

# **Лабораторная работа №5**

**Архитектура вычислительных систем**

Зарифбеков Амир

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>4</b>	<b>Выполнение самостоятельной работы</b>	<b>13</b>
<b>5</b>	<b>Выводы</b>	<b>15</b>

# Список иллюстраций

3.1	1.png	. . . . .	6
3.2	2.png	. . . . .	7
3.3	3.png	. . . . .	8
3.4	4.png	. . . . .	9
3.5	5.png	. . . . .	10
3.6	6.png	. . . . .	11
3.7	7.png	. . . . .	11
3.8	8.png	. . . . .	12
3.9	9.png	. . . . .	12
4.1	10.png	. . . . .	14

# 1 Цель работы

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

## 2 Задание

1. Создайте копию файла lab6-1.asm. Внесите изменения в программу (без использования
2. Получите исполняемый файл и проверьте его работу. На приглашение ввести строку
3. Создайте копию файла lab6-2.asm. Исправьте текст программы с использованием
4. Создайте исполняемый файл и проверьте его работу.

### 3 Выполнение лабораторной работы

- 1) . Откроем Midnight Commander используя команду mc. Пользуясь клавишами **←**, **→** и Enter перейдем в каталог созданный при выполнении лабораторной работы №4. С помощью клавиши F7 создаем папку lab05 и перейдем в созданный каталог.

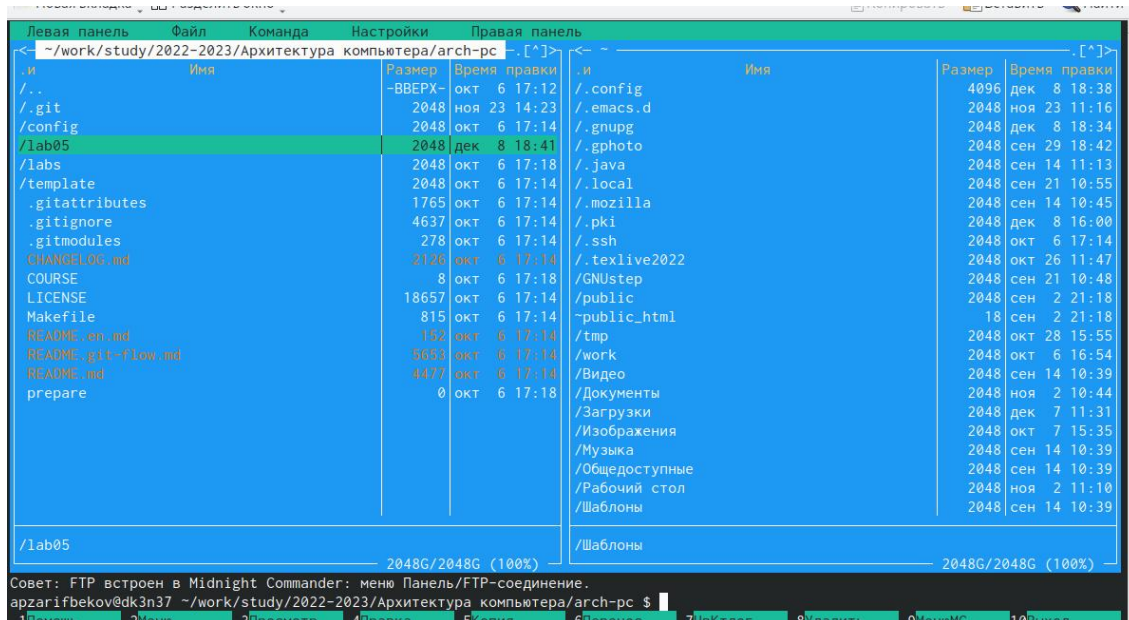


Рис. 3.1: 1.png

- 2) Пользуясь строкой ввода и командой touch создаем файл lab5-1.asm

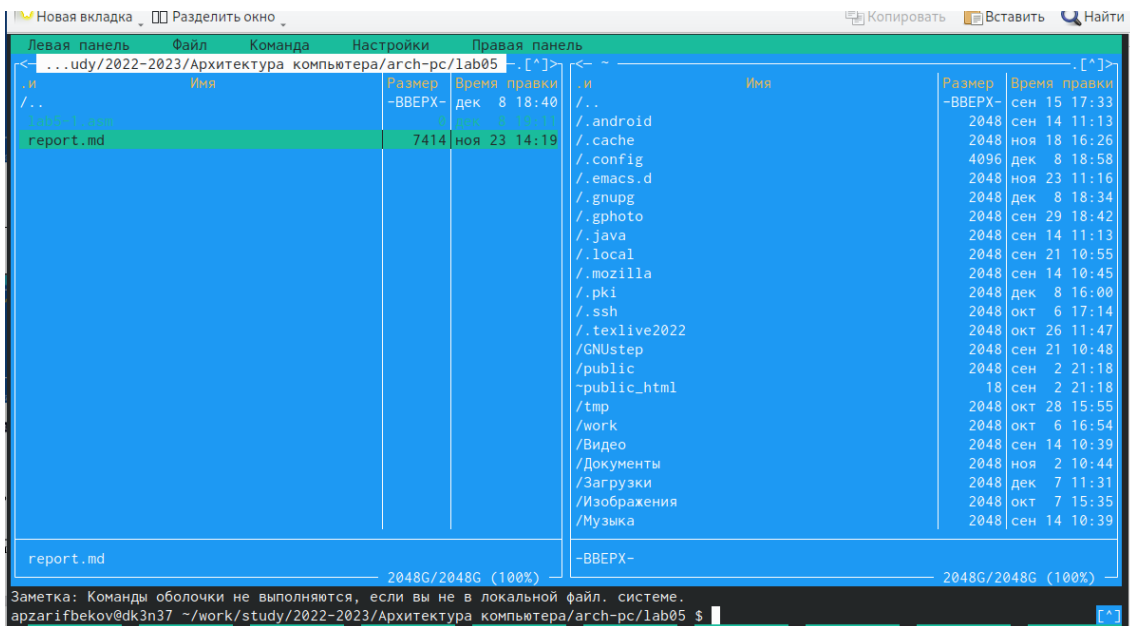


Рис. 3.2: 2.png

- 3) С помощью функциональной клавиши F4 откроем файл `lab5-1.asm` для редактирования во встроенном редакторе. Ввёл текст программы из листинга 6.1, сохранил изменения и закрыл файл.

```

lab5-1.asm          [----] 49 L:[ 1+ 2 3/ 35] *(237 /2431b) 0045 0x02D [ *
;
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;
;----- Объявление переменных -----
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку.',10 ; сообщение плюс
; символ перевода строки
msglen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не инициализированных данных
buf1: RESB 80 ; Буфер размером 80 байт
;----- Текст программы -----
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
;----- Системный вызов 'write' -----
; После вызова инструкции 'int 80h' на экран будет
; выведено сообщение из переменной 'msg' длиной 'msglen'
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла 1 - стандартный вывод
mov ecx,msg ; Адрес строки 'msg' в 'ecx'
mov edx,msglen ; Размер строки 'msg' в 'edx'
int 80h ; Вызов ядра
;----- системный вызов 'read' -----
; После вызова инструкции 'int 80h' программа будет ожидать ввода
; строки, которая будет записана в переменную 'buf1' размером 80 байт
mov eax,3 ; Системный вызов для чтения (sys_read)
mov ebx,0 ; Дескриптор файла 0 - стандартный ввод
mov ecx,buf1 ; Адрес буфера под вводимую строку
mov edx,80 ; Длина вводимой строки
int 80h ; Вызов ядра
;----- Системный вызов 'exit' -----
; После вызова инструкции 'int 80h' программа завершит работу
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра

```

Рис. 3.3: 3.png

- 4) С помощью функциональной клавиши F3 откроем файл lab6-1.asm для просмотра. Убедились, что файл содержит текст программы.



```

...pfu.edu.ru/home/a/p/apzarifbekov/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05/lab5-
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
;----- Объявление переменных -----
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не инициализированных данных
buf1: RESB 80 ; Буфер размером 80 байт
;----- Текст программы -----
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
;----- Системный вызов 'write' -----
; После вызова инструкции 'int 80h' на экран будет
; выведено сообщение из переменной 'msg' длиной 'msgLen'
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла 1 - стандартный вывод
mov ecx,msg ; Адрес строки 'msg' в 'ecx'
mov edx,msgLen ; Размер строки 'msg' в 'edx'
int 80h ; Вызов ядра
;----- системный вызов 'read' -----
; После вызова инструкции 'int 80h' программа будет ожидать ввода
; строки, которая будет записана в переменную 'buf1' размером 80 байт
mov eax,3 ; Системный вызов для чтения (sys_read)
mov ebx,0 ; Дескриптор файла 0 - стандартный ввод
mov ecx,buf1 ; Адрес буфера под вводимую строку
mov edx,80 ; Длина вводимой строки
int 80h ; Вызов ядра
;----- Системный вызов 'exit' -----
; После вызова инструкции 'int 80h' программа завершит работу
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра

```

Рис. 3.4: 4.png

- 5) Оттранслировали текст программы lab5-1.asm в объектный файл. Выполняли компоновку объектного файла и запустил получившийся исполняемый файл.

```

apzarifbekov@dk3n37 ~ $ mc

apzarifbekov@dk3n37 ~ $ touch lab5-1

apzarifbekov@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ lab05-1
bash: lab05-1: команда не найдена

apzarifbekov@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ touch lab5-1

apzarifbekov@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ touch lab5-1.asm

apzarifbekov@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ nasm -f elf lab5-1.asm
apzarifbekov@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ d -m elf_i386 -o lab5-1.o lab5-1.asm
bash: d: команда не найдена

apzarifbekov@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ./lab5-1
bash: ./lab5-1: Нет такого файла или каталога

apzarifbekov@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ./lab5-1
bash: ./lab5-1: Нет такого файла или каталога

apzarifbekov@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1.o lab5-1.o

apzarifbekov@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ./lab5-1
Введите строку:
Зарифбеков Амир Пайшанбиевич █

```

Рис. 3.5: 5.png

6) Скачал файл in\_out.asm. Скопировал файл в каталог с файлом lab5-1.asm

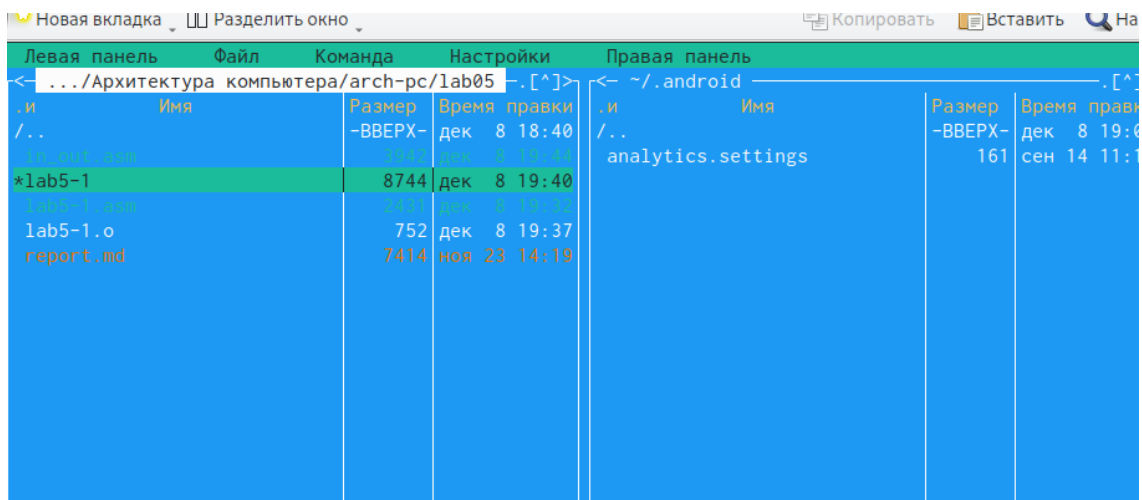


Рис. 3.6: 6.png

7) Создаем копию файла lab5-1.asm с именем lab5-2.asm

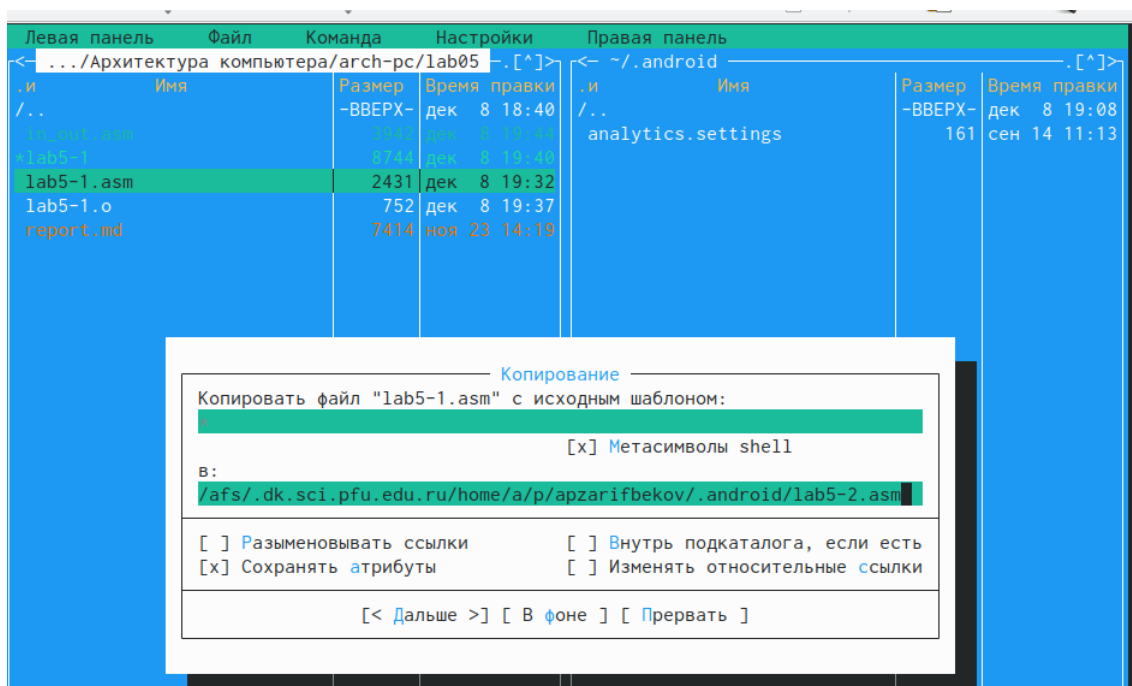


Рис. 3.7: 7.png

8) Исправил текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in\_out.asm В файле lab5-2.asm заменим подпрограмму sprintLF на sprint. Создаем исполняемый файл и проверили его

работу. Разница в выводе текста.

Рис. 3.8: 8.png

Рис. 3.9: 9.png

## 4 Выполнение самостоятельной работы

- 1) Создаем копию файла lab5-1.asm. Внесем изменения в программу. Получим исполняемый файл и проверим его работу. На приглашение ввести строку введем свою фамилию. Создаем копию файла lab5-2.asm. Исправим текст программы с использованием подпрограмм из внешнего файла in\_out.asm. Создаем исполняемый файл и проверим его работу.

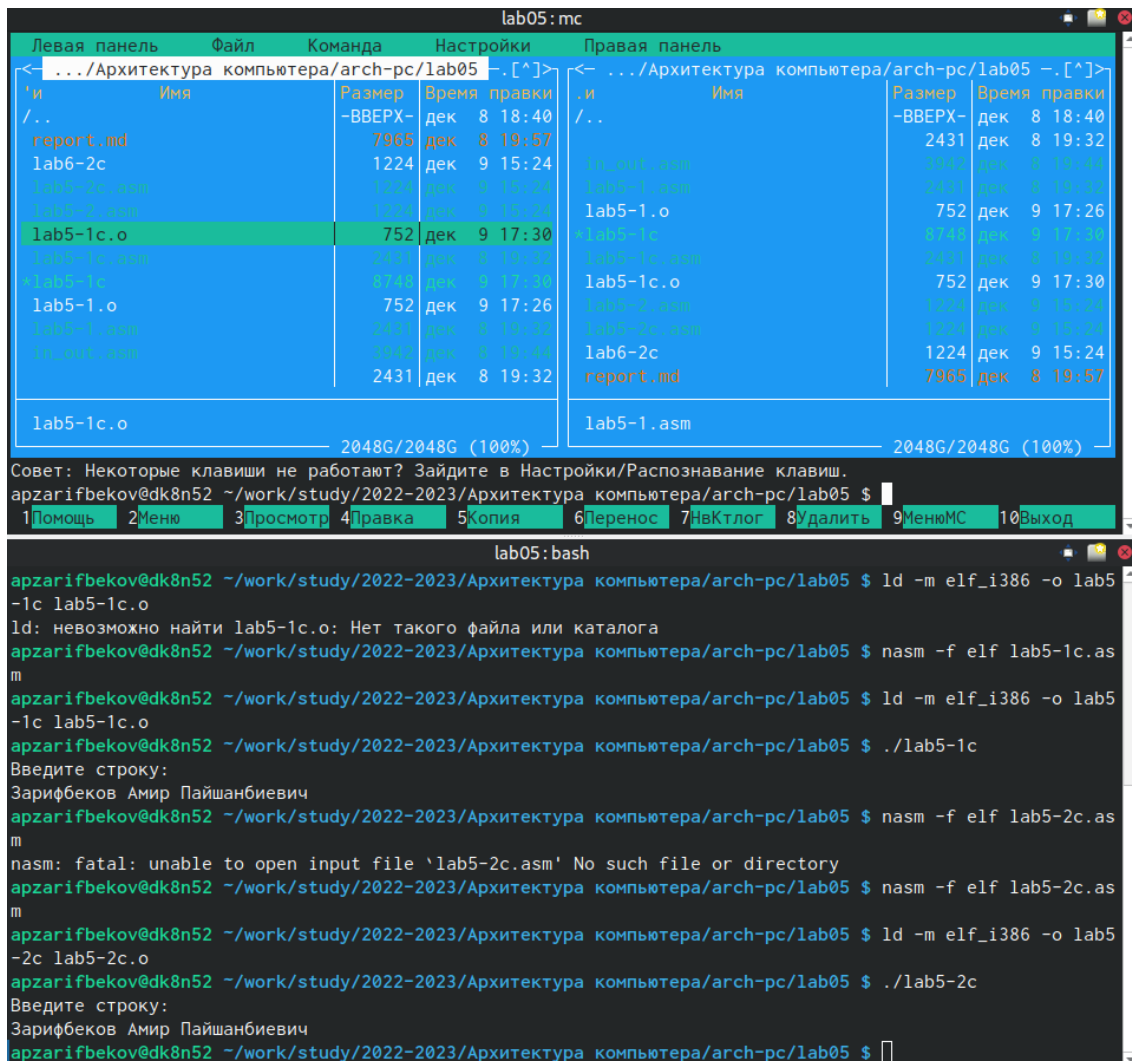


Рис. 4.1: 10.png

## 5 Выводы

Я приобрели практические навыки работы в Midnight Commander. И освоили инструкции языка ассемблера `mov` и `int`.