

# **Отчёт по лабораторной работе №5**

**Дисциплина: Архитектура компьютеров**

Филиппева Ксения Дмитриевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>6</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>5</b>	<b>Выполнение самостоятельной работы</b>	<b>11</b>
<b>6</b>	<b>Выводы</b>	<b>13</b>

# Список иллюстраций

4.1	Запуск файлового менеджера . . . . .	7
4.2	Создание папки “lab05” . . . . .	8
4.3	Создание файла в формате ассемблера . . . . .	8
4.4	Открытие файла в редакторе . . . . .	8
4.5	Проверка наличия текста . . . . .	9
4.6	Проверка корректности работы файла . . . . .	9
4.7	Скачанный файл в папке . . . . .	9
4.8	Вывод команды из файла . . . . .	10
5.1	Вывод измененной программы . . . . .	11
5.2	Вывод измененного файла . . . . .	11
5.3	Файлы на Github . . . . .	12

# 1 Цель работы

Приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера `mov` и `int`.

## 2 Задание

1. Выполнение лабораторной работы №5
2. Заполнение отчета по выполнению лабораторной работы №5 с помощью языка разметки Markdown
3. Выполнение заданий для самостоятельной работы

### **3 Теоретическое введение**

Midnight Commander (или просто mc) — это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управлению файловой системой, т.е. mc является файловым менеджером. Midnight Commander позволяет сделать работу с файлами более удобной и наглядной.

## 4 Выполнение лабораторной работы

Открываем терминал, запускаем файловый менеджер Midnight Commander с помощью команды “mc”. (рис. 4.1)

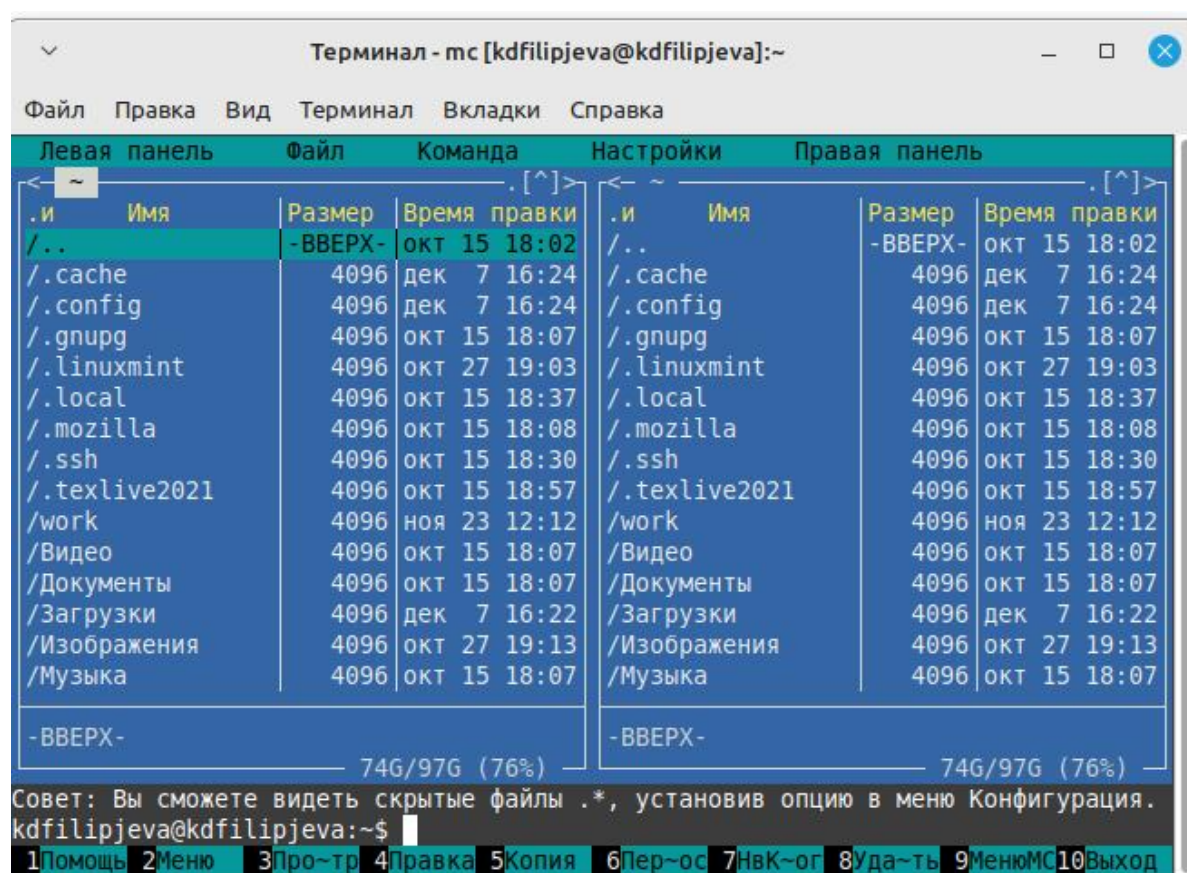


Рис. 4.1: Запуск файлового менеджера

После того, как открыли MC, переходим в каталог, созданный в ходе прошлой ЛР и создаем папку “lab05” используя клавишу F7. (рис. 4.2)

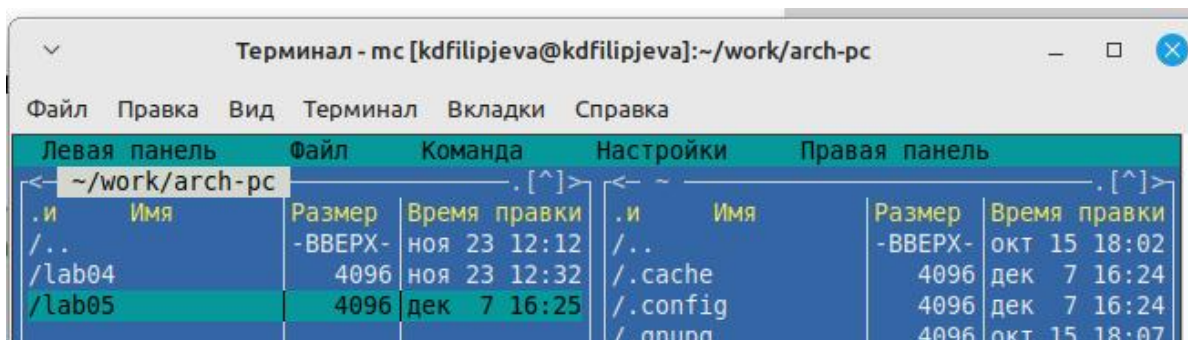


Рис. 4.2: Создание папки “lab05”

После того, как папка была создана, создадим в ней файл “lab5-1.asm”. (рис. 4.3)

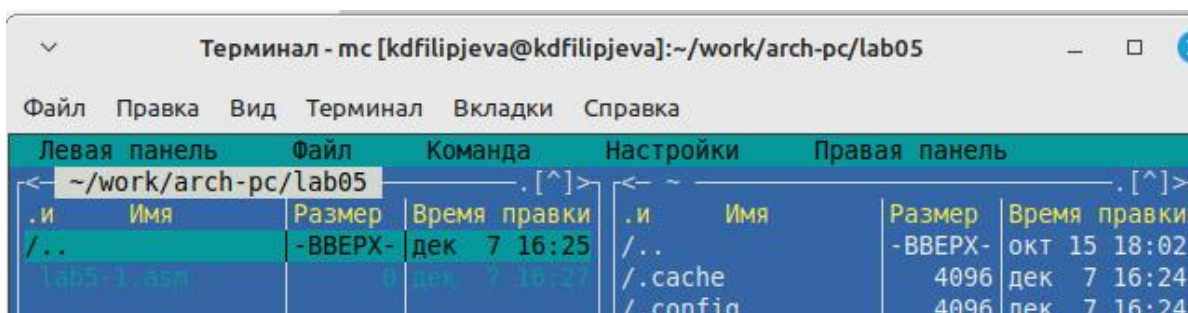


Рис. 4.3: Создание файла в формате ассемблера

С помощью клавиши F4 откроем файл “lab5-1.asm” с помощью встроенного редактора MC - nano. (рис. 4.4)

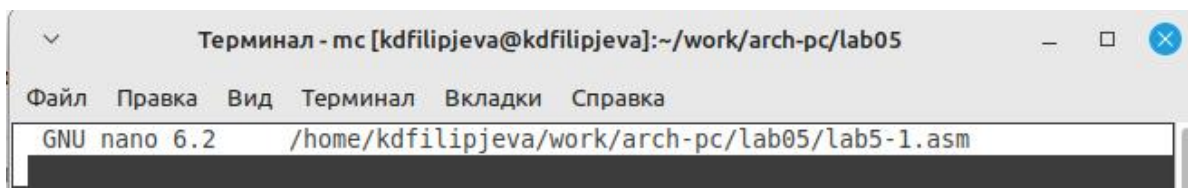


Рис. 4.4: Открытие файла в редакторе

Введем текст из листинга, сохраним изменения и проверим файл на наличие текста программы с помощью клавиши F3. (рис. 4.5)



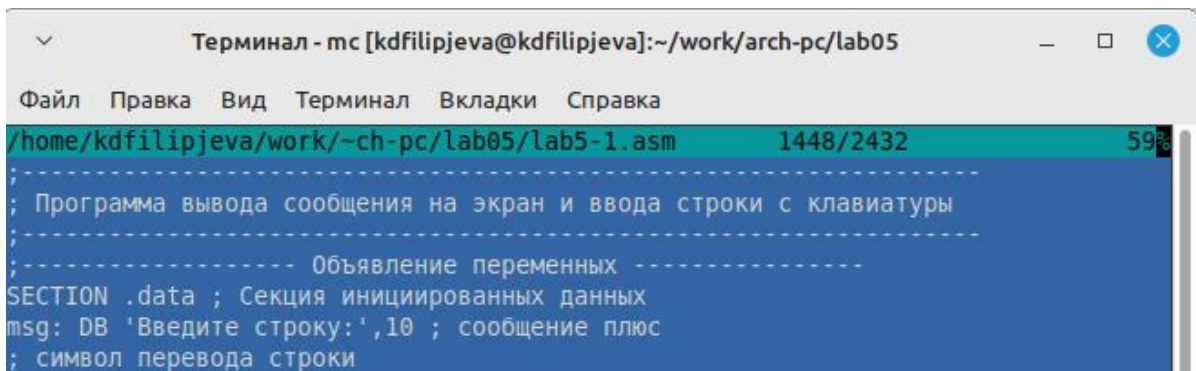


Рис. 4.5: Проверка наличия текста

Оттранслируем файл в объектный, и проверим корректность работы. (рис. 4.6)

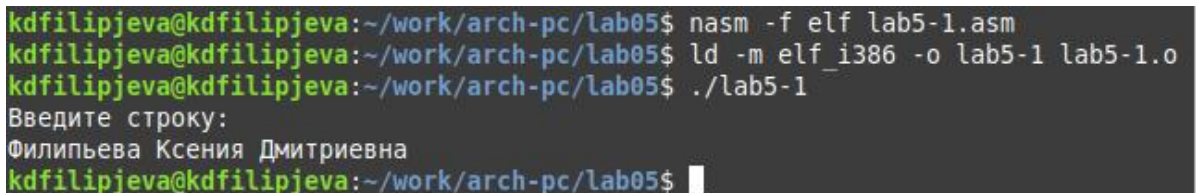


Рис. 4.6: Проверка корректности работы файла

Предварительно скачав файл “in\_out.asm” с ТУИСа добавим его в нашу папку. Так же создадим файл “lab5-2.asm”. (рис. 4.7)

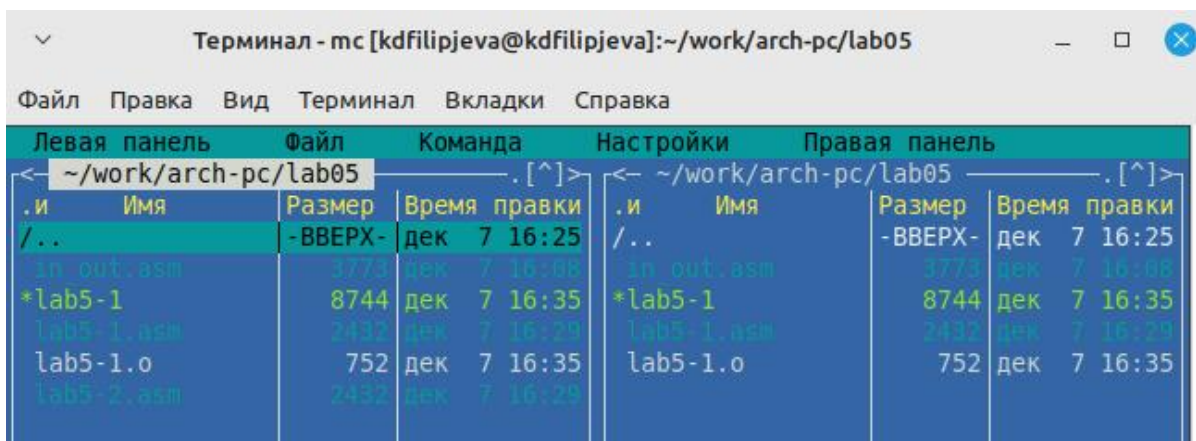


Рис. 4.7: Скачанный файл в папке

Исправим текст программы в файле “lab5-2.asm” с использованием команд из файла “in\_out.asm”. Создадим объектный файл. Из-за изменения команды “sprintLF” на “sprint” наша строка стала выводиться сразу после запроса, а не с новой строки. (рис. 4.8)

```
kdfilipjeva@kdfilipjeva:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
kdfilipjeva@kdfilipjeva:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
kdfilipjeva@kdfilipjeva:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Филиппева Ксения Дмитриевна
```

Рис. 4.8: Вывод команды из файла

## 5 Выполнение самостоятельной работы

1. Создадим копию файла “lab5-1.asm” и внесем изменения, чтобы при вводе текста, программа выводила его повторно. Превратим его в объектный файл. Проверим правильность внесенных изменений. (рис. 5.1)

```
kdfilipjeva@kdfilipjeva:~/work/arch-pc/lab05$ nasm -f elf lab51.asm
kdfilipjeva@kdfilipjeva:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab51 lab51.o
kdfilipjeva@kdfilipjeva:~/work/arch-pc/lab05$ ./lab51
Введите строку:
check
check
kdfilipjeva@kdfilipjeva:~/work/arch-pc/lab05$
```

Рис. 5.1: Вывод измененной программы

2. Создадим копию файла “lab5-2.asm” и внесем изменения, используя команды из файла “in\_out.asm”, так, чтобы при вводе текста, программа выводила его повторно. Превратим его в объектный файл и проверим правильность выполнения. (рис. 5.2)

```
kdfilipjeva@kdfilipjeva:~/work/arch-pc/lab05$ nasm -f elf lab52.asm
kdfilipjeva@kdfilipjeva:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab52 lab52.o
kdfilipjeva@kdfilipjeva:~/work/arch-pc/lab05$ ./lab52
Введите строку: check
check
kdfilipjeva@kdfilipjeva:~/work/arch-pc/lab05$
```

Рис. 5.2: Вывод измененного файла

3. Скопируем все файлы в локальный репозиторий и выгрузим на Github (рис. 5.3)

kd Filipjeva / study\_2023-2024\_arh-pc Public  
generated from [yamadharma/course-directory-student-template](#)

<> Code Issues Pull requests Actions Projects Security Insights

**Files**

master

Go to file

- config
- labs
  - lab01
  - lab02
  - lab03
  - lab04
  - lab05
  - presentation
  - report**
    - bib
    - image
    - pandoc
    - Makefile
    - in\_out.asm
    - lab5-1
    - lab5-1.asm
    - lab5-1.o
    - lab5-2
    - lab5-2.asm
    - lab5-2.o
    - lab51

study\_2023-2024\_arh-pc / labs / lab05 / report

kd Filipjeva feat(main): add files lab-5

Name	Last commit message
..	
bib	feat(main): make course structure
image	feat(main): add files lab-5
pandoc	feat(main): make course structure
Makefile	feat(main): make course structure
in_out.asm	feat(main): add files lab-5
lab5-1	feat(main): add files lab-5
lab5-1.asm	feat(main): add files lab-5
lab5-1.o	feat(main): add files lab-5
lab5-2	feat(main): add files lab-5
lab5-2.asm	feat(main): add files lab-5
lab5-2.o	feat(main): add files lab-5
lab51	feat(main): add files lab-5

Рис. 5.3: Файлы на Github

## **6 Выводы**

В ходе лабораторной работы мы освоили процедуры компиляции и сборки программ на языке ассемблера.