

Шаблон отчёта по лабораторной работе

Простейший вариант

Лупупа Чилеше

Содержание

1	Цель работы	5
2	Задание	6
2.1	Порядок выполнения лабораторной работы	6
3	Задание для самостоятельной работы	17
4	Выводы	23

Список иллюстраций

2.1	mc	6
2.2	~/work/arch-pc	7
2.3	lab5-1.asm	8
2.4	nano	9
2.5	листинга 5.1	10
2.6	f3	11
2.7	объектный файл	11
2.8	объектный файл	12
2.9	in_out.asm	13
2.10	Перенос в каталог lab05	14
2.11	lab5-2.asm.	14
2.12	lab5-2.asm.	14
2.13	sprintLF, sread и quit	15
2.14	sprint	16
3.1	lab5-3.asm	18
3.2	lab5-3.asm	19
3.3	lab5-3.asm	19

Список таблиц

1 Цель работы

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

2 Задание

2.1 Порядок выполнения лабораторной работы

1. Откройте Midnight Commander

`user@dk4n31:~$ mc`

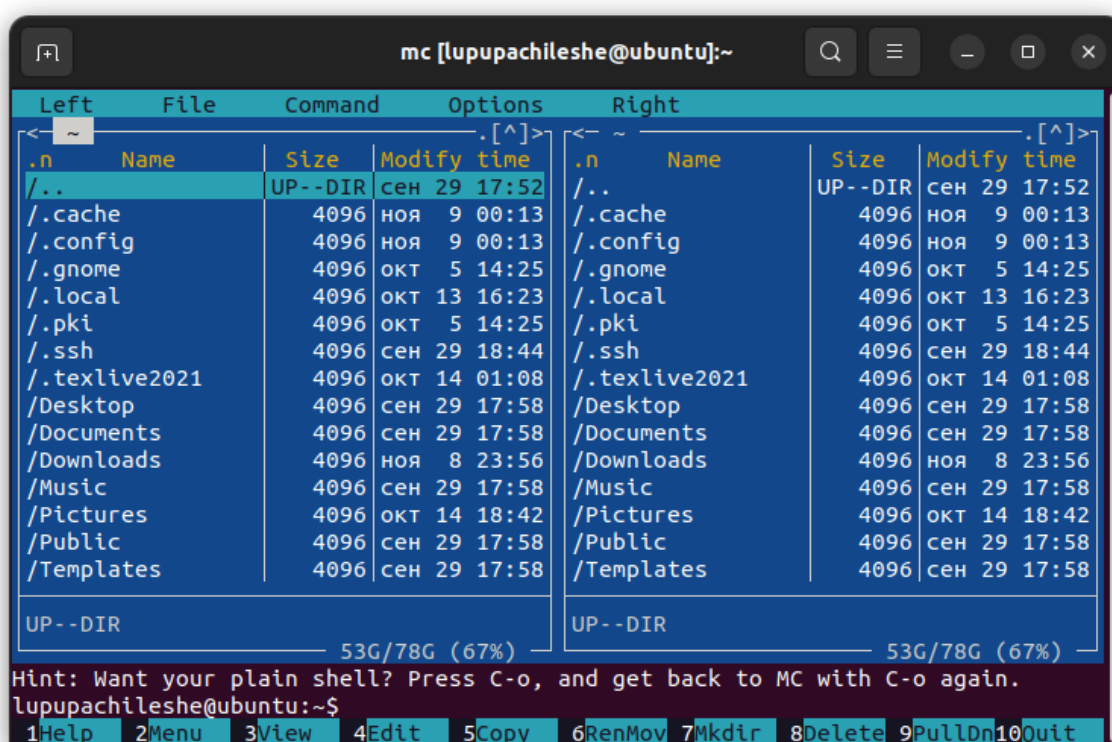


Рис. 2.1: mc

__комментарий__:Открыл Midnight Commander на терминале

2. Пользуясь клавишами **⌘** , **⌘** и Enter перейдите в каталог ~/work/arch-pc созданный при выполнении лабораторной работы №4

комментарий:С помощью стрелки Enter и клавиш я открыл каталог Arch-PC.

3. С помощью функциональной клавиши F7 создайте папку lab05 и перейдите в созданный каталог

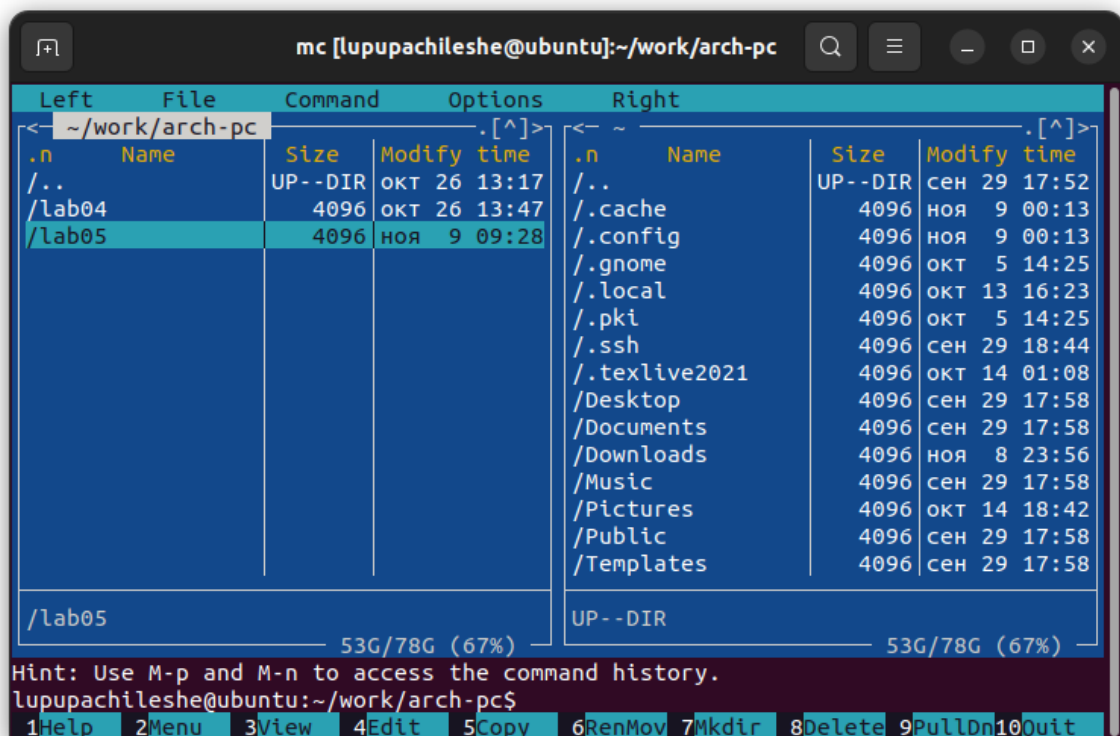


Рис. 2.2: ~/work/arch-pc

комментарий:Используя функциональную клавишу F7, я создал папку lab05.

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

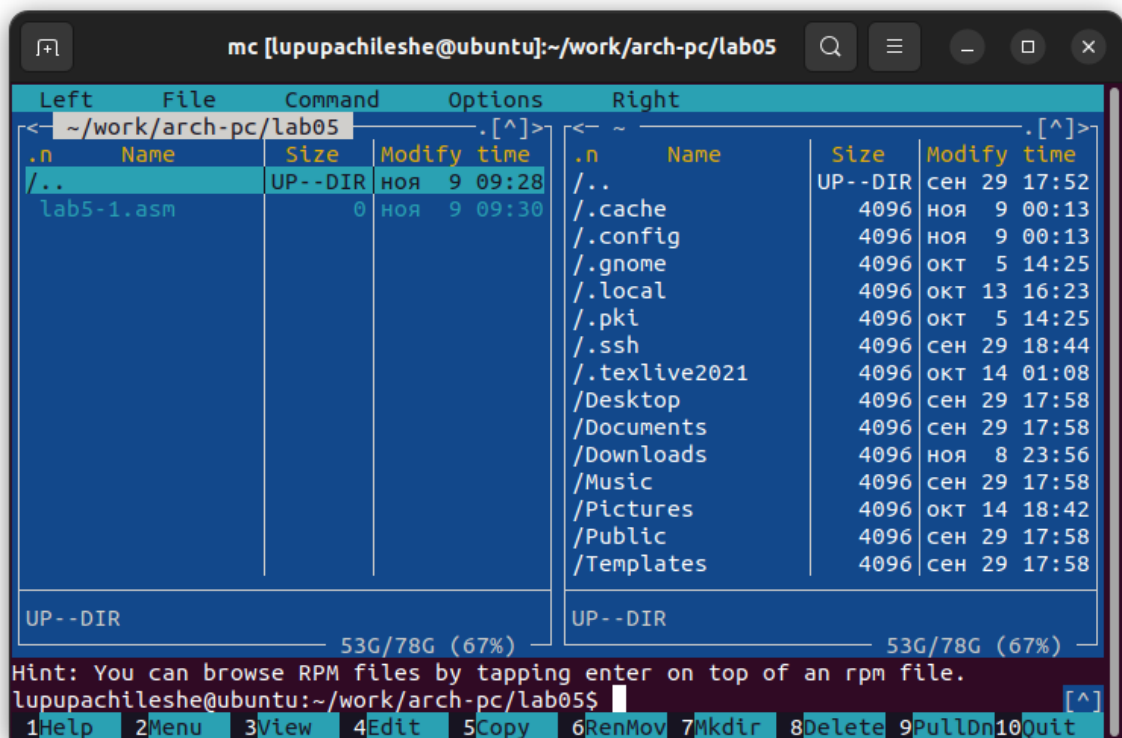


Рис. 2.3: lab5-1.asm

__комментарий__: С помощью команды `touch` я создал ассемблерный файл `lab5-1.asm`.

5. С помощью функциональной клавиши F4 откройте файл `lab5-1.asm` для редактирования во встроенном редакторе.

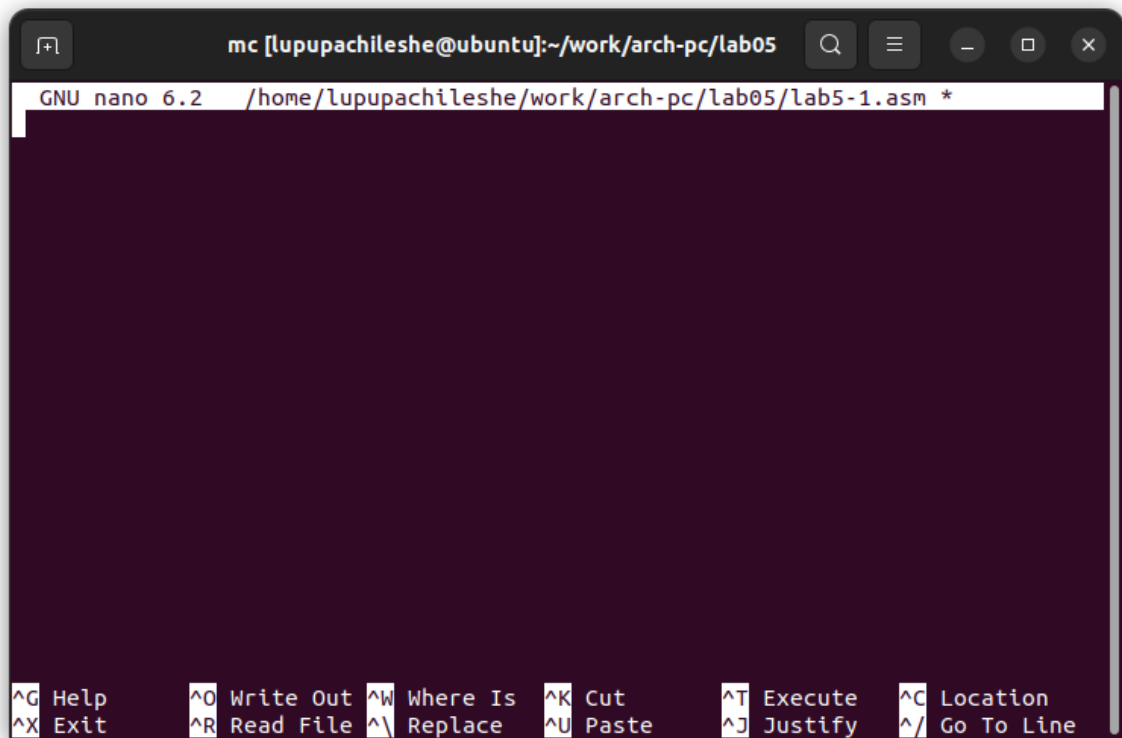


Рис. 2.4: nano

__комментарий__: Я открыл файл lab5-1.asm.

6. Введите текст программы из листинга 5.1 (можно без комментариев), сохраните изменения и закройте файл.

```
mc [lupupachileshe@ubuntu]:~/work/arch-pc/lab05
GNU nano 6.2 /home/lupupachileshe/work/arch-pc/lab05/lab5-1.asm *
;----- 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

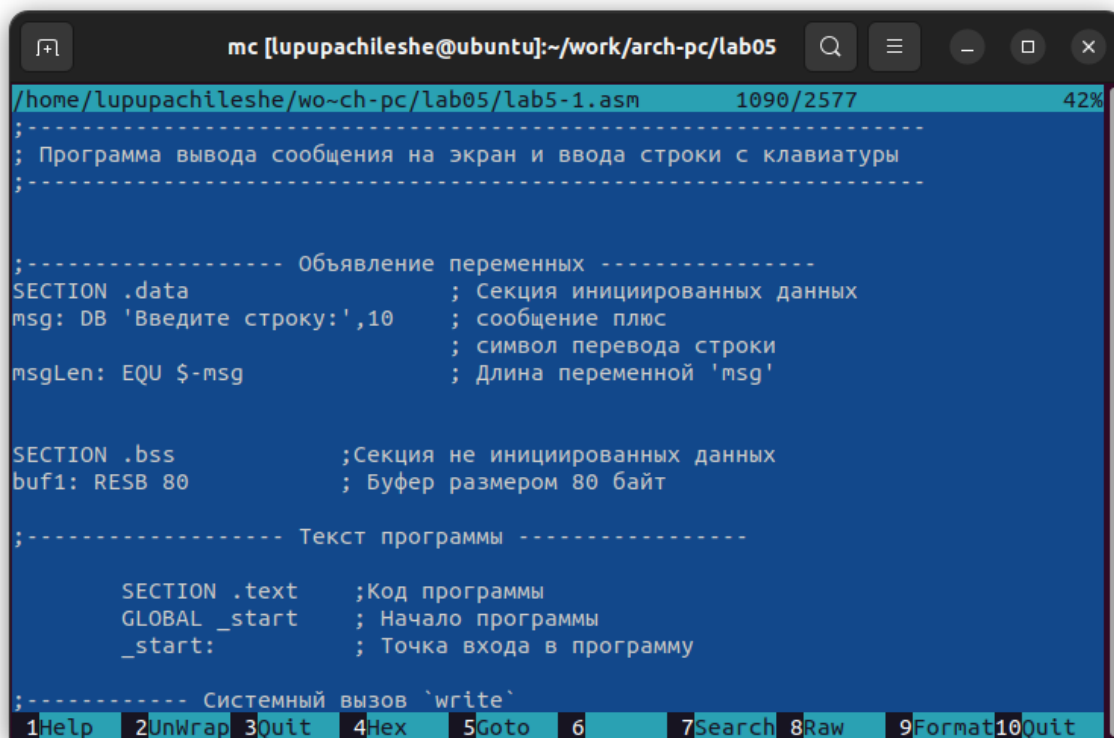
    mov eax, 3      ; Системный вызов для чтения (sys_read)
    mov ebx, 0      ; Дескриптор файла 0 - стандартный ввод
    mov ecx, buf1   ; Адрес буфера под вводимую строку
    mov edx, 80     ; Длина вводимой строки
    int 80h        ; Вызов ядра

[ Cancelled ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

Рис. 2.5: листинга 5.1

__комментарий__:Текст из листинга 5.1 я скопировал в файл lab5-1.asm.

7. С помощью функциональной клавиши F3 откройте файл lab5-1.asm для просмотра. Убедитесь, что файл содержит текст программы.



```
mc [lupupachileshe@ubuntu]:~/work/arch-pc/lab05
/home/lupupachileshe/wo~ch-pc/lab05/lab5-1.asm 1090/2577 42%
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----

;----- Объявление переменных -----
SECTION .data                ; Секция инициированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
                                ; символ перевода строки
msgLen: EQU $-msg            ; Длина переменной 'msg'

SECTION .bss                 ;Секция не инициированных данных
buf1: RESB 80                ; Буфер размером 80 байт

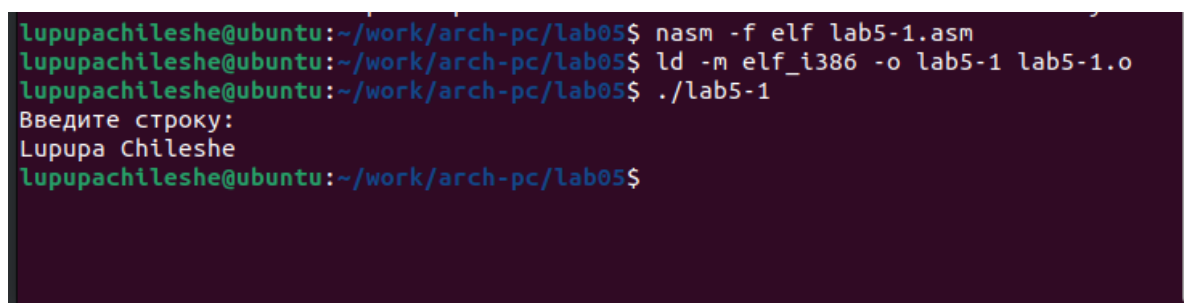
;----- Текст программы -----
SECTION .text                ;Код программы
GLOBAL _start                ; Начало программы
_start:                      ; Точка входа в программу

;----- Системный вызов `write` -----
1Help 2UnWrap 3Quit 4Hex 5Goto 6 7Search 8Raw 9Format10Quit
```

Рис. 2.6: f3

__комментарий__: С помощью функциональной клавиши f3 я просмотрел, чтобы убедиться

- Оттранслируйте текст программы lab5-1.asm в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл. Программа выводит строку 'Введите строку:' и ожидает ввода с клавиатуры. На запрос введите Ваши ФИО.



```
lupupachileshe@ubuntu:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
lupupachileshe@ubuntu:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
lupupachileshe@ubuntu:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Lurupa Chileshe
lupupachileshe@ubuntu:~/work/arch-pc/lab05$
```

Рис. 2.7: объектный файл

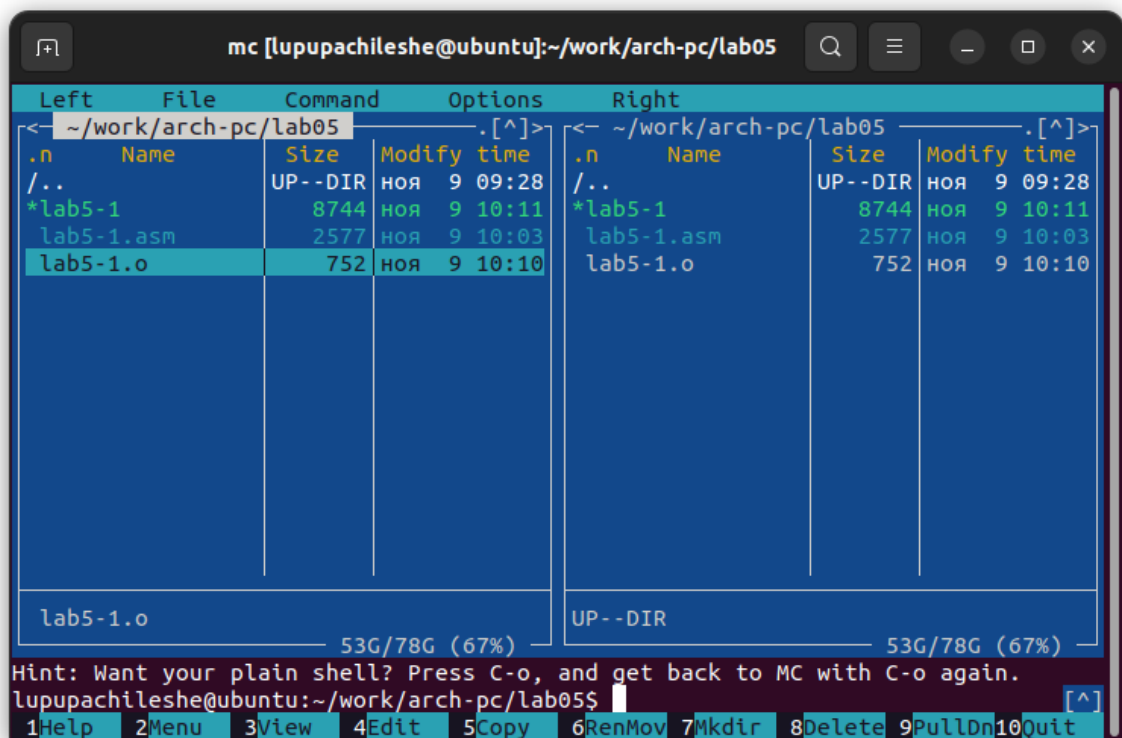
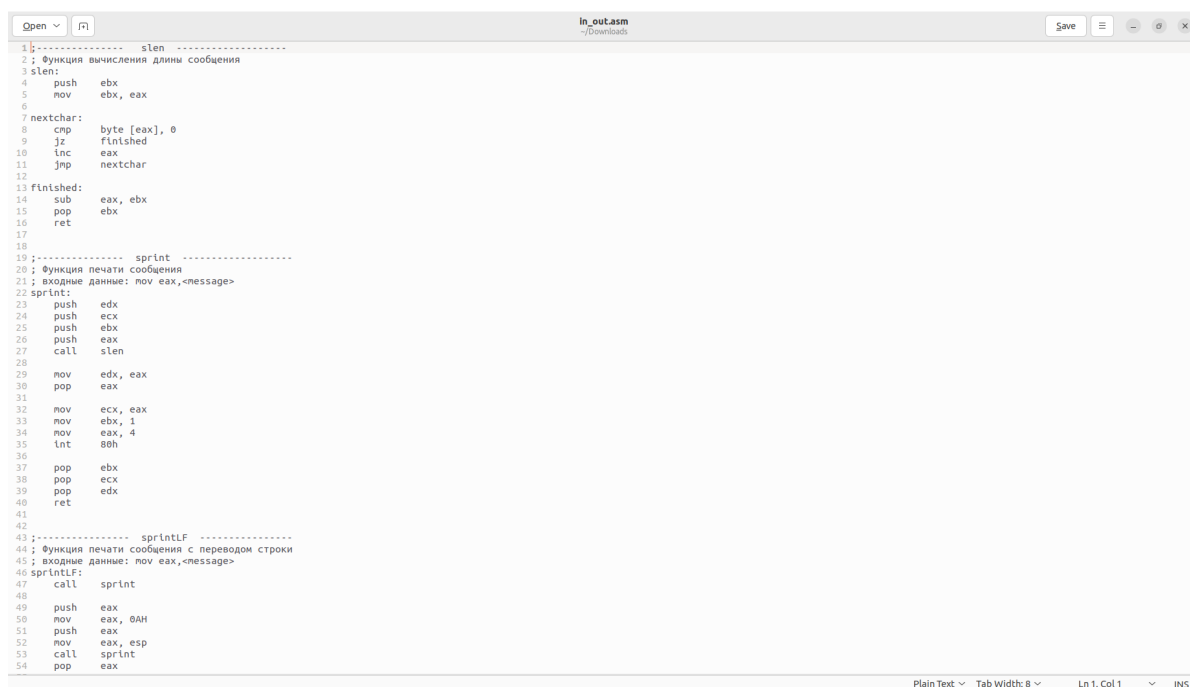


Рис. 2.8: объектный файл

__комментарий__:Перевел текст программы lab5-1.asm в объектный файл

9. Скачайте файл in_out.asm со страницы курса в ТУИС.



```
1 ;----- slen -----
2 ; Функция вычисления длины сообщения
3 slen:
4     push    ebx
5     mov     ebx, eax
6
7 nextchar:
8     cmp     byte [eax], 0
9     jz      finished
10    inc     eax
11    jmp     nextchar
12
13 finished:
14    sub     eax, ebx
15    pop     ebx
16    ret
17
18 ;----- sprint -----
19 ; Функция печати сообщения
20 ; входные данные: mov eax, <message>
21
22 sprint:
23     push    edx
24     push    ecx
25     push    ebx
26     push    eax
27     call    slen
28
29     mov     edx, eax
30     pop     eax
31
32     mov     ecx, eax
33     mov     ebx, 1
34     mov     eax, 4
35     int     80h
36
37     pop     ebx
38     pop     ecx
39     pop     edx
40     ret
41
42 ;----- sprintLF -----
43 ; Функция печати сообщения с переводом строки
44 ; входные данные: mov eax, <message>
45
46 sprintLF:
47     call    sprint
48
49     push    eax
50     mov     eax, 0Ah
51     push    eax
52     mov     eax, esp
53     call    sprint
54     pop     eax
```

Рис. 2.9: in_out.asm

__комментарий__: Скачал файл 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.

В одной из панелей mc откройте каталог с файлом lab5-1.asm. В другой панели каталог со скаченным файлом in_out.asm (для перемещения между панелями используйте Tab). Скопируйте файл in_out.asm в каталог с файлом lab5-1.asm с помощью функциональной клавиши F5.

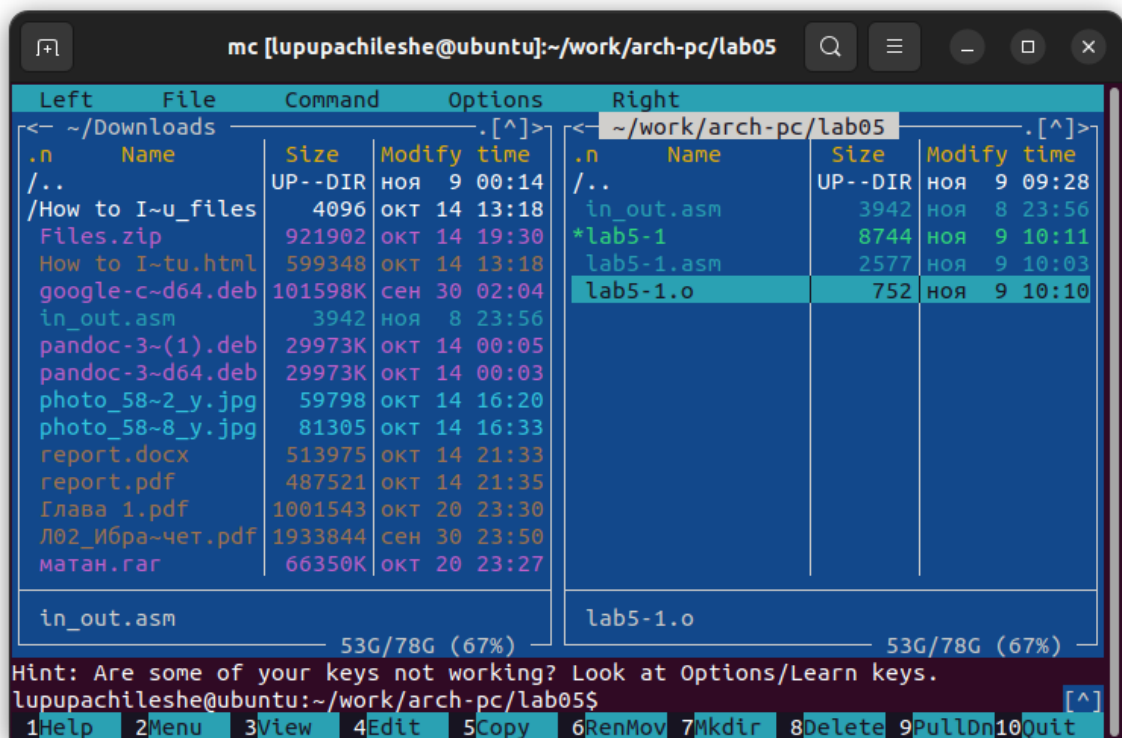


Рис. 2.10: Перенос в каталог lab05

__комментарий__:Передайте скачанный файл в лабораторию 5.

11. С помощью функциональной клавиши F6 создайте копию файла lab5-1.asm с именем lab5-2.asm. Выделите файл lab5-1.asm, нажмите клавишу F6 , введите имя файла lab5-2.asm и нажмите клавишу Enter.

*lab5-2	8744	ноя 11 20:40	*lab5-2	8744	ноя 11 20:40
lab5-2.asm	2578	ноя 11 19:36	lab5-2.asm	2578	ноя 11 19:36
lab5-2.o	752	ноя 11 20:33	lab5-2.o	752	ноя 11 20:33

Рис. 2.11: lab5-2.asm.

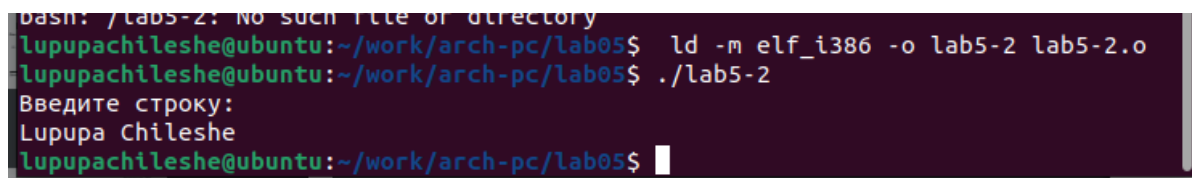
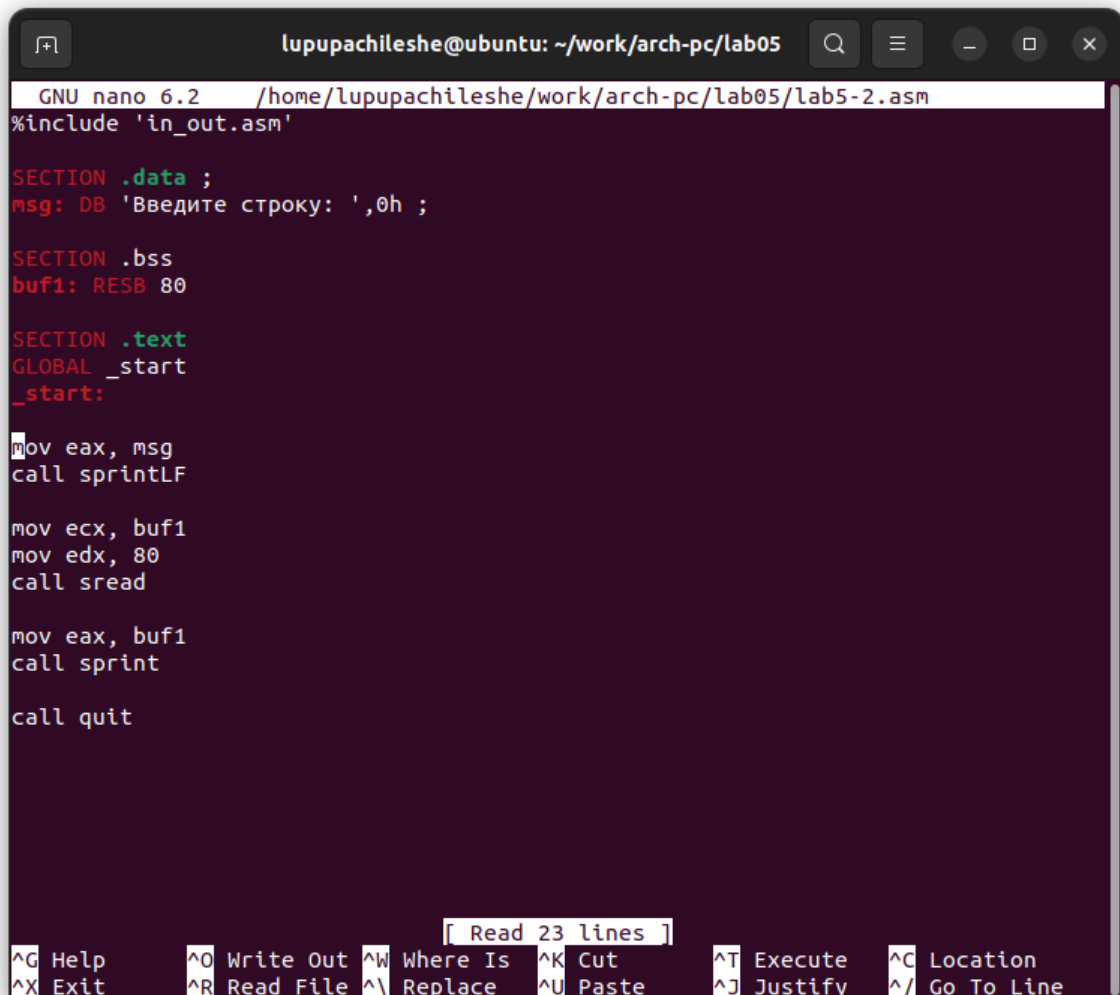


Рис. 2.12: lab5-2.asm.

12. Исправьте текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in_out.asm (используйте подпрограммы sprintLF, sread и quit) в соответствии с листингом 5.2. Создайте исполняемый файл и проверьте его работу.



```
GNU nano 6.2 /home/lupupachileshe/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

mov eax, buf1
call sprint

call quit
```

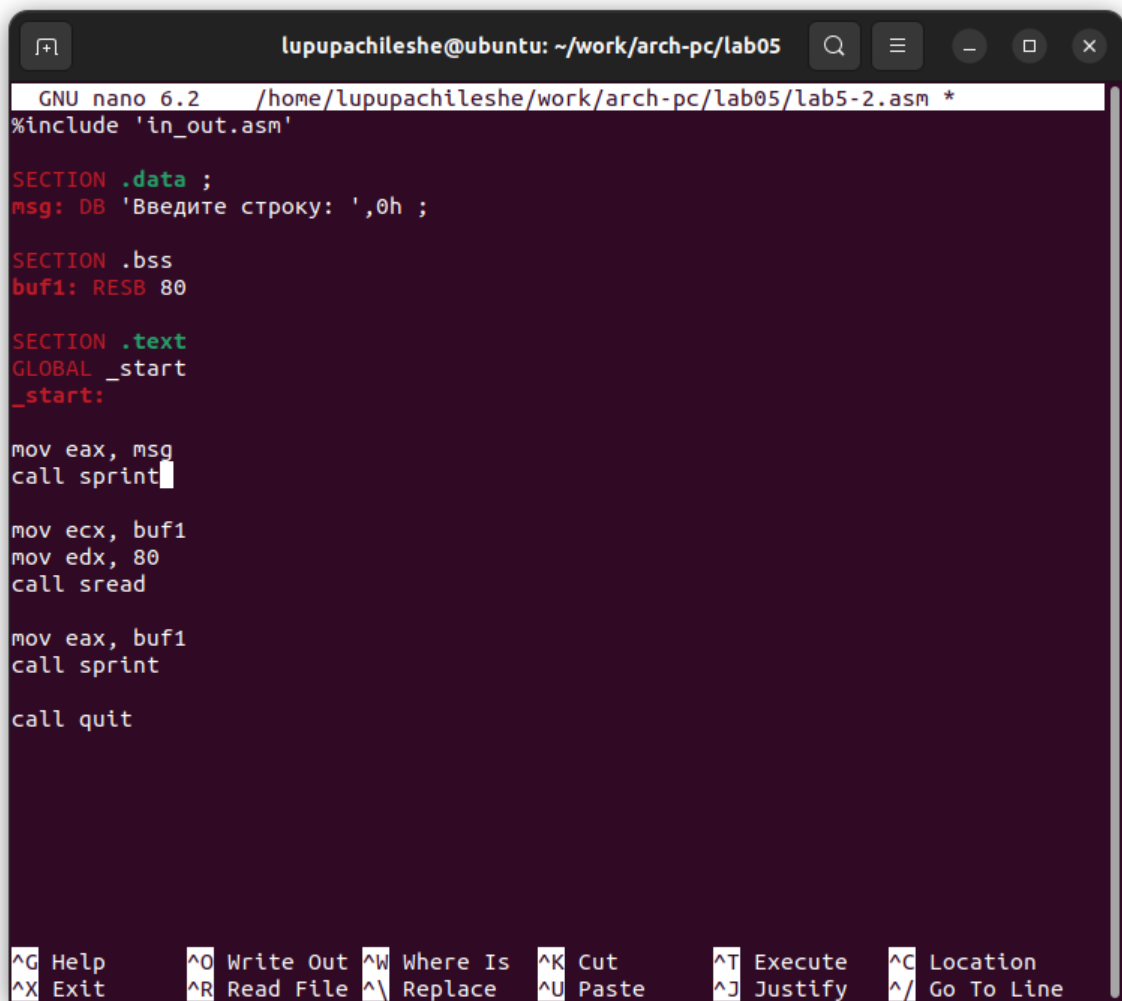
[Read 23 lines]

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location
^X Exit	^R Read File	^I Replace	^U Paste	^J Justify	^_ Go To Line

Рис. 2.13: sprintLF, sread и quit

__комментарий__:Исправил текст программы в файле lab5-2.asm с помощью подпрограмм

13. В файле lab5-2.asm замените подпрограмму sprintLF на sprint. Создайте исполняемый файл и проверьте его работу. В чем разница?



```
GNU nano 6.2 /home/lupupachileshe/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 sprint

mov ecx, buf1
mov edx, 80
call sread

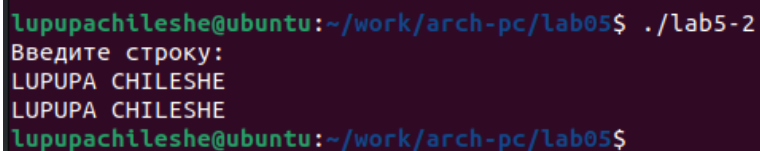
mov eax, buf1
call sprint

call quit
```

Help Write Out Where Is Cut Execute Location
Exit Read File Replace Paste Justify Go To Line

Рис. 2.14: sprint

комментарий: Я изменил sprintLF на sprint.



```
lupupachileshe@ubuntu:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
LUPUPA CHILESHE
LUPUPA CHILESHE
lupupachileshe@ubuntu:~/work/arch-pc/lab05$
```


3 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.

```
lupupachileshe@ubuntu: ~/work/arch-pc/lab05
GNU nano 6.2 /home/lupupachileshe/work/arch-pc/lab05/lab5-3.asm
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----

;----- Объявление переменных -----
SECTION .data ; Секция инициированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'

SECTION .bss ;Секция не инициированных данных
buf1: RESB 80 ; Буфер размером 80 байт

;----- Текст программы -----

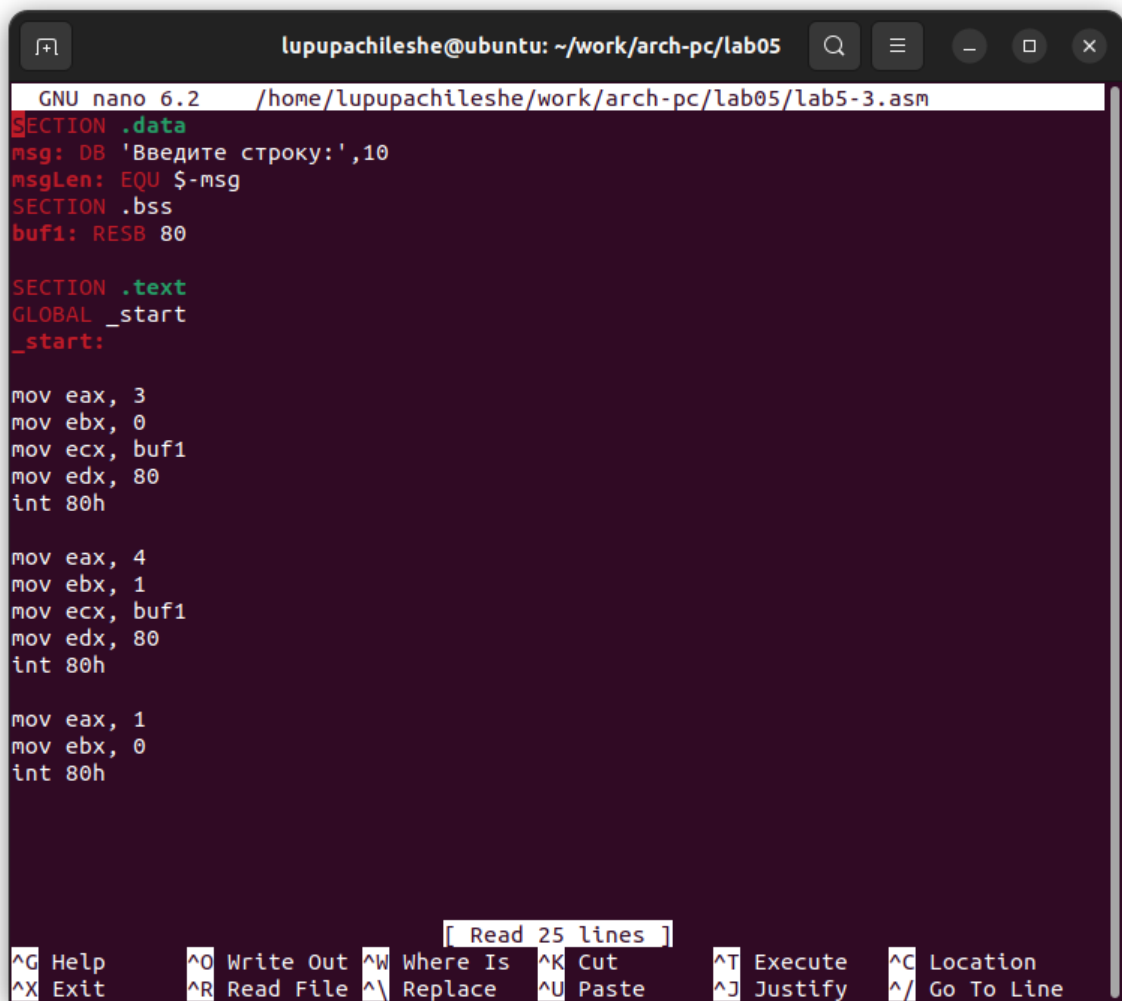
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу

;----- 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

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line
```

Рис. 3.1: lab5-3.asm



```
lupupachileshe@ubuntu: ~/work/arch-pc/lab05
GNU nano 6.2 /home/lupupachileshe/work/arch-pc/lab05/lab5-3.asm
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:

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

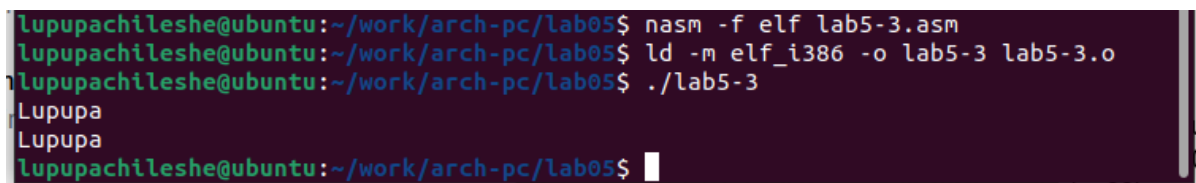
mov eax, 4
mov ebx, 1
mov ecx, buf1
mov edx, 80
int 80h

mov eax, 1
mov ebx, 0
int 80h

[ Read 25 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste       ^J Justify    ^_ Go To Line
```

Рис. 3.2: lab5-3.asm

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



```
lupupachileshe@ubuntu:~/work/arch-pc/lab05$ nasm -f elf lab5-3.asm
lupupachileshe@ubuntu:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-3 lab5-3.o
lupupachileshe@ubuntu:~/work/arch-pc/lab05$ ./lab5-3
Lupupa
Lupupa
lupupachileshe@ubuntu:~/work/arch-pc/lab05$
```

Рис. 3.3: lab5-3.asm

3. Создайте копию файла lab5-2.asm. Исправьте текст программы с использо-

вание 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.

mc [lupupachileshe@ubuntu]:~/work/arch-pc/lab05

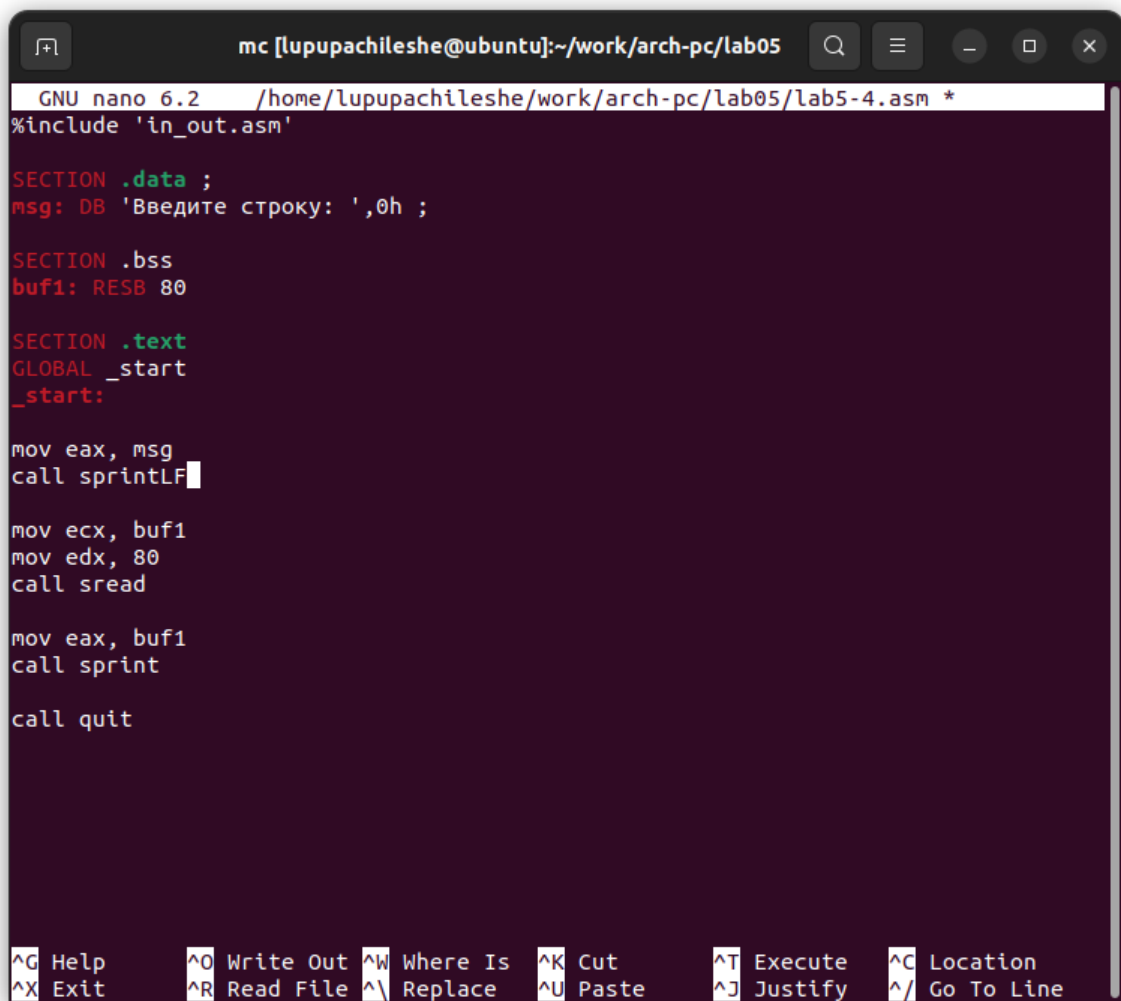
Left	File	Command	Options	Right
<-	~/work/arch-pc/lab05			<-
Left	File	Command	Options	Right
.n	Name	Size	Modify time	.n
UP--DIR	UP--DIR	UP--DIR	UP--DIR	UP--DIR
*/a.out	*/a.out	8744	ноя 11 20:38	*/a.out
in_out.asm	in_out.asm	3942	ноя 8 23:56	in_out.asm
*lab5-1	*lab5-1	8744	ноя 9 10:11	*lab5-1
lab5-1.asm	lab5-1.asm	2578	ноя 11 19:36	lab5-1.asm
lab5-1.o	lab5-1.o	752	ноя 9 10:10	lab5-1.o
*lab5-2	*lab5-2	9092	ноя 11 22:39	*lab5-2
lab5-2.asm	lab5-2.asm	251	ноя 11 22:41	lab5-2.asm
lab5-2.o	lab5-2.o	1328	ноя 11 22:38	lab5-2.o
*lab5-3	*lab5-3	8744	ноя 11 22:57	*lab5-3
lab5-3.asm	lab5-3.asm	283	ноя 11 22:56	lab5-3.asm
lab5-3.o	lab5-3.o	752	ноя 11 22:57	lab5-3.o
lab5-4.asm	lab5-4.asm	251	ноя 11 22:41	lab5-4.asm

lab5-2.asm 53G/78G (67%) UP--DIR 53G/78G (67%)

Hint: Want your plain shell? Press C-o, and get back to MC with C-o again.

lupupachileshe@ubuntu:~/work/arch-pc/lab05\$

1Help 2Menu 3View 4Edit 5Copy 6RenMov 7Mkdir 8Delete 9PullDn10Quit



```
mc [lupupachileshe@ubuntu]:~/work/arch-pc/lab05
GNU nano 6.2 /home/lupupachileshe/work/arch-pc/lab05/lab5-4.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

mov eax, buf1
call sprint

call quit

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify  ^_ Go To Line
```

4. Создайте исполняемый файл и проверьте его работу.

```
mc [lupupachileshe@ubuntu]:~/work/arch-pc/lab05
GNU nano 6.2 /home/lupupachileshe/work/arch-pc/lab05/lab5-4.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 sprint

mov ecx, buf1
mov edx, 80
call sread

mov eax, buf1
call sprint

call quit

[ Read 23 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

```
lupupachileshe@ubuntu:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-4 lab5-4.o
lupupachileshe@ubuntu:~/work/arch-pc/lab05$ ./lab5-4
Введите строку:
Chileshe
Chileshe
lupupachileshe@ubuntu:~/work/arch-pc/lab05$
```

4 Выводы

Я научился работать с Midnight Commander и языком ассемблера asm.