

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5**

дисциплина: Архитектура компьютера

Студент: Татур Платон Андреевич

Группа: НКАбд-05-25

**МОСКВА**

2025 г.

## **5.1 Цель работы:**

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

## 5.3. Порядок выполнения лабораторной работы:

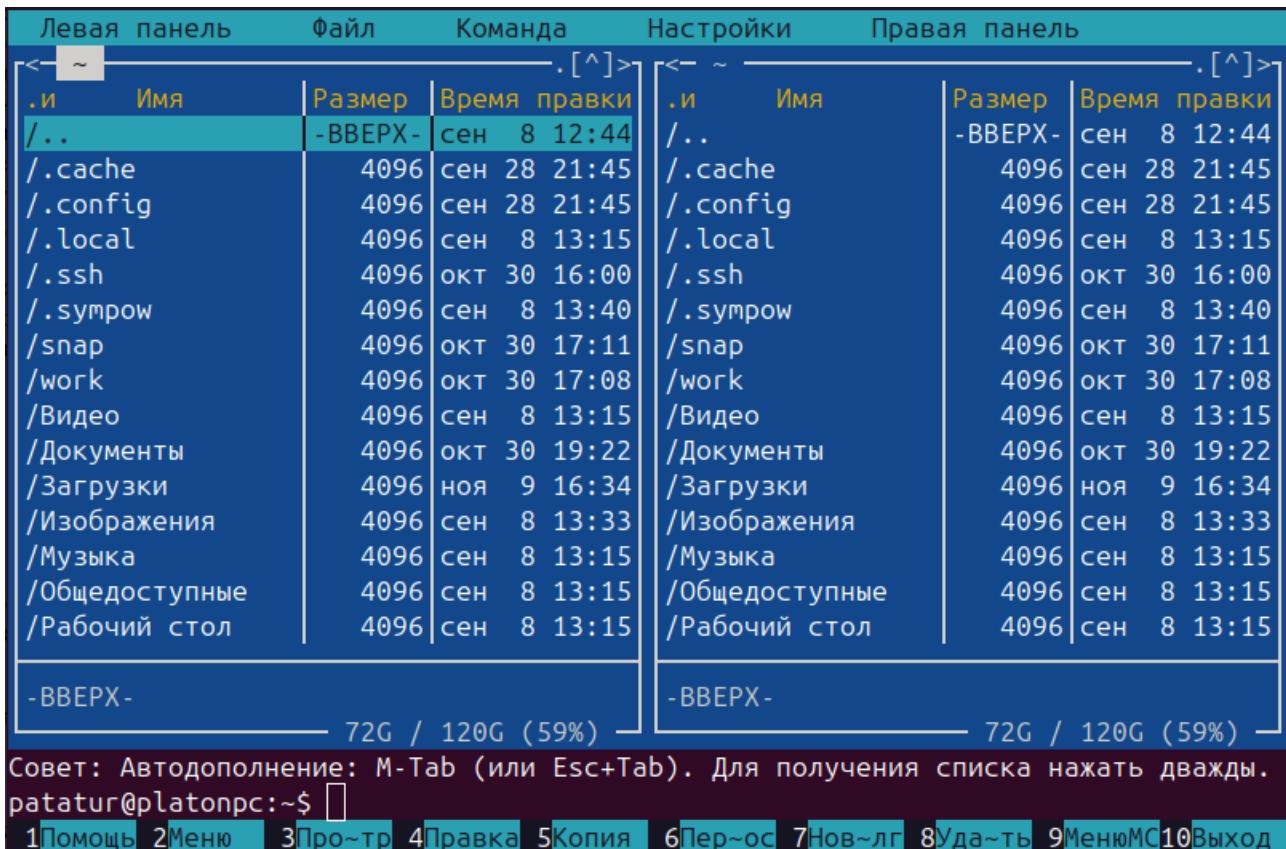


Рис. 1

1) Открыл Midnight Commander с помощью команды mc.(см Рис. 1)

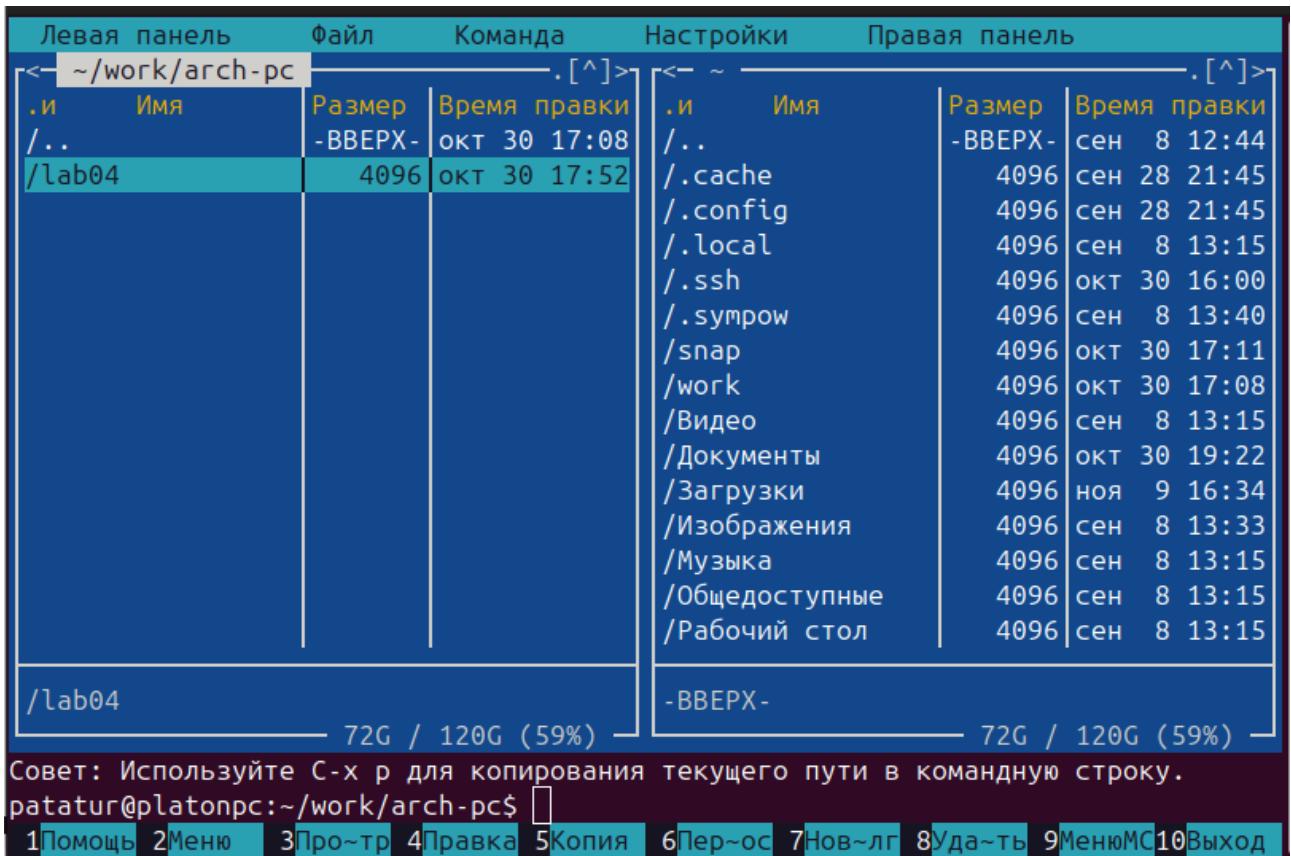


Рис. 2

2) Пользуясь клавишами `↑`, `↓` и `Enter` прешёл в каталог `~/work/arch-pc` созданный при выполнении лабораторной работы №4 (см Рис. 2)

Левая панель		Файл	Команда	Настройки	Правая панель		
.и	Имя	Размер	Время правки	.и	Имя	Размер	Время правки
/..		-ВВЕРХ-	ноя 9 16:51	/..		-ВВЕРХ-	сен 8 12:44
				/.cache		4096	сен 28 21:45
				/.config		4096	сен 28 21:45
				/.local		4096	сен 8 13:15
				/.ssh		4096	окт 30 16:00
				/.sympow		4096	сен 8 13:40
				/.snap		4096	окт 30 17:11
				/.work		4096	окт 30 17:08
				/Видео		4096	сен 8 13:15
				/Документы		4096	окт 30 19:22
				/Загрузки		4096	ноя 9 16:34
				/Изображения		4096	сен 8 13:33
				/Музыка		4096	сен 8 13:15
				/Общедоступные		4096	сен 8 13:15
				/Рабочий стол		4096	сен 8 13:15

-ВВЕРХ- 72G / 120G (59%) - ВВЕРХ- 72G / 120G (59%)

Совет: Вы можете отменить запросы на подтверждение в Настройки/Подтверждение.

pataturg@platonpc:~/work/arch-pc/lab05\$

1Помощь 2Меню 3Про~тр 4Правка 5Копия 6Пер~ос 7Нов~лг 8Уда~ть 9МенюМС 10Выход

Рис. 3

3) С помощью функциональной клавиши F7 создал папку lab05 и перешёл в созданный каталог. (см Рис. 3)

Левая панель		Файл	Команда	Настройки	Правая панель					
<-	~/work/arch-pc/lab05	..	[^]>	<- ~	..	[^]>				
.	Имя	Размер	Время правки	.	Имя	Размер				
/..		-BBERX-	ноя 9 16:51	/..		-BBERX-				
				/.cache	4096	сен 28 21:45				
				/.config	4096	сен 28 21:45				
				/.local	4096	сен 8 13:15				
				/.ssh	4096	окт 30 16:00				
				/.sympow	4096	сен 8 13:40				
				/.snap	4096	окт 30 17:11				
				/.work	4096	окт 30 17:08				
				/Видео	4096	сен 8 13:15				
				/Документы	4096	окт 30 19:22				
				/Загрузки	4096	ноя 9 16:34				
				/Изображения	4096	сен 8 13:33				
				/Музыка	4096	сен 8 13:15				
				/Общедоступные	4096	сен 8 13:15				
				/Рабочий стол	4096	сен 8 13:15				
-BBERX-		72G / 120G (59%)		-BBERX-		72G / 120G (59%)				
Совет: Вы можете отменить запросы на подтверждение в Настройки/Подтверждение.										
patatur@platoppc:~/work/arch-pc/lab05\$ touch lab5-1.asm										
1Помощь	2Меню	3Про~тр	4Правка	5Копия	6Пер~ос	7Нов~лг				
8Уда~ть	9МенюMS	10Выход								

Puc. 4.1

Puc. 4.2

4. Пользуясь строкой ввода и командой touch создал файл lab5-1.asm (см Рис. 4.1 и Рис. 4.2)

GNU nano 7.2 /home/patatur/work/arch-pc/lab05/lab5-1.asm

[ Прочитано 0 строк ]

**^G Справка** **^O Записать** **^W Поиск** **^K Вырезать** **^T Выполнить** **^C Позиция**  
**^X Выход** **^R ЧитФайл** **^V Замена** **^U Вставить** **^J Выровнять** **^/ К строке**

Рис. 5

5. С помощью функциональной клавиши F4 открыл файл lab5-1.asm для редактирования в редакторе nano. (см Рис. 5)

GNU nano 7.2 /home/patatur/work/arch-pc/lab05/lab5-1.asm \*

```
;-----  
; 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)
```

**^G Справка** **^O Записать** **^W Поиск** **^K Вырезать** **^T Выполнить** **^C Позиция**  
**^X Выход** **^R ЧитФайл** **^V Замена** **^U Вставить** **^J Выровнять** **^/ К строке**

Рис 6.1

```
GNU nano 7.2          /home/patatur/work/arch-pc/lab05/lab5-1.asm *
; 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

^G Справка ^O Записать ^W Поиск      ^K Вырезать ^T Выполнить ^C Позиция
^X Выход   ^R ЧитФайл ^\ Замена      ^U Вставить ^J Выровнять ^/ К строке
```

Рис. 6.2

6) Ввёл текст программы из листинга 5.1, сохранил изменения (*Ctrl + x (выход) > Y (сохранить изменения) > Enter*) и закрыл файл.

```
/home/patatur/work/arch-pc/lab05/lab5-1.asm          1421/2516      56%
;-----;
; 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'
1Помощь 2Раз~рн 3Выход 4Нех 5Пер~ти 6 7Поиск 8Исх~ый 9Формат10Выход
```

Рис. 7

7) С помощью функциональной клавиши F3 открыл файл lab5-1.asm для просмотра.

Убедился, что файл содержит текст программы.

```
patatur@platonpc:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
patatur@platonpc:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
patatur@platonpc:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Татур Платон Андреевич
patatur@platonpc:~/work/arch-pc/lab05$ █
```

Рис. 8

- 8) 1. Скрыл панели Midnight Commander, с помощью комбинации Ctrl + o.
2. Оттранслировал текст программы lab5-1.asm в объектный файл.
3. Выполнил компоновку объектного файла и запустил получившийся исполняемый файл.
4. Программа выводит строку 'Введите строку:' и ожидает ввода с клавиатуры. На запрос ввел свои ФИО.

(см Рис. 8)

### 5.3.1. Подключение внешнего файла in\_out.asm

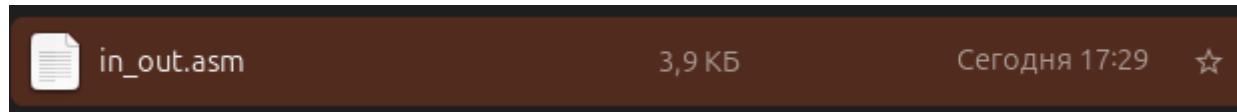


Рис. 9

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

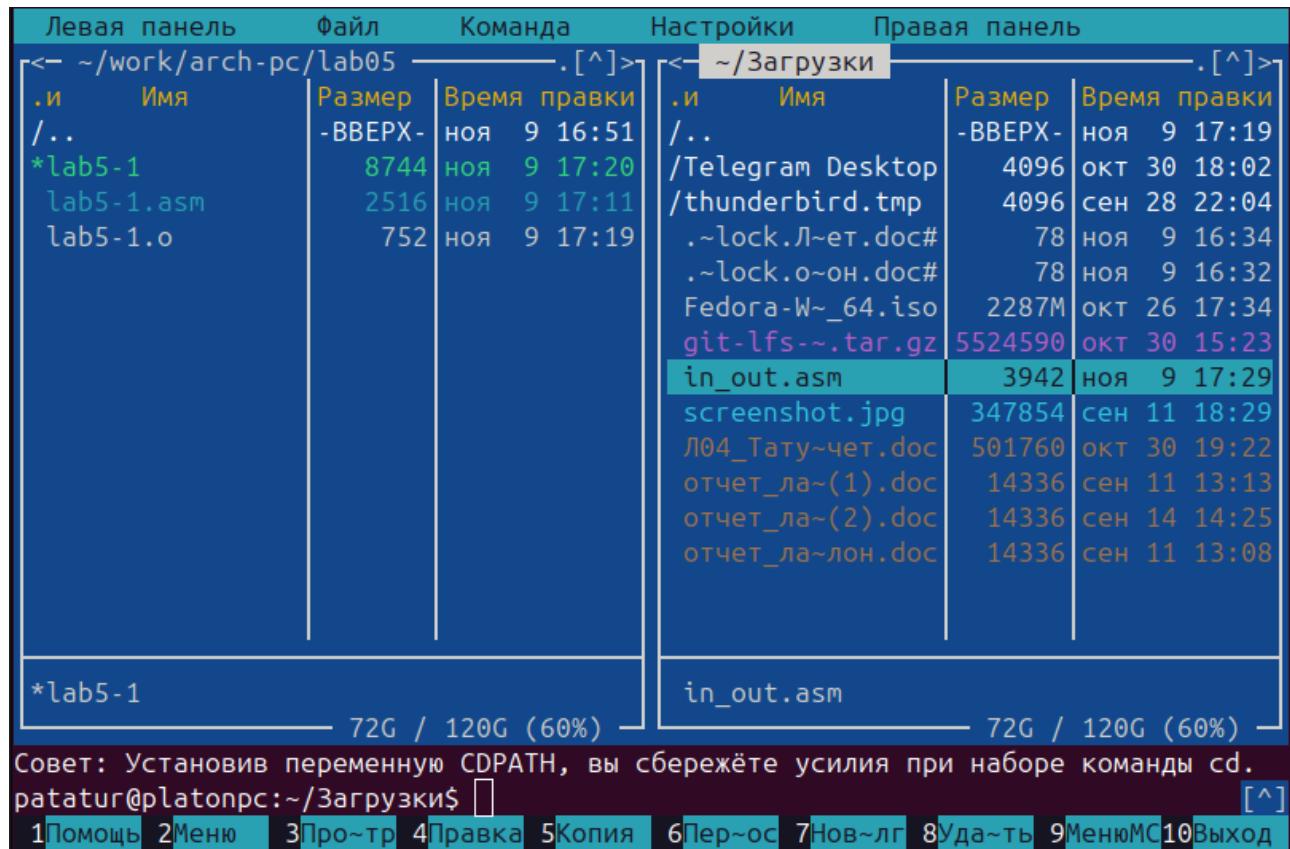


Рис. 10.1

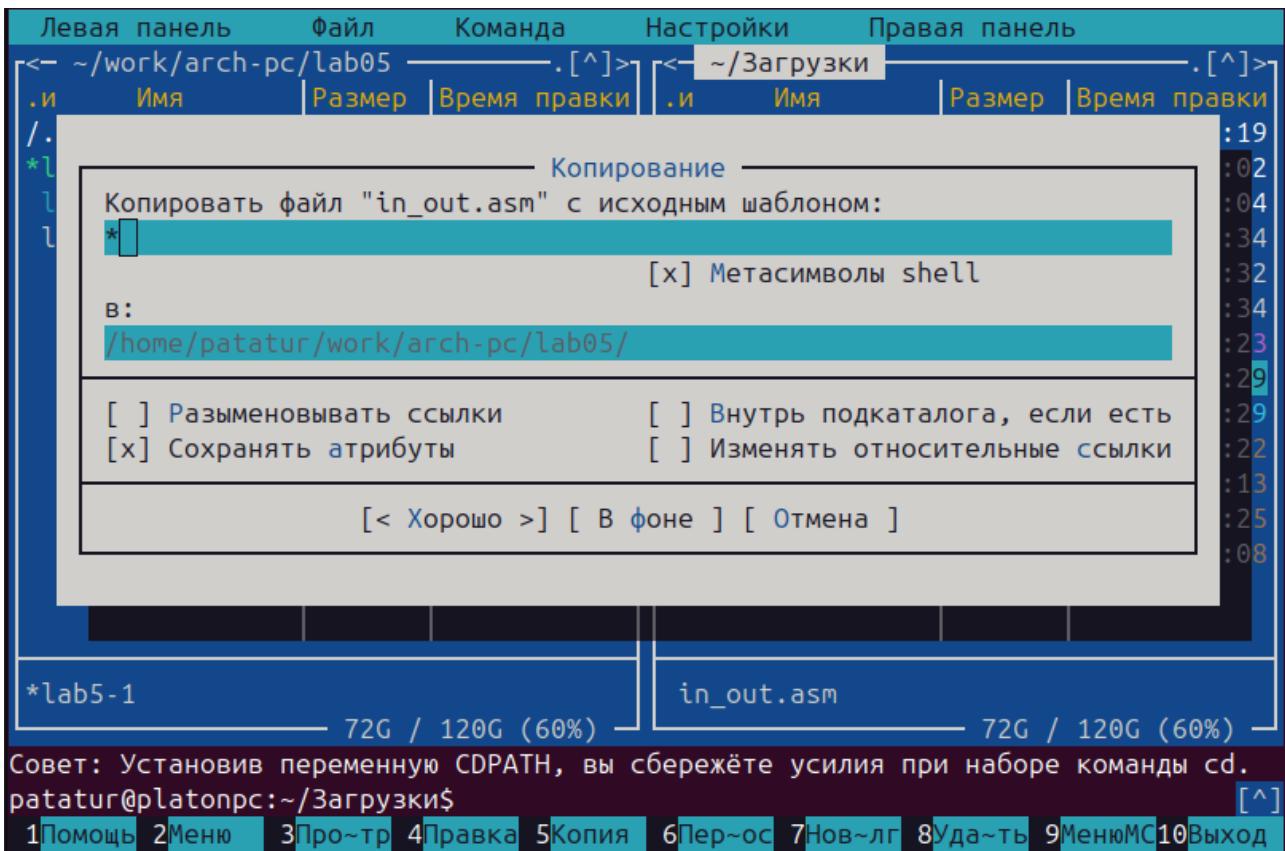


Рис. 10.2

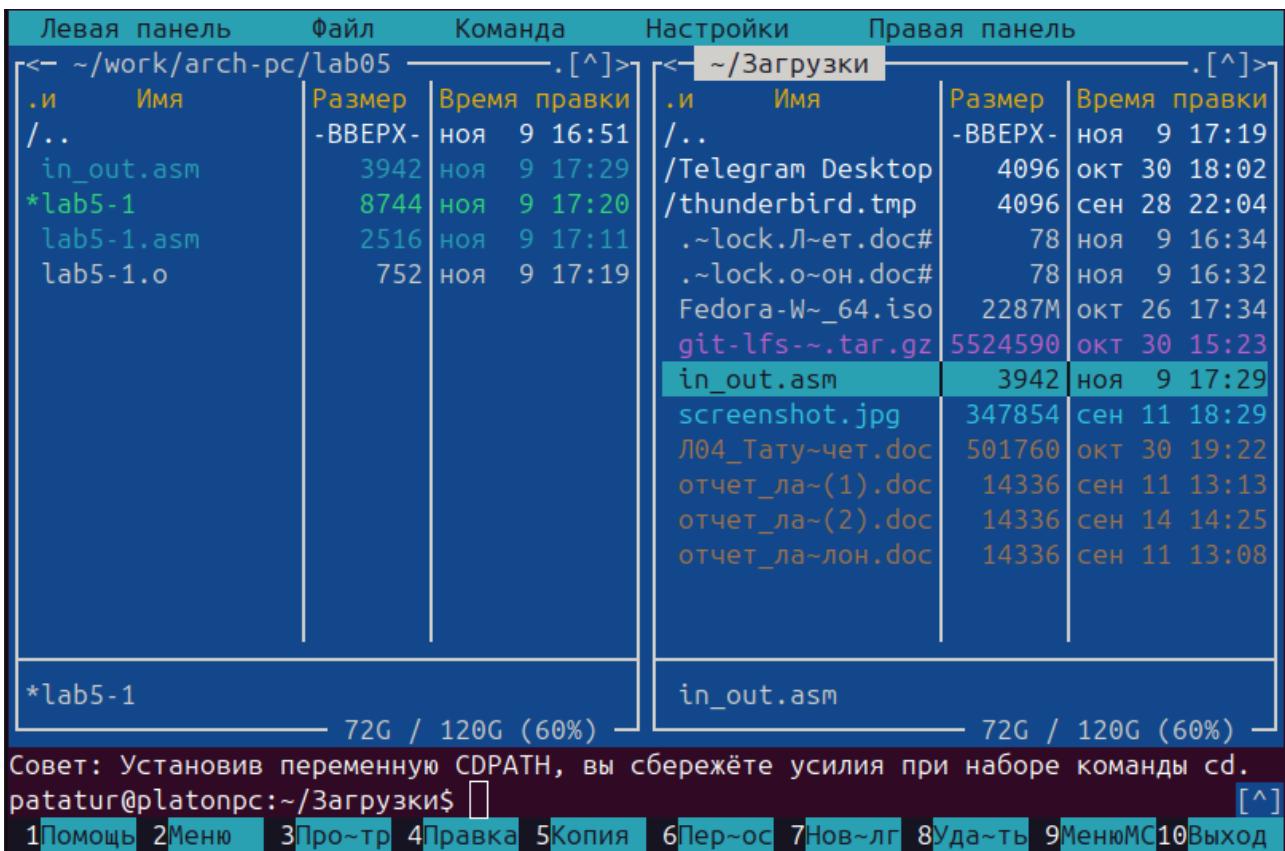


Рис. 10.3

10) 1. В одной из панелей mc открыл каталог с файлом lab5-1.asm. В другой панели каталог со скаченным файлом in\_out.asm (для перемещения между панелями использовал Tab). (см Рис. 10.1)

2. Скопировал файл in\_out.asm в каталог с файлом lab5-1.asm с помощью функциональной клавиши F5. (см Рис. 10.2 и Рис. 10.3)

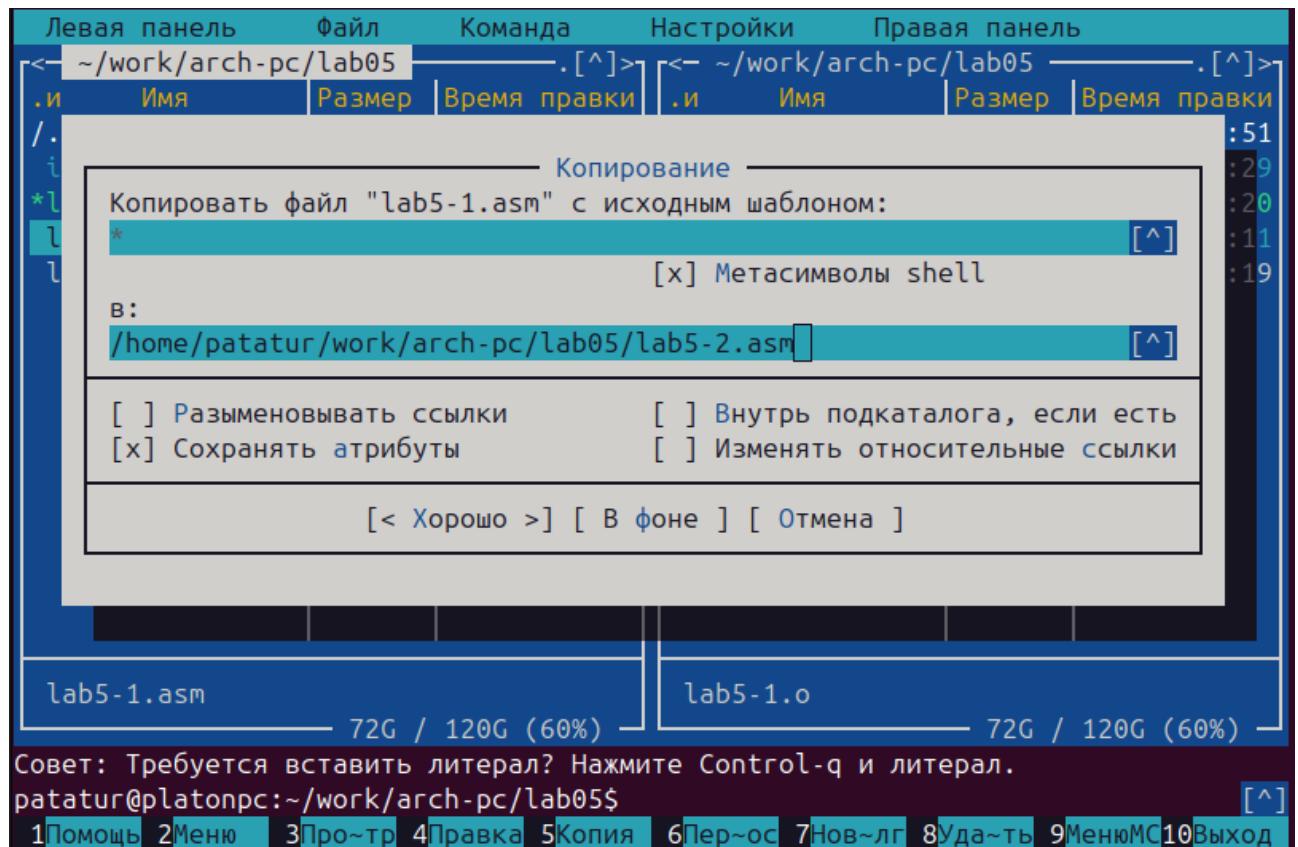


Рис. 11.1

Левая панель		Файл	Команда	Настройки	Правая панель
<- ~/work/arch-pc/lab05					<- ~/work/arch-pc/lab05
.	Имя	Размер	Время правки	.	Имя
/..		-ВВЕРХ-	ноя 9 16:51	/..	
in_out.asm		3942	ноя 9 17:29	in_out.asm	
*lab5-1		8744	ноя 9 17:20	*lab5-1	
<b>lab5-1.asm</b>		2516	ноя 9 17:11	lab5-1.asm	
lab5-1.o		752	ноя 9 17:19	lab5-1.o	
lab5-2.asm		2516	ноя 9 17:11	lab5-2.asm	

lab5-1.asm	72G / 120G (60%)
lab5-1.o	72G / 120G (60%)

Совет: Требуется вставить литерал? Нажмите Control-q и литерал.

patatur@platonpc:~/work/arch-pc/lab05\$ [^]

1Помощь 2Меню 3Про~тр 4Правка 5Копия 6Пер~ос 7Нов~лг 8Уда~ть 9МенюМС 10Выход

Рис. 11.2

11. С помощью функциональной клавиши F5 создал копию файла lab5-1.asm с именем lab5-2.asm. (см Рис. 11.1 и Рис. 11.2)

```
GNU nano 7.2          /home/patatur/work/arch-pc/lab05/lab5-2.asm *
;-----[-----;
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----[-----;
%include 'in_out.asm' ; подключение внешнего файла

SECTION .data          ; Секция инициализированных данных
msg: DB 'Введите строку: ',0h ; сообщение

SECTION .bss ; Секция не инициализированных 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 sprintLF ; 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

^G Справка ^O Записать ^W Поиск      ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^\ Замена       ^U Вставить ^J Выровнять ^/ К строке
```

Рис. 12.1

```
patatur@platonpc:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
patatur@platonpc:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
```

Рис. 12.2

```
patatur@platonpc:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Татур Платон Андреевич
patatur@platonpc:~/work/arch-pc/lab05$ 
```

Рис. 12.3

12. Исправил текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in\_out.asm (используя подпрограммы sprintLF, sread и quit) в соответствии с листингом 5.2 (см Рис. 12.1). Создал исполняемый файл и проверил его работу. (см Рис. 12.2 и Рис. 12.3)

```

GNU nano 7.2          /home/patatur/work/arch-pc/lab05/lab5-2.asm
-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
-----
%include 'in_out.asm' ; подключение внешнего файла

SECTION .data          ; Секция инициализированных данных
msg: DB 'Введите строку: ',0h ; сообщение

SECTION .bss ; Секция не инициализированных 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

^G Справка ^O Записать ^W Поиск      ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^\ Замена      ^U Вставить ^J Выровнять ^/ К строке

```

Рис. 13.1

```

patatur@platonpc:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
patatur@platonpc:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
patatur@platonpc:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Татур Платон Андреевич
patatur@platonpc:~/work/arch-pc/lab05$ 

```

Рис. 13.2

13. В файле lab5-2.asm заменил подпрограмму sprintLF на sprint. Создал исполняемый файл и проверил его работу. Разница в том что sprintLF переносит ввод на следующую строку после “Введите строку”, а sprint нет.

## 5.4. Задание для самостоятельной работы

1. Создайте копию файла lab5-1.asm. Внесите 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 введённую строку на экран.

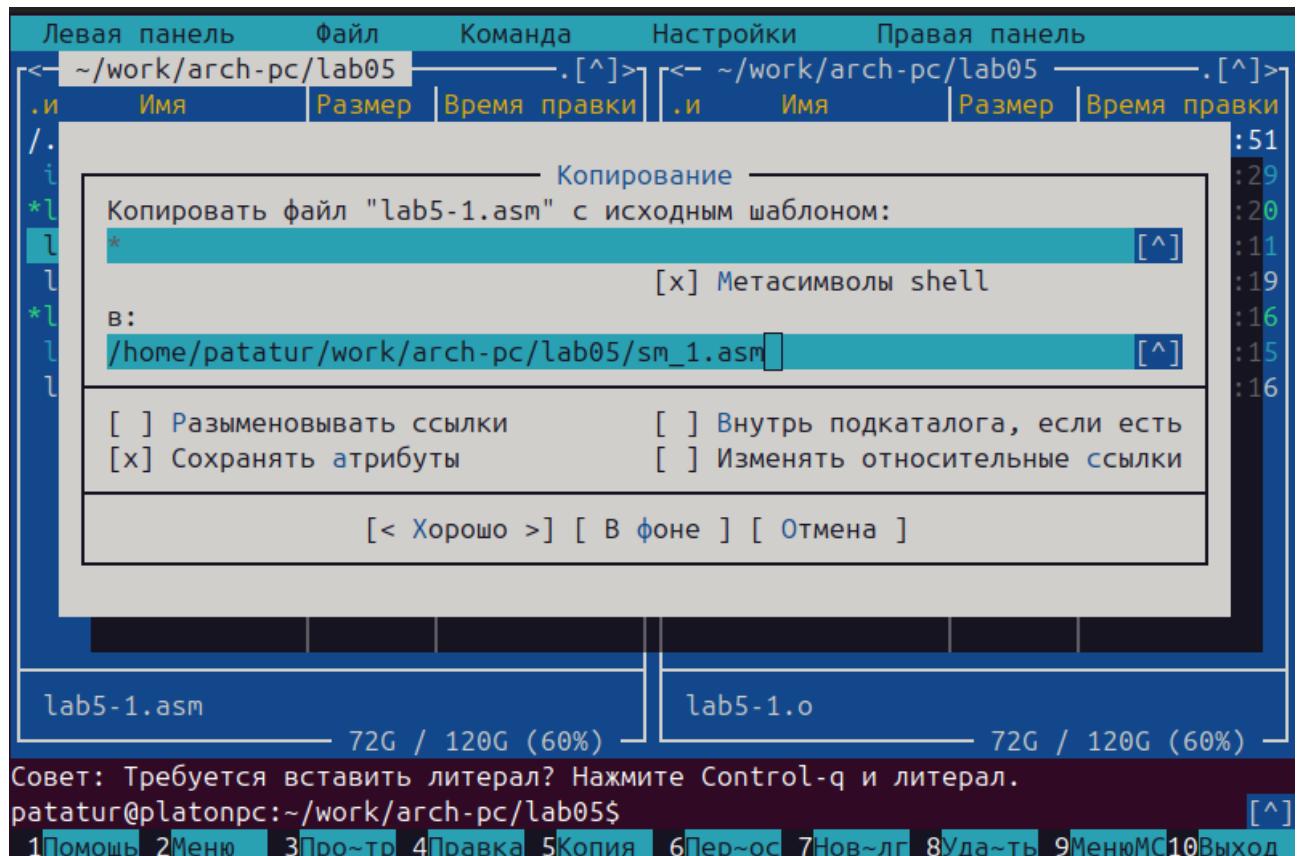


Рис. ср1.1

Левая панель		Файл	Команда	Настройки	Правая панель					
<-	~/work/arch-pc/lab05		.[ <sup>^</sup> ]	<-	~/work/arch-pc/lab05	.[ <sup>^</sup> ]				
.	Имя	Размер	Время правки	.	Имя	Размер				
..		-БВЕРХ-	ноя 9 16:51	..		-БВЕРХ-				
in_out.asm		3942	ноя 9 17:29	in_out.asm		3942				
*lab5-1		8744	ноя 9 17:20	*lab5-1		8744				
lab5-1.asm		2516	ноя 9 17:11	lab5-1.asm		2516				
lab5-1.o		752	ноя 9 17:19	lab5-1.o		752				
*lab5-2		9092	ноя 9 18:16	*lab5-2		9092				
lab5-2.asm		1259	ноя 9 18:15	lab5-2.asm		1259				
lab5-2.o		1312	ноя 9 18:16	lab5-2.o		1312				
sm_1.asm		2516	ноя 9 17:11	sm_1.asm		2516				
lab5-1.asm		72G / 120G (60%)		lab5-1.o		72G / 120G (60%)				
Совет: Требуется вставить литерал? Нажмите Control-q и литерал.										
patatur@platonpc:~/work/arch-pc/lab05\$ [ <sup>^</sup> ]										
1Помощь 2Меню 3Про~тр 4Правка 5Копия 6Пер~ос 7Нов~лг 8Уда~ть 9МенюМС 10Выход										

Рис. ср1.2

```

GNU nano 7.2          /home/patatur/work/arch-pc/lab05/sm_1.asm
;-----
; 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)
;
```

**Справка** **Записать** **Поиск** **Вырезать** **Выполнить** **Позиция**  
**Выход** **ЧитФайл** **Замена** **Вставить** **Выровнять** **К строке**

Рис. ср1.3

```
GNU nano 7.2          /home/pataturg/work/argch-pc/lab05/sm_1.asm
; 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

----- изменения повторный вызов 'write' -----
mov eax,4      ; Системный вызов для записи (sys_write)
mov ebx,1      ; Описатель файла 1 - стандартный вывод
mov ecx,buf1  ; Адрес строки 'buf1' в 'ecx'
mov edx,msgLen ; Размер строки 'msg' в 'edx'

^G Справка ^O Записать ^W Поиск      ^K Вырезать ^T Выполнить ^C Позиция
^X Выход   ^R ЧитФайл ^\ Замена      ^U Вставить ^J Выровнять ^/ К строке
```

Рис. ср1.4

```
GNU nano 7.2          /home/pataturg/work/argch-pc/lab05/sm_1.asm
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

----- изменения повторный вызов 'write' -----
mov eax,4      ; Системный вызов для записи (sys_write)
mov ebx,1      ; Описатель файла 1 - стандартный вывод
mov ecx,buf1  ; Адрес строки 'buf1' в 'ecx'
mov edx,msgLen ; Размер строки 'msg' в 'edx'
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
□

^G Справка ^O Записать ^W Поиск      ^K Вырезать ^T Выполнить ^C Позиция
^X Выход   ^R ЧитФайл ^\ Замена      ^U Вставить ^J Выровнять ^/ К строке
```

Рис. ср1.5

Ответ: Создал копию файла lab5-1.asm с именем sm\_1.asm (см Рис. сп1.1 и сп1.2.). Внес изменения в программу (без использования внешнего файла in\_out.asm) (см Рис. сп1.5). Для вывода введенной ранее строки я добавил ещё один вызов ‘write’, и заменил адрес выводимой строки с msg на buf1. В buf1 ранее записалась введенная строка. (см Рис. сп1.5)

2. Получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите свою фамилию.

```
patatur@platonpc:~/work/arch-pc/lab05$ nasm -f elf sm_1.asm
patatur@platonpc:~/work/arch-pc/lab05$ ld -m elf_i386 -o sm_1 sm_1.o
patatur@platonpc:~/work/arch-pc/lab05$ ./sm_1
Введите строку:
Татур
Татур
patatur@platonpc:~/work/arch-pc/lab05$
```

Рис. сп2

Ответ: На Рис. сп2 видно, что программа успешно выполняется

3. Создайте копию файла lab5-2.asm. Исправьте 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.

```
GNU nano 7.2          /home/patatur/work/arch-pc/lab05/sm_2.asm *
-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
-----
%include 'in_out.asm' ; подключение внешнего файла

SECTION .data          ; Секция инициализированных данных
msg: DB 'Введите строку: ',0h ; сообщение

SECTION .bss ; Секция не инициализированных 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 sprintLF ; вызов подпрограммы печати сообщения
    mov ecx, buf1 ; запись адреса переменной в `ECX`
    mov edx, 80   ; запись длины вводимого сообщения в `EBX`
    call sread    ; вызов подпрограммы ввода сообщения

-----
изменения

^G Справка ^O Записать ^W Поиск      ^K Вырезать ^T Выполнить ^C Позиция
^X Выход   ^R ЧитФайл  ^\ Замена       ^U Вставить ^J Выровнять ^/ К строке
```

Рис. ср3.1

```
GNU nano 7.2          /home/patatur/work/arch-pc/lab05/sm_2.asm *
msg: DB 'Введите строку: ',0h ; сообщение

SECTION .bss ; Секция не инициализированных 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 sprintLF ; вызов подпрограммы печати сообщения
    mov ecx, buf1 ; запись адреса переменной в `ECX`
    mov edx, 80   ; запись длины вводимого сообщения в `EBX`
    call sread    ; вызов подпрограммы ввода сообщения

-----
изменения
    mov eax,buf1 ; из buf1 в `EAX`
    mov ecx,eax ; из `EAX` в `ECX`
    call sprint ; вызов подпрограммы печати сообщения

    call quit    ; вызов подпрограммы завершения
    ^

^G Справка ^O Записать ^W Поиск      ^K Вырезать ^T Выполнить ^C Позиция
^X Выход   ^R ЧитФайл  ^\ Замена       ^U Вставить ^J Выровнять ^/ К строке
```

Рис. ср3.2

Ответ: Создал копию файла lab5-2.asm с именем sm\_2.asm. Внес изменения в программу (с использованием внешнего файла in\_out.asm) (см Рис. ср3.2). Для вывода введенной ранее строки я переместил адрес переменной buf1 в ecx, а затем из ecx в eax и вызвал подпрограмму sprint. (см Рис. ср3.2)

4. Создайте исполняемый файл и проверьте его работу.

```
patatur@platonpc:~/work/arch-pc/lab05$ nasm -f elf sm_2.asm
patatur@platonpc:~/work/arch-pc/lab05$ ld -m elf_i386 -o sm_2 sm_2.o
patatur@platonpc:~/work/arch-pc/lab05$ ./sm_2
Введите строку:
Татур
Татур
patatur@platonpc:~/work/arch-pc/lab05$ █
```

Рис. ср4

Ответ: На Рис. ср4 видно, что программа успешно выполняется

## **Вывод**

В ходе выполнения лабораторной работы я приобрел практические навыки работы в Midnight Commander. Освоил инструкций языка ассемблера mov и int