

Отчет по лабораторной работе №5

Воинов Кирилл

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Задание для самостоятельной работы	13
4	Выводы	15

Список иллюстраций

2.1	Открытие Midnight Commander	5
2.2	Переход в каталог ~/work/arch-pc	6
2.3	Создание папки lab05	6
2.4	Переход в созданный каталог и создание файла lab5-1.asm	7
2.5	Открытие файла lab5-1.asm	7
2.6	Ввод текста программы из листинга 5.1 1	8
2.7	Ввод текста программы из листинга 5.1 2	8
2.8	Оттранслирование, компоновка, запуск получившегося исполняе- мого файла	9
2.9	Скаченный файл	9
2.10	Копирование файла в нужный каталог	10
2.11	Создание копии файла lab5-1.asm с именем lab5-2.asm	10
2.12	Исправление текста файла lab5-2.asm	11
2.13	Создание исполняемого файла и проверка его работы	11
2.14	Результат замены в lab5-2.asm	12
3.1	lab5-1_1.asm	13
3.2	lab5-1_1.asm результат	14
3.3	lab5-2_1.asm	14
3.4	lab5-2_1.asm результат	14

1 Цель работы

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

2 Выполнение лабораторной работы

1. Открываю Midnight Commander.(рис. 2.1).

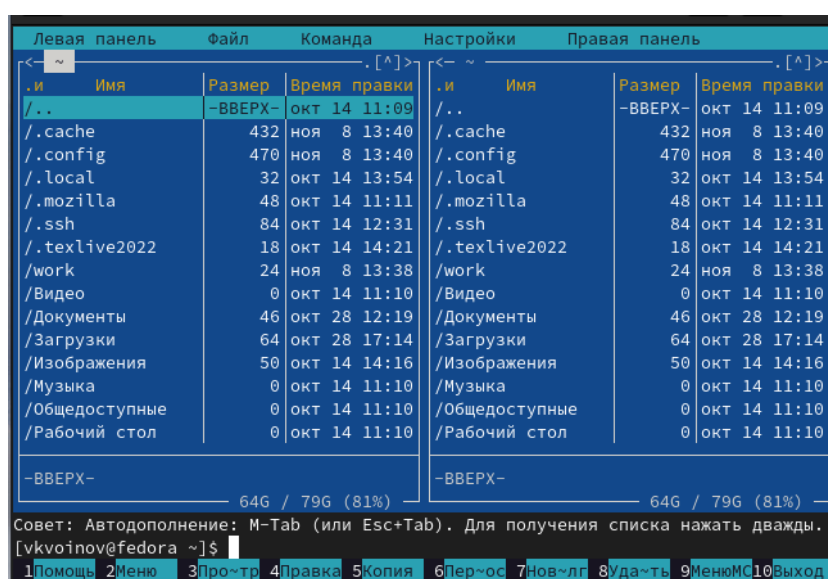


Рис. 2.1: Открытие Midnight Commander

2. Перехожу в каталог ~/work/arch-рс созданный при выполнении лабораторной работы No4.(рис. 2.2).

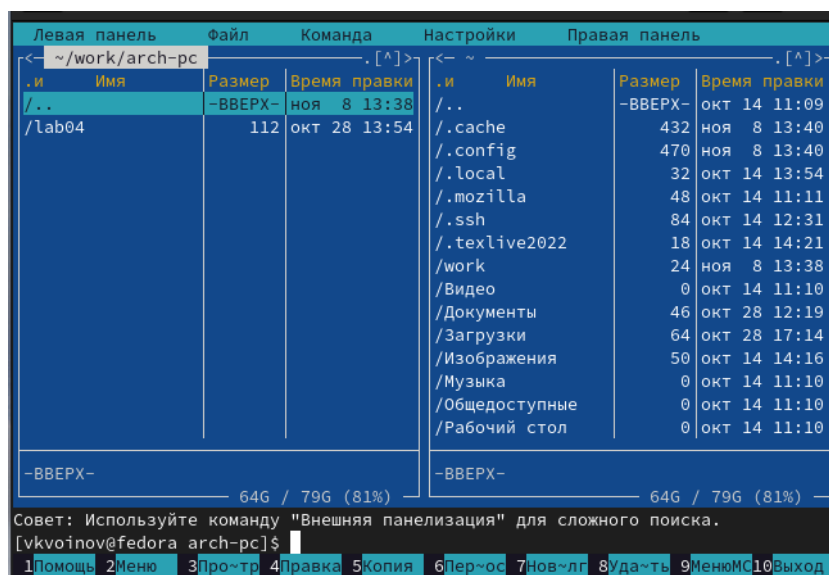


Рис. 2.2: Переход в каталог ~/work/arch-pc

- С помощью функциональной клавиши F7 создаю папку lab05 и перехожу в созданный каталог.(рис. 2.3) и (рис. 2.4)

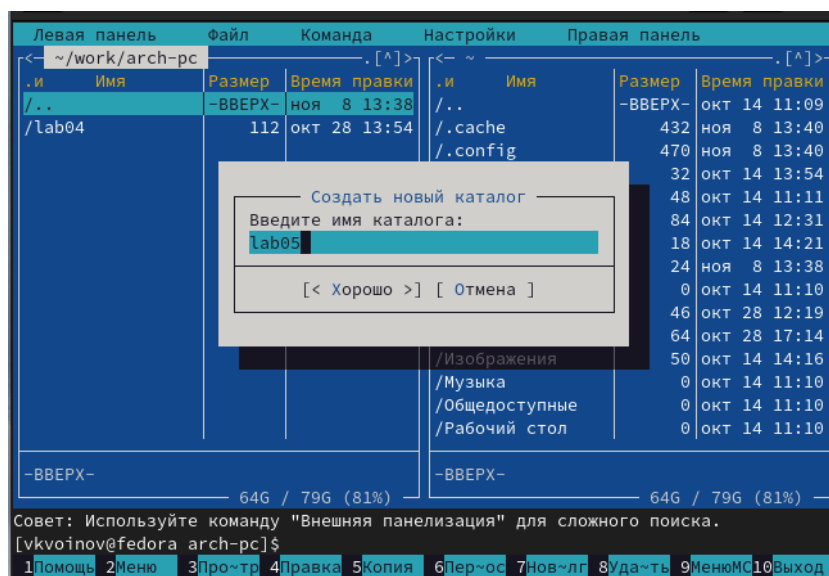
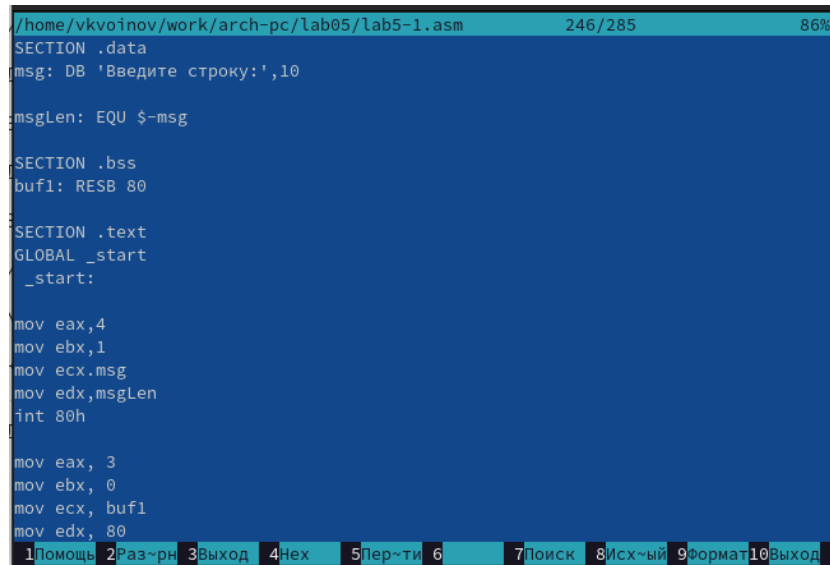


Рис. 2.3: Создание папки lab05

- Пользуясь строкой ввода и командой touch создаю файл lab5-1.asm.(рис. 2.4)

7. С помощью функциональной клавиши F3 открываю файл lab5-1.asm для просмотра. Убеждаюсь, что файл содержит текст программы. (рис. 2.6) и (рис. 2.7)



```
/home/vkvoinov/work/arch-pc/lab05/lab5-1.asm 246/285 86%
SECTION .data
msg: DB 'Введите строку:',10

msgLen: EQU $-msg

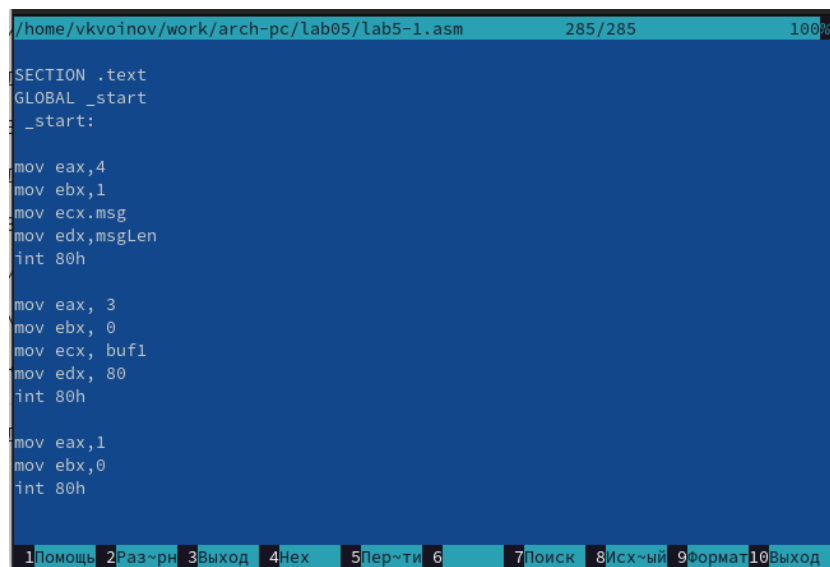
SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h

mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
```

Рис. 2.6: Ввод текста программы из листинга 5.1 1



```
/home/vkvoinov/work/arch-pc/lab05/lab5-1.asm 285/285 100%
SECTION .text
GLOBAL _start
_start:

mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h

mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h

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

Рис. 2.7: Ввод текста программы из листинга 5.1 2

8. Оттранслирую текст программы lab5-1.asm в объектный файл. Выполню компоновку объектного файла и запускаю получившийся исполняемый файл. (рис. 2.8)

```
[vkvoinov@fedora lab05]$ nasm -f elf lab5-1.asm
[vkvoinov@fedora lab05]$ ld -m elf_i386 lab5-1.o -o lab5-1
[vkvoinov@fedora lab05]$ ./lab5-1
Введите строку:
Воинов Кирилл Викторович
[vkvoinov@fedora lab05]$
```

Рис. 2.8: Оттранслирование, компоновка, запуск получившегося исполняемого файла

9. Скачиваю файл in_out.asm со страницы курса в ТУИС.(рис. 2.9)

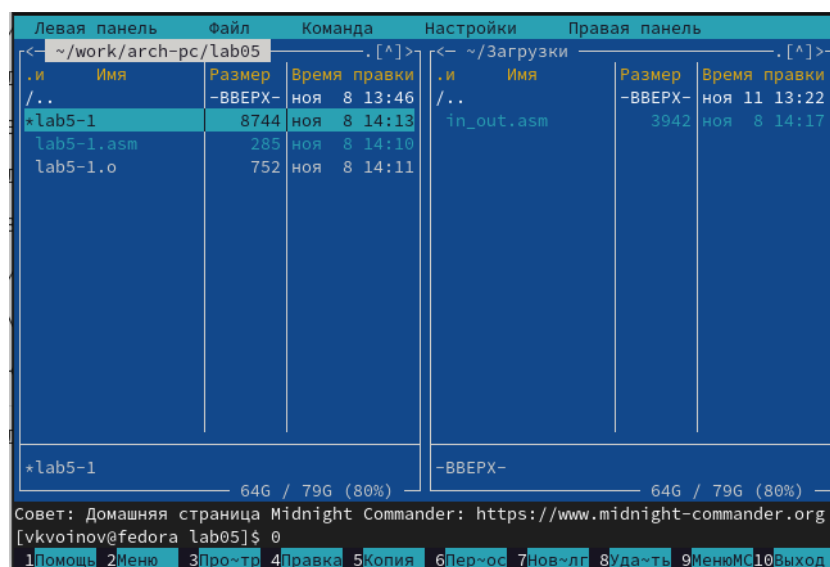


Рис. 2.9: Скаченный файл

10. Копирую файл in_out.asm в тот же каталог, что и файл с программой, в которой он используется.(рис. 2.10)

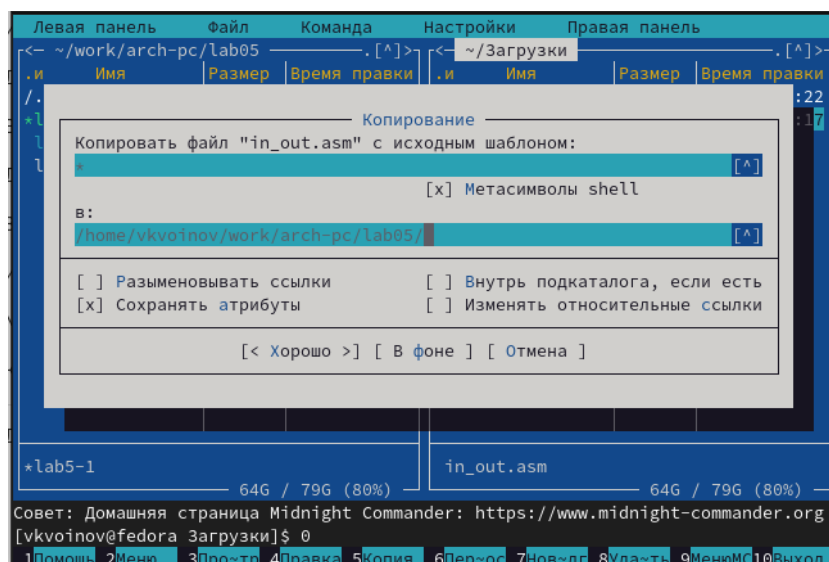


Рис. 2.10: Копирование файла в нужный каталог

11. С помощью функциональной клавиши F6 создаю копию файла lab5-1.asm с именем lab5-2.asm. Выделяю файл lab5-1.asm, нажимаю клавишу F6, ввожу имя файла lab5-2.asm и нажимаю клавишу Enter.(рис. 2.11)

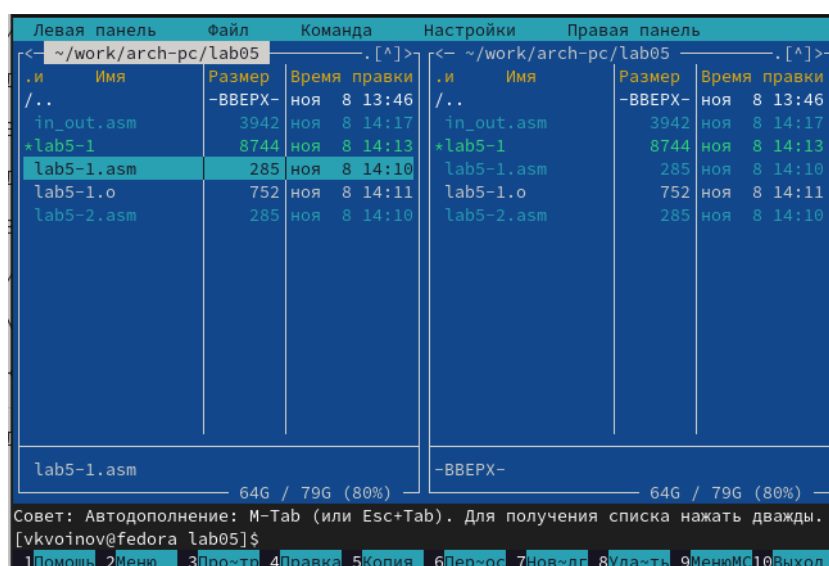
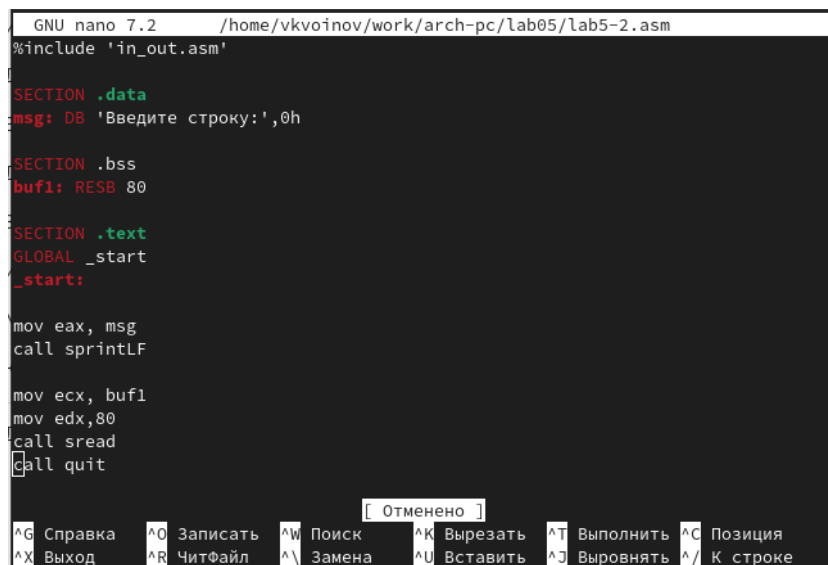


Рис. 2.11: Создание копии файла lab5-1.asm с именем lab5-2.asm

12. Исправляю текст программы в файле lab5-2.asm с использованием подпро-

грамм из внешнего файла in_out.asm в соответствии с листингом 5.2. Создаю исполняемый файл и проверяю его работу.(рис. 2.12) и (рис. 2.13)



```

GNU nano 7.2 /home/vkvoinov/work/arch-pc/lab05/lab5-2.asm
#include 'in_out.asm'

SECTION .data
msg: DB 'Введите строку:',0h

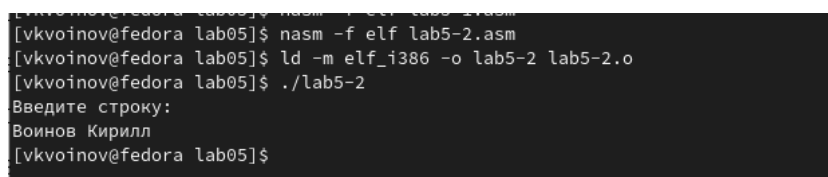
SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax, msg
call sprintLF

mov ecx, buf1
mov edx, 80
call sread
call quit
  
```

Рис. 2.12: Исправление текста файла lab5-2.asm



```

[vkvoinov@fedora lab05]$ nasm -f elf lab5-2.asm
[vkvoinov@fedora lab05]$ ld -m elf_i386 -o lab5-2 lab5-2.o
[vkvoinov@fedora lab05]$ ./lab5-2
Введите строку:
Воинов Кирилл
[vkvoinov@fedora lab05]$
  
```

Рис. 2.13: Создание исполняемого файла и проверка его работы

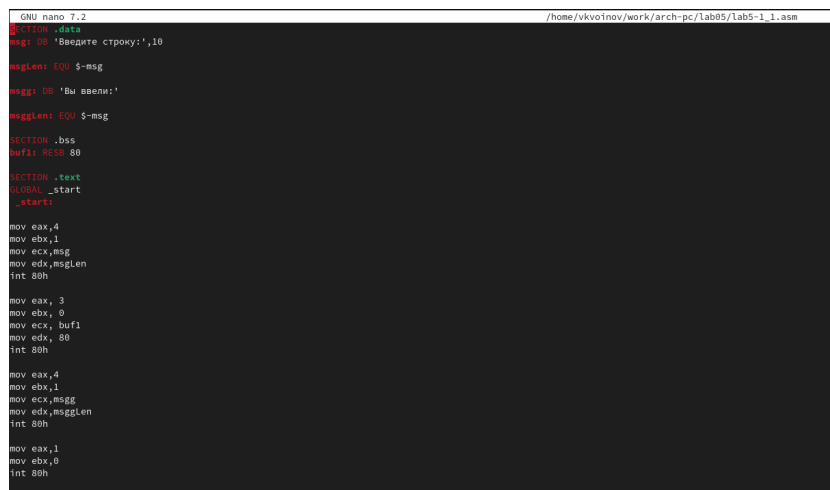
13. В файле lab5-2.asm заменяю подпрограмму sprintLF на sprint. Создаю исполняемый файл и проверяю его работу. Разница в том, что sprintLF работает аналогично sprint, но при выводе на экран добавляет к сообщению символ перевода строки. То есть в первом случае ввод происходил на следующей строке после 'Введите строку:' а во втором - на этой же строке.(рис. 2.14)

```
[vkvoinov@fedora lab05]$ nasm -f elf lab5-2.asm
[vkvoinov@fedora lab05]$ ld -m elf_i386 -o lab5-2 lab5-2.o
[vkvoinov@fedora lab05]$ ./lab5-2
Введите строку:Воинов Кирилл
[vkvoinov@fedora lab05]$
```

Рис. 2.14: Результат замены в lab5-2.asm

3 Задание для самостоятельной работы

1. Создаю копию файла lab5-1.asm с названием lab5-1_1.asm. Вношу изменения в программу.(рис. 3.1)



```
GNU nano 7.2 /home/vkvoinov/work/arch-pc/lab85/lab5-1_1.asm
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
msgg: DB 'Вы ввели:'
msggLen: EQU $-msgg
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h
mov eax,3
mov ebx,0
mov ecx,buf1
mov edx,80
int 80h
mov eax,4
mov ebx,1
mov ecx,msgg
mov edx,msggLen
int 80h
mov eax,1
mov ebx,0
int 80h
```

Рис. 3.1: lab5-1_1.asm

2. Получаю исполняемый файл и проверяю его работу. На приглашение ввожу свою фамилию.(рис. 3.2)

```
[vkvoinov@fedora lab05]$ ./lab5-1_1
Введите строку:
Воинов
Вы ввели:Воинов
[vkvoinov@fedora lab05]$
```

Рис. 3.2: lab5-1_1.asm результат

3. Создаю копию файла lab5-2.asm. Исправляю текст программы с использование подпрограмм из внешнего файла in_out.asm.(рис. 3.3)

```
GNU nano 7.2 /home/vkvoinov/work/arch-pc/lab05/lab5-2_1.asm
#include "in_out.asm"

SECTION .data
msg: db "Введите строку:",0h
msgg: db "Вы ввели:"
SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax, msg
call sprint

mov ecx, buf1
mov edx, 80
call sread

mov eax, msgg
call sprint
mov eax, buf1
call sprint

call quit
```

Рис. 3.3: lab5-2_1.asm

4. Создаю исполняемый файл и проверяю его работу.(рис. 3.4)

```
[vkvoinov@fedora lab05]$ nasm -f elf lab5-2_1.asm
[vkvoinov@fedora lab05]$ ld -m elf_i386 -o lab5-2_1 lab5-2_1.o
[vkvoinov@fedora lab05]$ ./lab5-2_1
Введите строку:Воинов
Вы ввели:Воинов
[vkvoinov@fedora lab05]$
```

Рис. 3.4: lab5-2_1.asm результат

4 Выводы

На этой лабораторной работе я приобрел практические навыки работы в Midnight Commander, освоил инструкции языка ассемблера `mov` и `int`.