

# **Отчёт по лабораторной работе №5**

**Архитектура вычислительных систем**

Валиева Марина Русланбековна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Выполнение самостоятельной работы</b>	<b>13</b>
<b>4</b>	<b>Выводы</b>	<b>14</b>

## Список иллюстраций

2.1	открытие mc . . . . .	6
2.2	создание папки lab5 и переход в созданный каталог . . . . .	8
2.3	создание файла lab5-1.asm . . . . .	8
2.4	открытие файла для редактирования во встроенном редакторе . .	8
2.5	введение текст программы из листинга 6.1 . . . . .	9
2.6	оттранслирование текста, выполнение компоновки объектного файла и запуск получившейся исполняемый файл . . . . .	9
2.7	выполнение этих действий . . . . .	10
2.8	замена подпрограммы . . . . .	12
3.1	Выполнение . . . . .	13

## Список таблиц

# 1 Цель работы

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

## 2 Выполнение лабораторной работы

1. Откроем Midnight Commander



Рис. 2.1: открытие mc

2. Пользуясь клавишами **⌘**, **⌘** и Enter перейдите в каталог ~/work/arch-rc созданный при выполнении лабораторной работы No5.



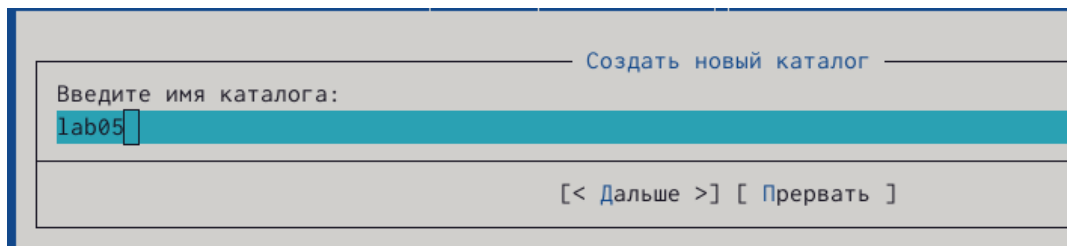


Рис. 2.2: создание папки lab5 и переход в созданный каталог

4. Пользуясь строкой ввода и командой touch создадим файл lab5-1.asm.

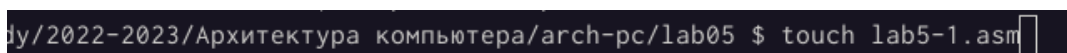


Рис. 2.3: создание файла lab5-1.asm

5. С помощью функциональной клавиши F4 откройте файл lab6-1.asm для редактирования во встроенном редакторе. Как правило в качестве встроенного редактора Midnight Commander используется редакторы nano или mcedit

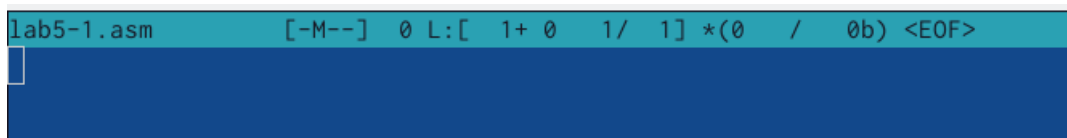


Рис. 2.4: открытие файла для редактирования во встроенном редакторе

6. Введем текст программы из листинга 6.1 , сохраните изменения и закройте файл.



```
lab5-1.asm [BM--] 0 L:[ 1+21 22/ 38] *(1445/2572b) 0109 0x06D
Листинг 6.1. Программа вывода сообщения на экран и ввода строки с
клавиатуры
;-----
; 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'
```

Рис. 2.5: введение текст программы из листинга 6.1

7. С помощью функциональной клавиши F3 откроем файл lab6-1.asm для просмотра. Убедимся, что файл содержит текст программы.
8. Оттранслируем текст программы lab5-1.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл. Программа выводит строку 'Введите строку:' и ожидает ввода с клавиатуры. На запрос введем наши данные.

```
mrvalieva@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ nasm -f elf lab5-1.asm
mrvalieva@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o
mrvalieva@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ./lab5-1
Введите строку:
Валиева Марина
mrvalieva@dk3n35 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $
```

Рис. 2.6: оттранслирование текста, выполнение компоновки объектного файла и запуск получившейся исполняемый файл

9. Скачаем файл in\_out.asm со страницы курса в ТУИС.

10. Подключаемый файл `in_out.asm` должен лежать в том же каталоге, что и файл с программой, в которой он используется.
11. С помощью функциональной клавиши F6 создадим копию файла `lab6-1.asm` с именем `lab6-2.asm`. Выделим файл `lab6-1.asm`, нажмем клавишу F6, введем имя файла `lab6-2.asm` и нажмем клавишу Enter.

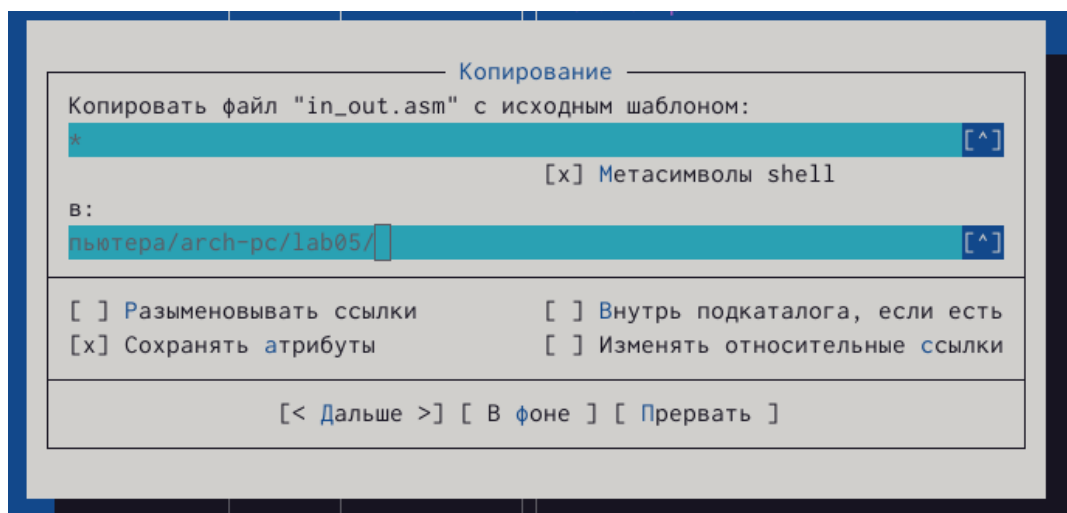
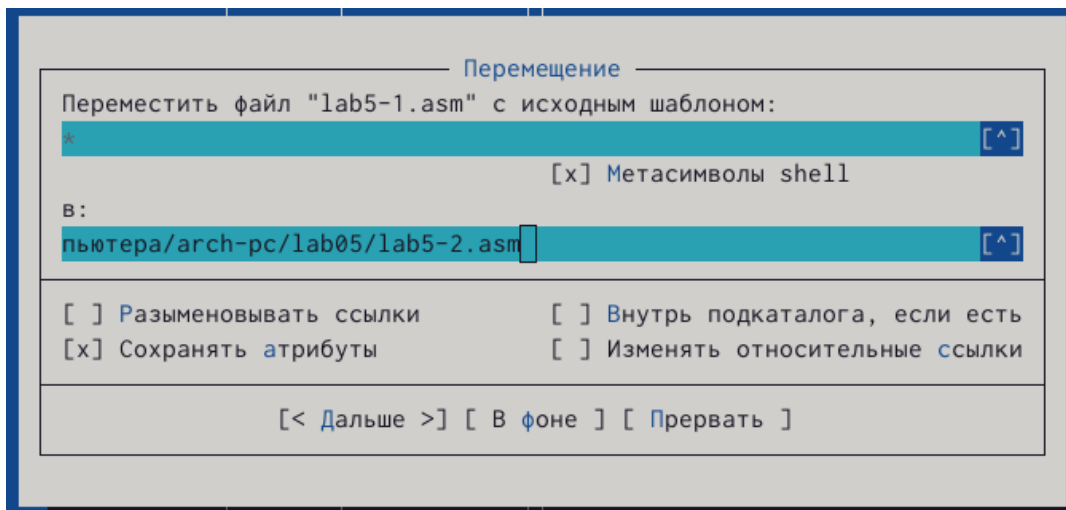


Рис. 2.7: выполнение этих действий

12. Исправим текст программы в файле `lab6-2.asm` с использованием подпрограмм из внешнего файла `in_out.asm` (используем подпрограммы `sprintf`, `sread` и `quit`) в соответствии с листингом 6.2. Создадим исполняемый файл и проверим его работу.

in_out.asm	3942	ноя 24 12:22
*lab5-1	8744	дек 7 11:24
lab5-1.asm	2434	дек 7 11:24
lab5-1.o	752	дек 7 11:24



Левая панель	Файл	Команда	Наст
<-	... компьютера/arch-pc/lab05	-.[^]>	<-
.и	Имя	Размер	Время правки
/..		-ВВЕРХ-	ноя 24 12:33
in_out.asm		3942	ноя 24 12:22
*lab5-1		8744	дек 7 11:24
lab5-1.o		752	дек 7 11:24
lab5-2.asm		2434	дек 7 11:24

- В файле lab6-2.asm заменим подпрограмму `sprintLF` на `sprint`. Создадим исполняемый файл и проверим его работу.

```

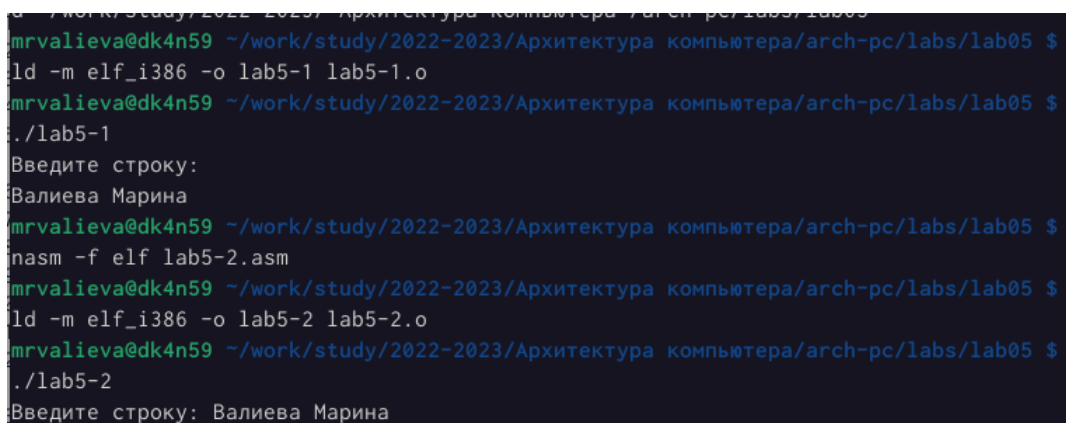
GNU nano 6.3 /afs/.dk.sci.p
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция инициированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не инициированных данных
buf1: RESB 80 ; Буфер размером 80 байт
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
mov eax, msg ; запись адреса выводимого сообщения в `EAX`
call sprint ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса переменной в `EAX`
mov edx, 80 ; запись длины вводимого сообщения в `EBX`
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения

```

Рис. 2.8: замена подпрограммы

### 3 Выполнение самостоятельной работы

1. Создадим копию файла lab6-1.asm. Внесем изменения в программу (без использования внешнего файла in\_out.asm), так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введенную строку на экран.
2. Получим исполняемый файл и проверим его работу. На приглашение ввести строку введите свою фамилию.
3. Создадим копию файла lab6-2.asm. Исправим текст программы с использование подпрограмм из внешнего файла in\_out.asm, так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введенную строку на экран.
4. Создадим исполняемый файл и проверьте его работу.



```
mrvalieva@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $  
ld -m elf_i386 -o lab5-1 lab5-1.o  
mrvalieva@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $  
./lab5-1  
Введите строку:  
Валиева Марина  
mrvalieva@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $  
nasm -f elf lab5-2.asm  
mrvalieva@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $  
ld -m elf_i386 -o lab5-2 lab5-2.o  
mrvalieva@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $  
./lab5-2  
Введите строку: Валиева Марина
```

Рис. 3.1: Выполнение

## 4 Выводы

В ходе лабораторной работы №5 мы приобрели практические навыки работы в Midnight Commander и освоили инструкции языка ассемблера `mov` и `int`.