladi@fedora lab05]$
```

#fig:004

width=70%}

3. Запуск исполняемого файла Я выполнил сгенерированный исполняемый файл, расположенный в каталоге. (рис.??)

```

[pcaladi@fedora lab05]$ ./hello
Hello world!
[pcaladi@fedora lab05]$ ls
hello  hello.asm  hello.o  lab5  lab5.asm  lab5.o  list.lst  main  obj.o
[pcaladi@fedora lab05]$ cat hello.asm
; 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
[pcaladi@fedora lab05]$

```

#fig:005

width=70%}

4. Задание для самостоятельной работы
5. В каталоге ~/work/arch-pc/lab05 я использовал команду `ср` для создания файла `hello.asm` с именем `lab5.asm`. (рис.??)

```

hello world!
[pcaladi@fedora lab05]$ cp hello.asm lab5.asm
[pcaladi@fedora lab05]$ ls
hello hello.asm hello.o lab5.asm list.lst main obj.o
[pcaladi@fedora lab05]$ cat lab5.asm
; 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

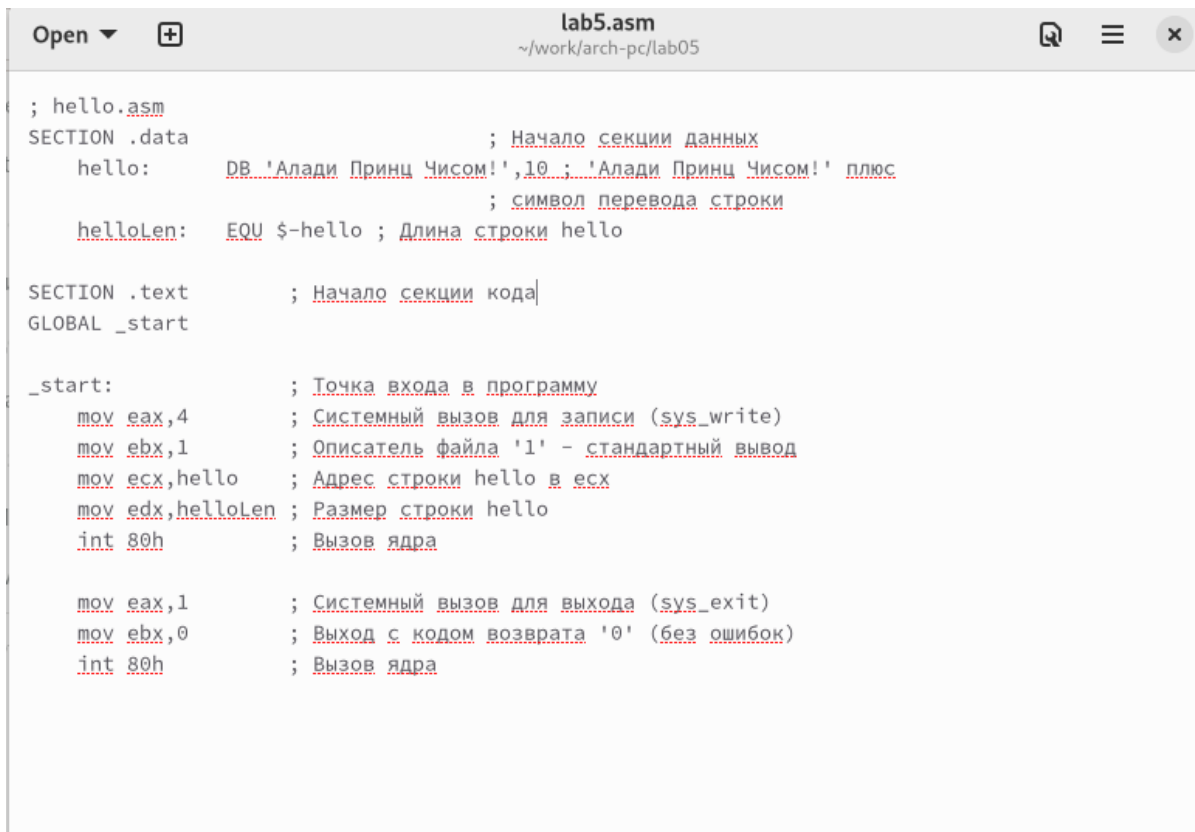
    mov eax,1                                ; Системный вызов для выхода (sys_exit)
    mov ebx,0                                ; Выход с кодом возврата '0' (без ошибок)
    int 80h                                  ; Вызов ядра
[pcaladi@fedora lab05]$

```

#fig:006

width=70%}

2. С помощью текстового редактора я отредактировал текст программы в файле lab5.asm, чтобы вместо Hello world! на экране отображалась строка моего имени и фамилии.(рис.??)



```
Open  + lab5.asm
~ /work/arch-pc/lab05

; hello.asm
SECTION .data ; Начало секции данных
    hello:     DB 'Алади Принц Чисом!',10 ; 'Алади Принц Чисом!' плюс
                                           ; символ перевода строки
    helloLen:   EQU $-hello ; Длина строки hello

SECTION .text ; Начало секции кода
GLOBAL _start

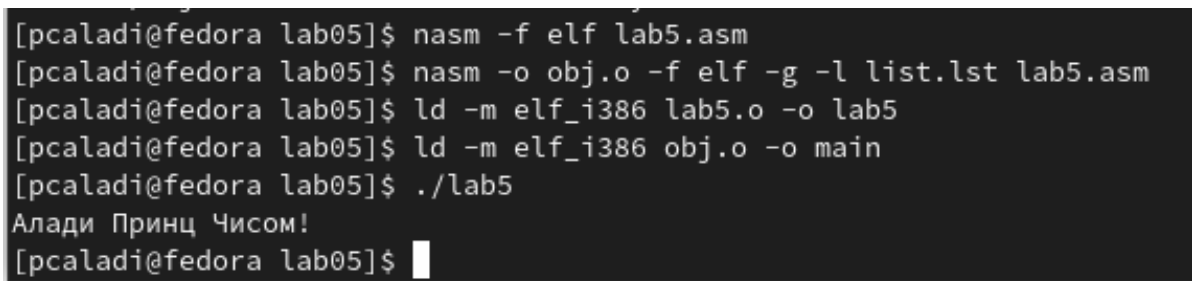
_start:        ; Точка входа в программу
    mov eax,4   ; Системный вызов для записи (sys_write)
    mov ebx,1   ; Описатель файла '1' - стандартный вывод
    mov ecx,hello ; Адрес строки hello в ecx
    mov edx,helloLen ; Размер строки hello
    int 80h     ; Вызов ядра

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

#fig:007

width=70%}

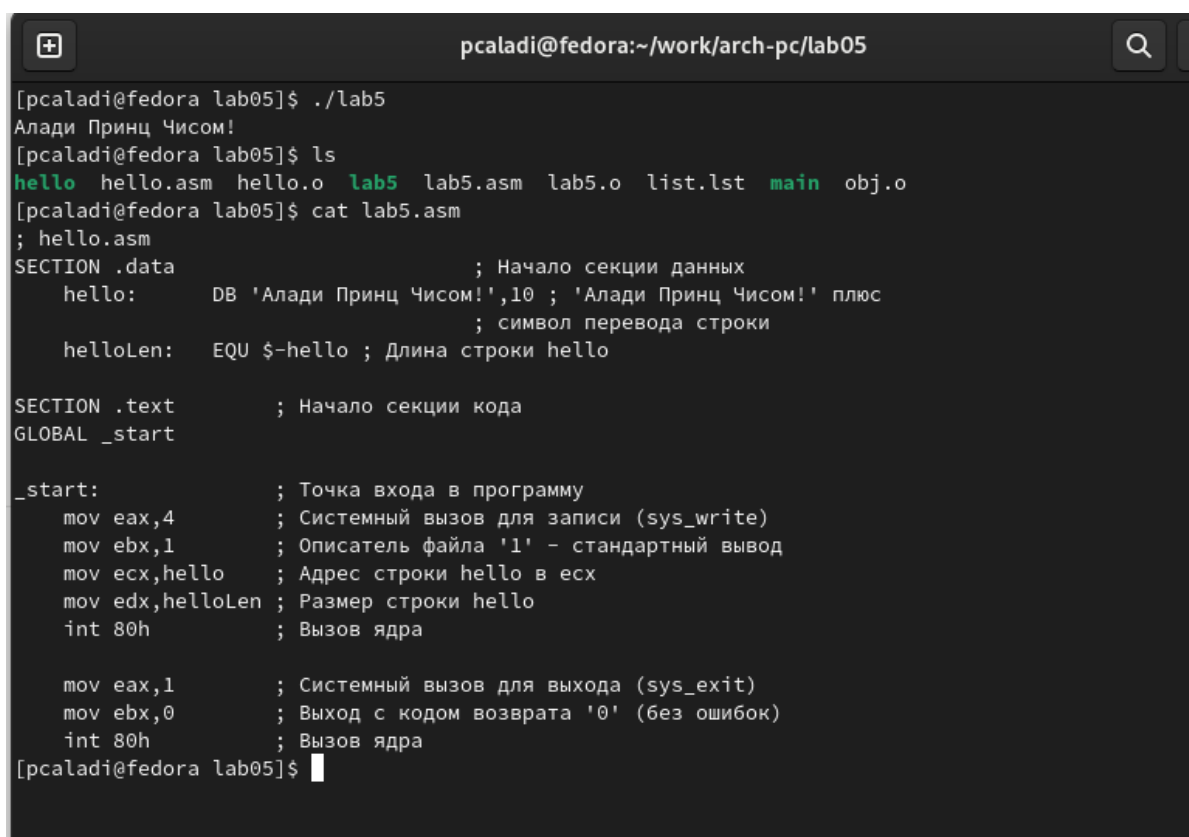
3. Я перевел получившийся текст программы lab5.asm в объектный файл. Связал объект и запустил полученный исполняемый файл.(рис.??, ??)



```
[pcaladi@fedora lab05]$ nasm -f elf lab5.asm
[pcaladi@fedora lab05]$ nasm -o obj.o -f elf -g -l list.lst lab5.asm
[pcaladi@fedora lab05]$ ld -m elf_i386 lab5.o -o lab5
[pcaladi@fedora lab05]$ ld -m elf_i386 obj.o -o main
[pcaladi@fedora lab05]$ ./lab5
Алади Принц Чисом!
[pcaladi@fedora lab05]$
```

#fig:008

width=70%}



A terminal window titled 'pcaladi@fedora:~/work/arch-pc/lab05'. The user runs './lab5', which outputs 'Алади Принц Чисом!'. Then, the user runs 'ls', showing files 'hello.asm', 'hello.o', 'lab5', 'lab5.asm', 'lab5.o', 'list.lst', 'main', and 'obj.o'. Finally, the user runs 'cat lab5.asm', displaying the following assembly code:

```
[pcaladi@fedora lab05]$ ./lab5
Алади Принц Чисом!
[pcaladi@fedora lab05]$ ls
hello hello.asm hello.o lab5 lab5.asm lab5.o list.lst main obj.o
[pcaladi@fedora lab05]$ cat lab5.asm
; hello.asm
SECTION .data                                ; Начало секции данных
    hello:      DB 'Алади Принц Чисом!',10 ; 'Алади Принц Чисом!' плюс
                                           ; символ перевода строки
    helloLen:   EQU $-hello ; Длина строки hello

SECTION .text                                ; Начало секции кода
GLOBAL _start

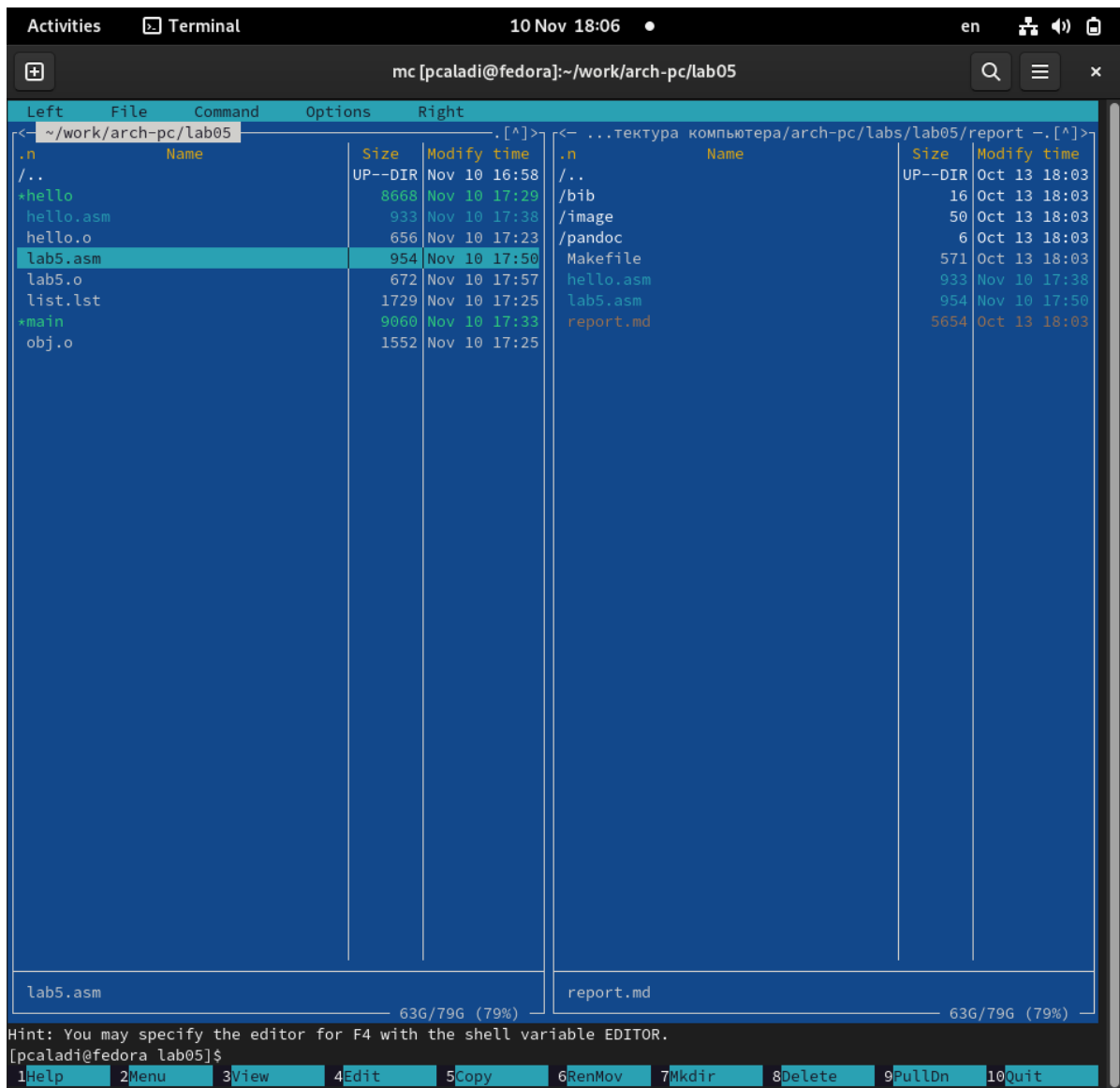
_start:                                       ; Точка входа в программу
    mov eax,4                                ; Системный вызов для записи (sys_write)
    mov ebx,1                                ; Описатель файла '1' - стандартный вывод
    mov ecx,hello                            ; Адрес строки 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
[pcaladi@fedora lab05]$
```

#fig:009

width=70%}

4. Я скопировал файлы hello.asm и lab5.asm в свой локальный репозиторий, а также загрузил файлы на github.(рис.??



#fig:010

width=70%}

5 Выводы

В ходе этой лабораторной работы я приобрел практические навыки освоения процедур компиляции и ассемблера программ, написанных на ассемблере NASM.

Список литературы