

. Основы работы с Midnight Commander (mc). Структура программы на языке ассемблера NASM. Системные вызовы в ОС GNU Linux

Язык разметки Markdown

Татьяна Александровна Буллер

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
2.1	Задание 1:	5
2.2	Задание 2:	6
2.3	Задание 3:	6
2.4	Задание 4:	7
2.5	Задание 5:	8
2.6	Задание 6:	9
2.7	Задание 7:	11
2.8	Задание 8:	12
2.9	Задание 9:	14
3	Задание для самостоятельной работы	16
3.1	Задание 10:	16
3.2	Задание 11:	18
4	Вывод	20

Список иллюстраций

2.1	Переход в каталог курса и введение команды на создание файла .	5
2.2	Выбор текстового редактора для дальнейшей работы с файлом . .	6
2.3	Введение текста программы из листинга	7
2.4	Тот же самый текст, но теперь открытый в Midnight Commander .	8
2.5	Превращение текста программы в объектный код	9
2.6	Создание нового объектного файла и файла листинга программы	9
2.7	Выполнение программы	9
2.8	Копирование файла	10
2.9	Файл скопирован	11
2.10	Создание копии файла	12
2.11	Исправленный код программы	13
2.12	Компиляция программы	13
2.13	Исправленный код программы	14
2.14	Компиляция обновленной программы	14
3.1	Исправленный код программы	17
3.2	Компиляция обновленной программы	17
3.3	Исправленный код программы	19
3.4	Компиляция обновленной программы	19

1 Цель работы

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

2 Выполнение лабораторной работы

2.1 Задание 1:

Откройте Midnight Commander. Пользуясь клавишами **←**, **→** и **Enter** перейдите в каталог `~/work/arch-рс`. Пользуясь строкой ввода и командой `touch` создайте файл `lab5-1.asm`.

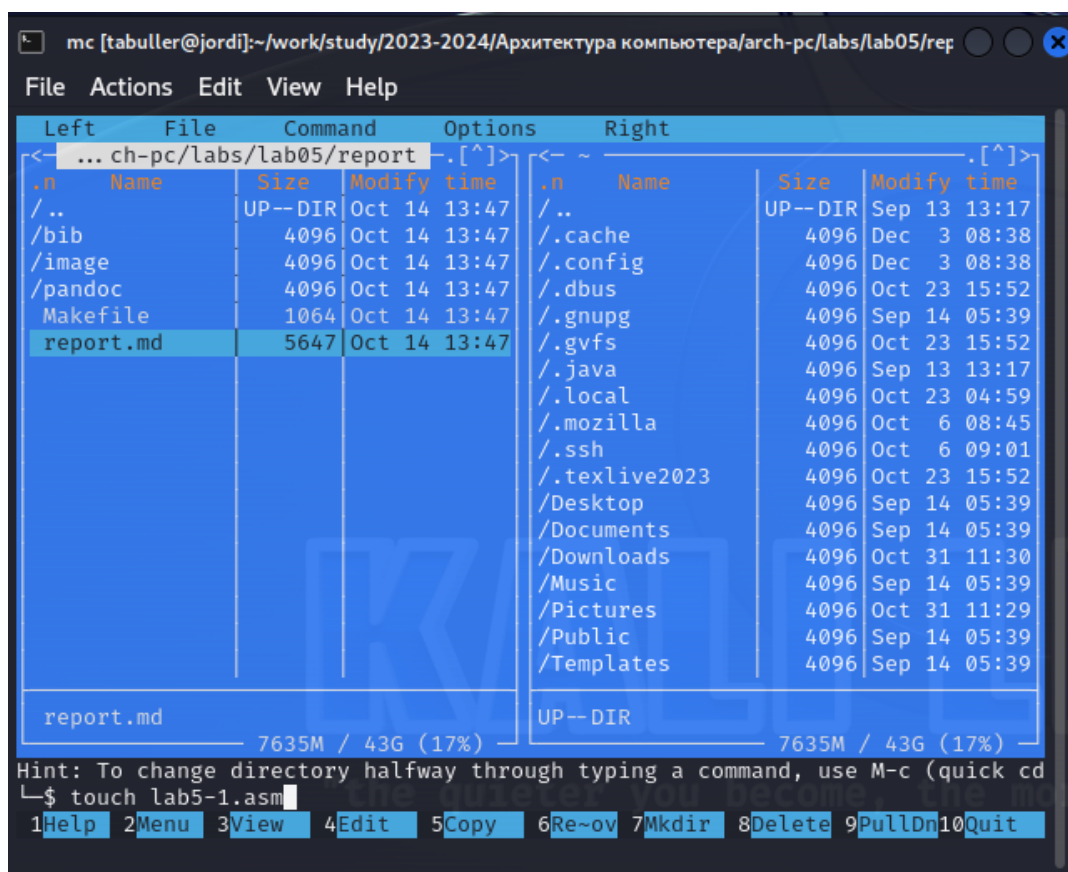
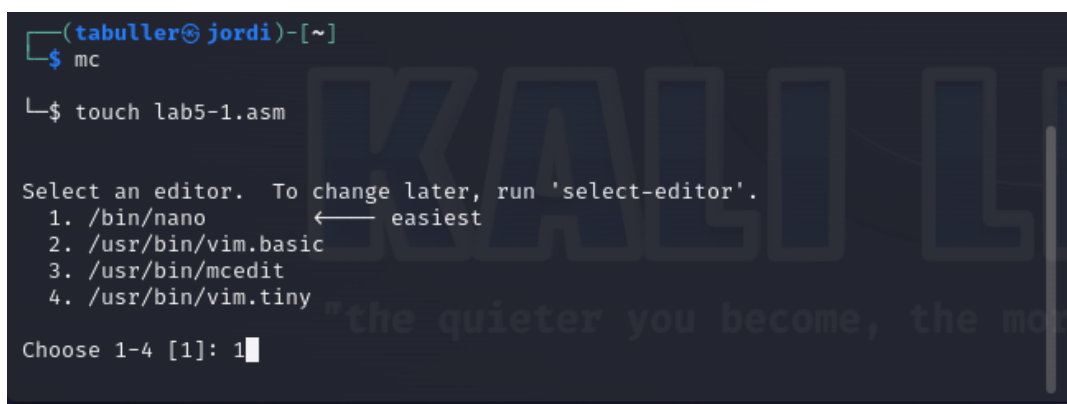


Рис. 2.1: Переход в каталог курса и введение команды на создание файла

Midnight Commander не был предустановлен на используемом мною дистрибутиве, но после установки работал исправно. В столбце справа можно видеть содержание директории lab05/report, в столбце слева - домашней директории. В строчке снизу введена команда для создания файла.

2.2 Задание 2:

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



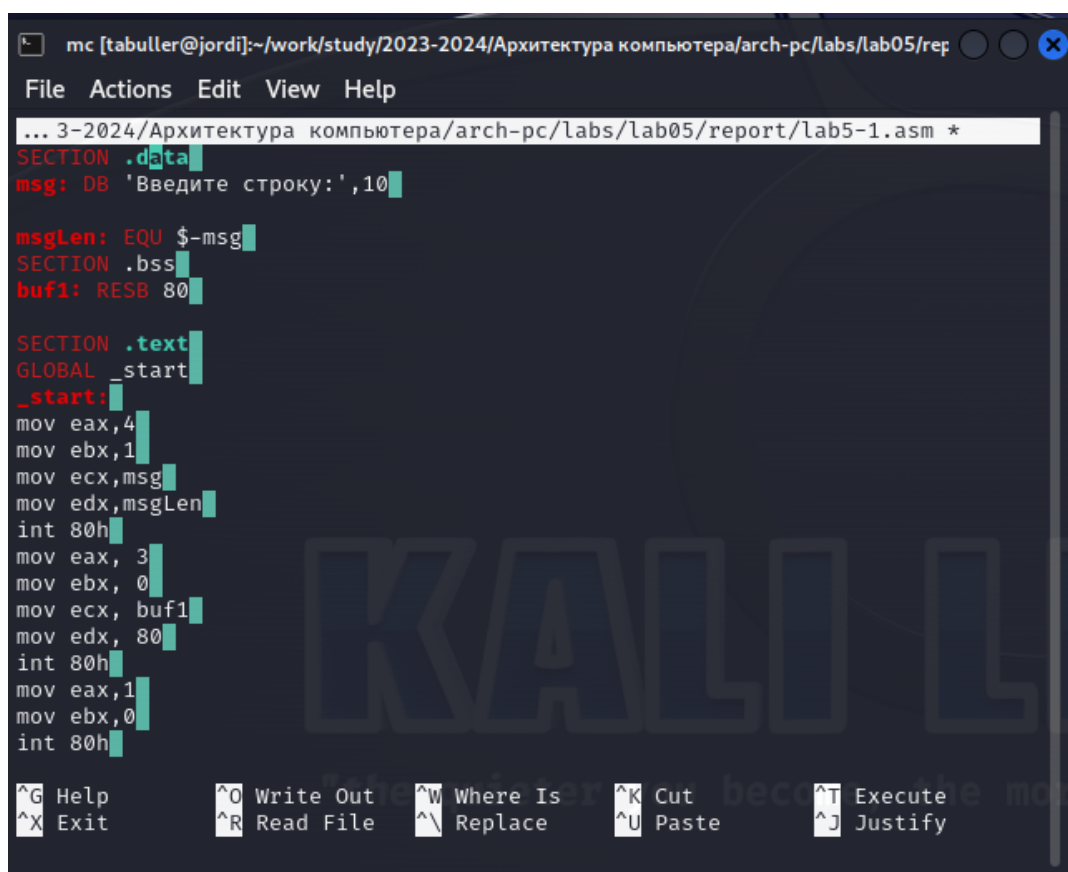
```
(tabuller@jordi)-[~]  
$ mc  
L$ touch lab5-1.asm  
  
Select an editor. To change later, run 'select-editor'.  
1. /bin/nano ← easiest  
2. /usr/bin/vim.basic  
3. /usr/bin/mcedit  
4. /usr/bin/vim.tiny  
Choose 1-4 [1]: 1
```

Рис. 2.2: Выбор текстового редактора для дальнейшей работы с файлом

Из предложенных на выбор редакторов я выбрала nano, для чего ввела в строку цифру 1.

2.3 Задание 3:

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



```
mc [tabuller@jordi]:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05/rep
File Actions Edit View Help
... 3-2024/Архитектура компьютера/arch-pc/labs/lab05/report/lab5-1.asm *
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
int 80h
mov eax,1
mov ebx,0
int 80h

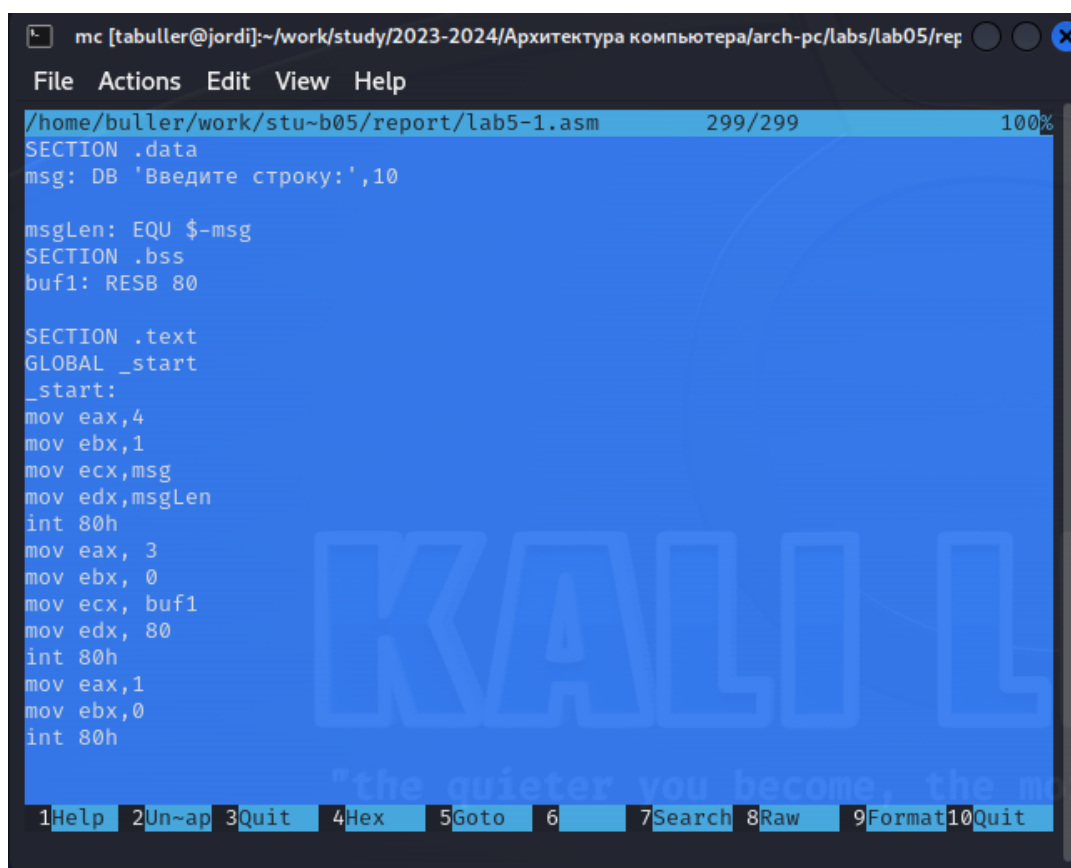
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

Рис. 2.3: Введение текста программы из листинга

Текст программы скопирован, комментарии удалены. Изменения сохранены. Работает.

2.4 Задание 4:

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



```
mc [tabuller@jordi]:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05/rep
File Actions Edit View Help
/home/buller/work/stu~b05/report/lab5-1.asm 299/299 100%
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
int 80h
mov eax,1
mov ebx,0
int 80h

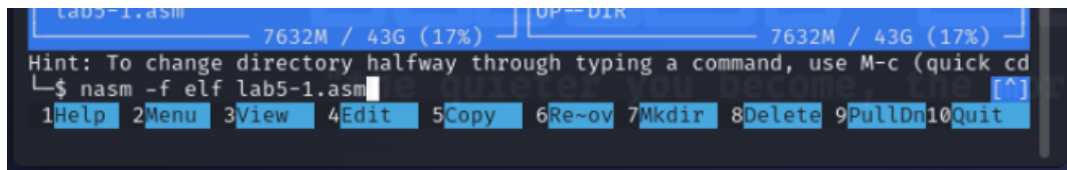
1Help 2Un-ap 3Quit 4Hex 5Goto 6 7Search 8Raw 9Format10Quit
```

Рис. 2.4: Тот же самый текст, но теперь открытый в Midnight Commander

Символ-в-символ тот же самый текст, что был введен и до этого. Сохранение успешно.

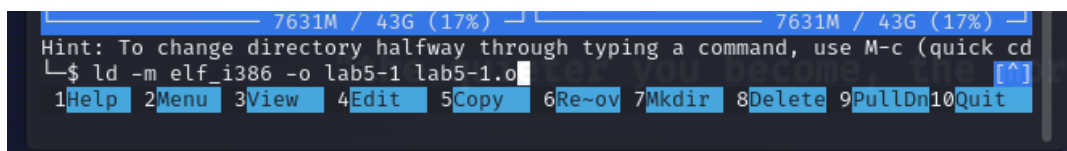
2.5 Задание 5:

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.



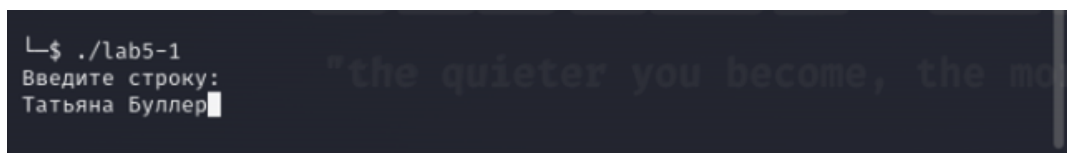
```
lab5-1.asm 7632M / 43G (17%) UP=DIR 7632M / 43G (17%)
Hint: To change directory halfway through typing a command, use M-c (quick cd
L$ nasm -f elf lab5-1.asm
1Help 2Menu 3View 4Edit 5Copy 6Re-ov 7Mkdir 8Delete 9PullDn10Quit
```

Рис. 2.5: Превращение текста программы в объектный код



```
7631M / 43G (17%) 7631M / 43G (17%)
Hint: To change directory halfway through typing a command, use M-c (quick cd
L$ ld -m elf_i386 -o lab5-1 lab5-1.o
1Help 2Menu 3View 4Edit 5Copy 6Re-ov 7Mkdir 8Delete 9PullDn10Quit
```

Рис. 2.6: Создание нового объектного файла и файла листинга программы



```
L$ ./lab5-1
Введите строку:
Татьяна Буллер
```

Рис. 2.7: Выполнение программы

Программа скомпилировалась и выполнилась успешно: попросила меня ввести строку и закрылась сразу же после ввода.

2.6 Задание 6:

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`.

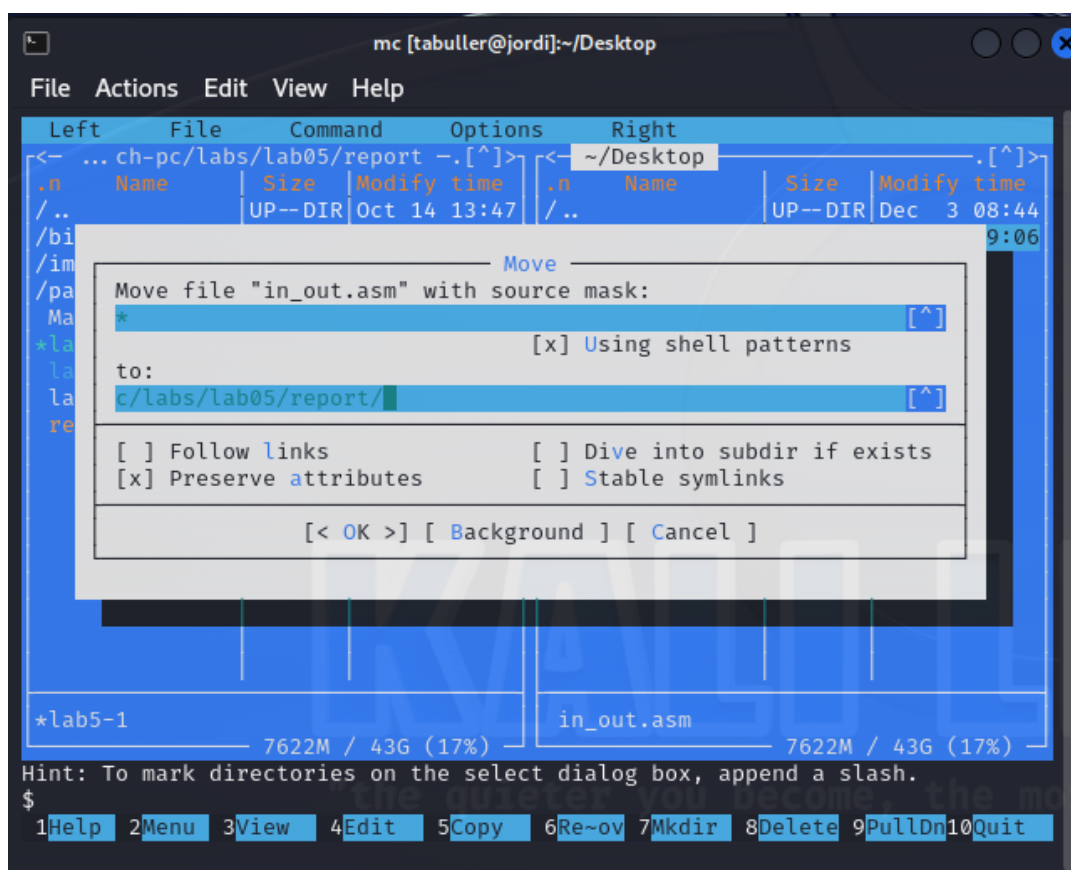


Рис. 2.8: Копирование файла

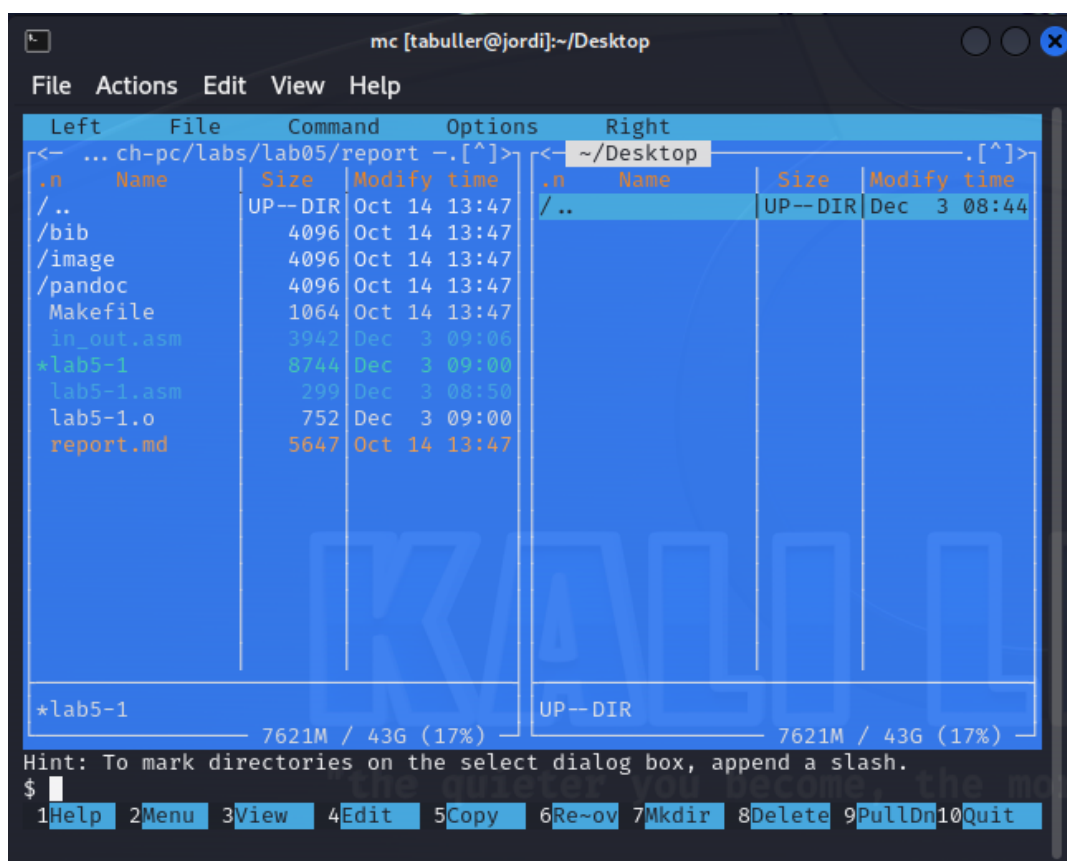


Рис. 2.9: Файл скопирован

Файл, лежавший до этого на рабочем столе, исчез оттуда после копирования и появился в директории лабораторной работы: копирование выполнено успешно.

2.7 Задание 7:

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

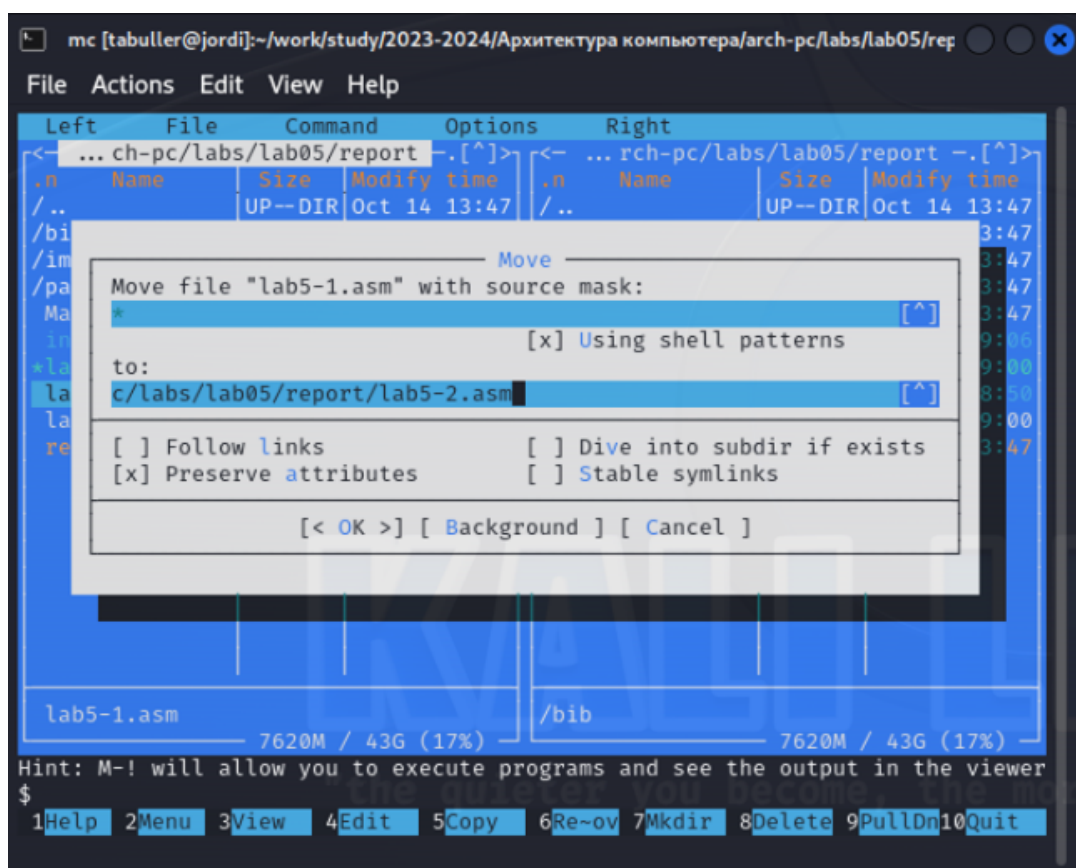
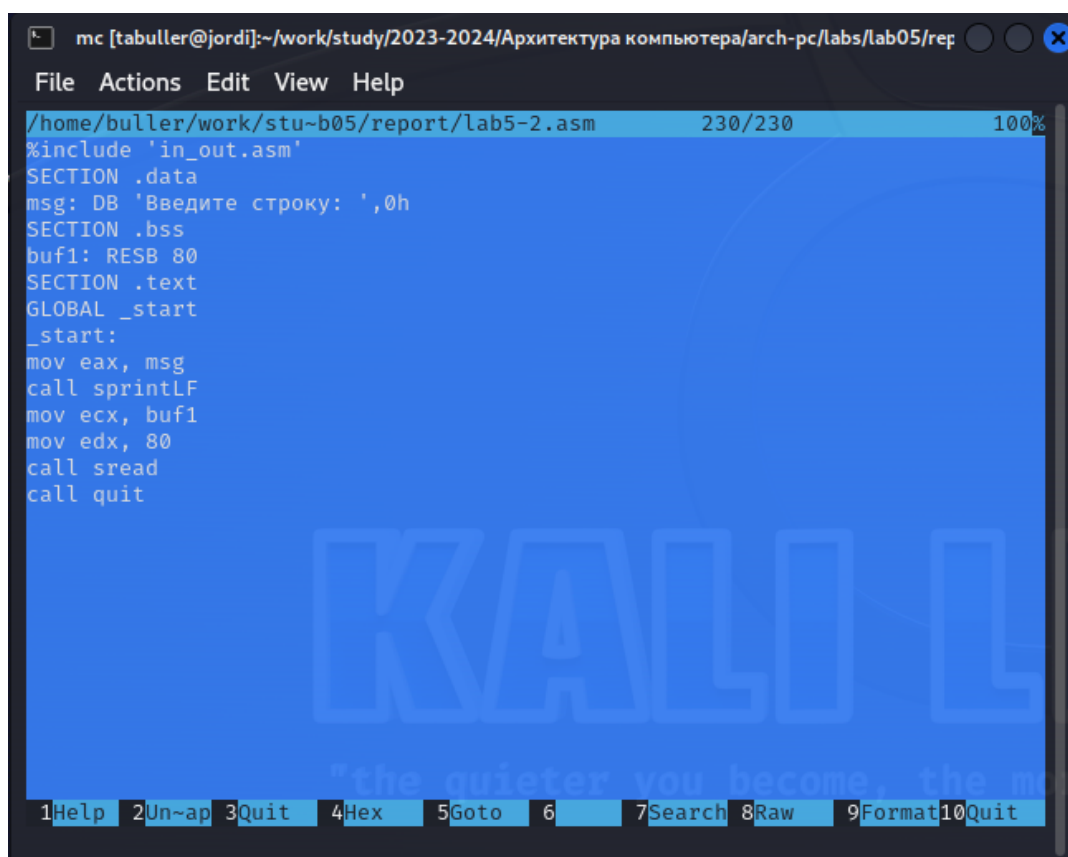


Рис. 2.10: Создание копии файла

2.8 Задание 8:

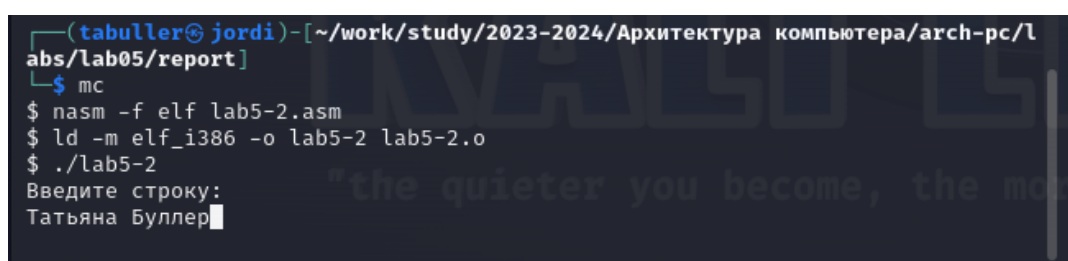
Исправьте текст программы в файле `lab5-2.asm` с использованием подпрограмм из внешнего файла `in_out.asm` (используйте подпрограммы `sprintf`, `sread` и `quit`) в соответствии с листингом 5.2. Создайте исполняемый файл и проверьте его работу



```
mc [tabuller@jordi]:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05/rep
File Actions Edit View Help
/home/buller/work/stu-b05/report/lab5-2.asm 230/230 100%
%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. 2.11: Исправленный код программы

В начало файла указана команда на подключение внешнего файла `in_out.asm`, в котором описаны подпрограммы `sprintLF`, `sread` и `quit`. Они вызываются далее в тексте программы, заменяя часть команд `mov`.



```
(tabuller@jordi)-[~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05/report]
$ mc
$ nasm -f elf lab5-2.asm
$ ld -m elf_i386 -o lab5-2 lab5-2.o
$ ./lab5-2
Введите строку:
Татьяна Буллер
```

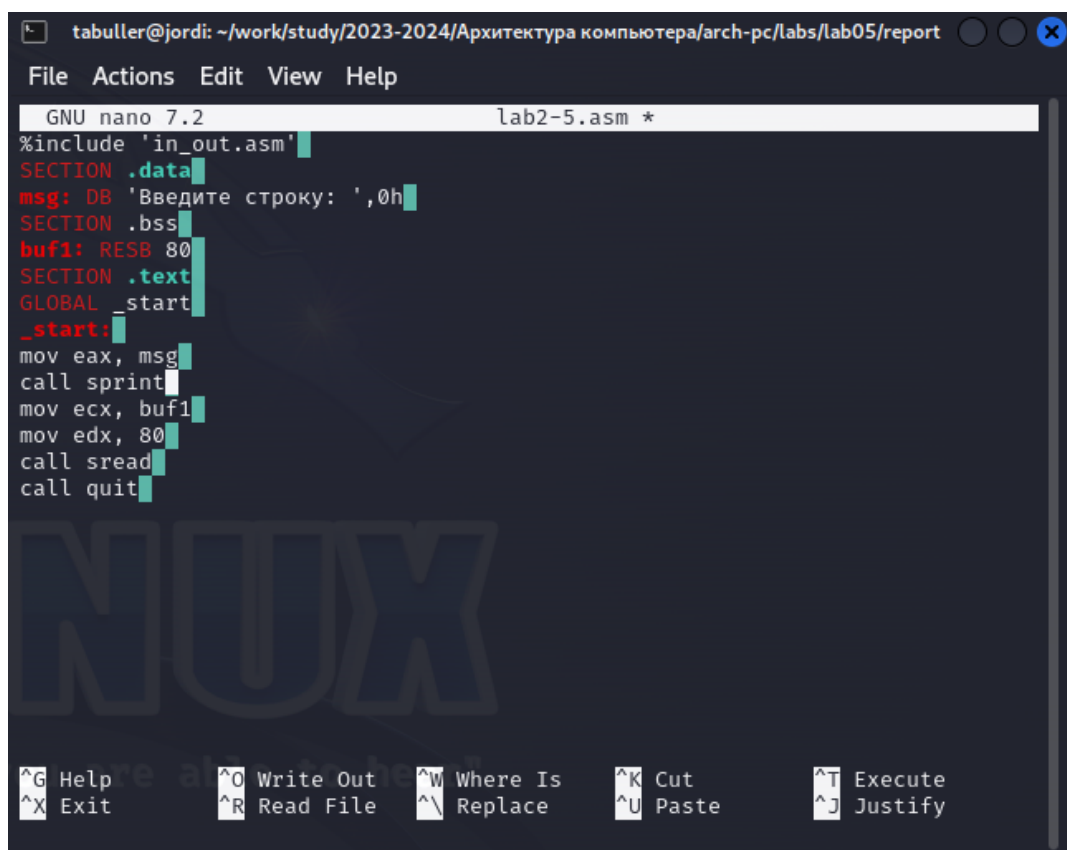
Рис. 2.12: Компиляция программы

Программа скомпилировалась успешно и снова попросила меня ввести строку,

после чего завершилась.

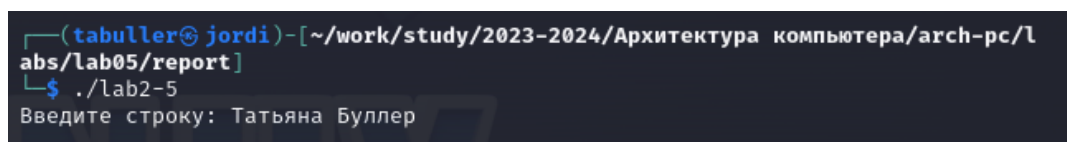
2.9 Задание 9:

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



```
tabuller@jordi: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05/report
File Actions Edit View Help
GNU nano 7.2 lab2-5.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
call quit
```

Рис. 2.13: Исправленный код программы



```
(tabuller@jordi)-[~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05/report]
$ ./lab2-5
Введите строку: Татьяна Буллер
```

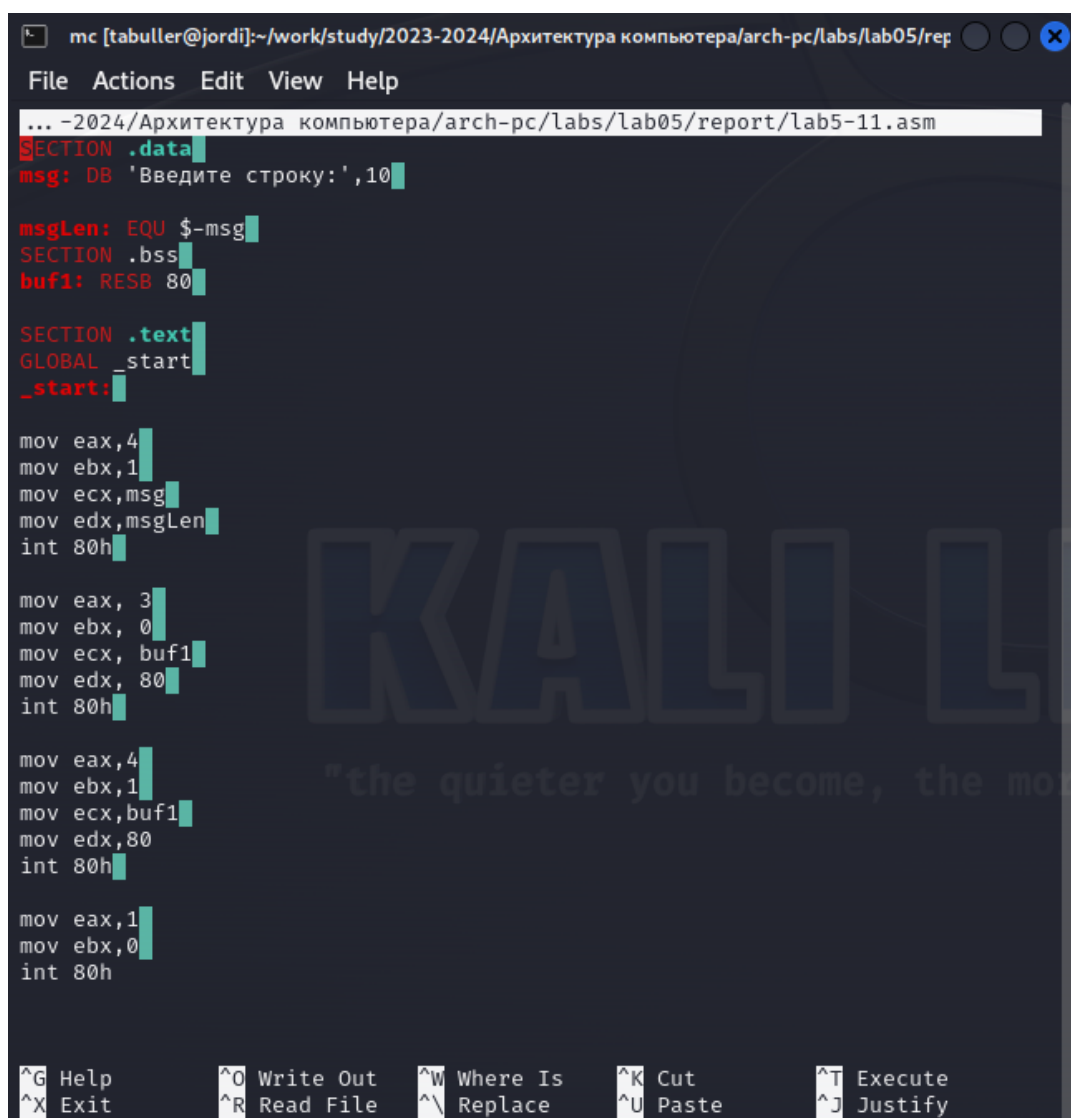
Рис. 2.14: Компиляция обновленной программы

Программа скомпилировалась без значительных изменений, но замена `sprintf` на `sprint` привела к тому, что исчез символ переноса строки при выводе сообщения на экран.

3 Задание для самостоятельной работы

3.1 Задание 10:

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 введенную строку на экран. Получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите свою фамилию.



```
mc [tabuller@jordi]:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05/rep
File Actions Edit View Help
... -2024/Архитектура компьютера/arch-pc/labs/lab05/report/lab5-11.asm
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

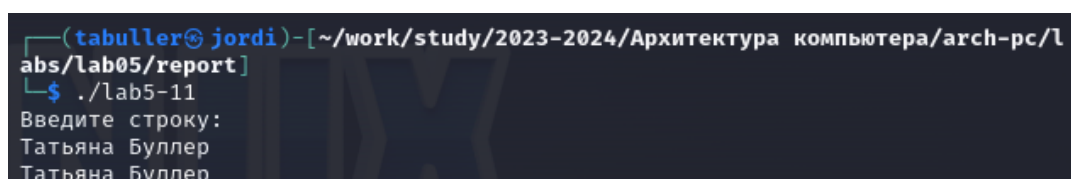
mov eax,3
mov ebx,0
mov ecx,buf1
mov edx,80
int 80h

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

mov eax,1
mov ebx,0
int 80h

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

Рис. 3.1: Исправленный код программы



```
(tabuller@jordi)-[~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05/report]
$ ./lab5-11
Введите строку:
Татьяна Буллер
Татьяна Буллер
```

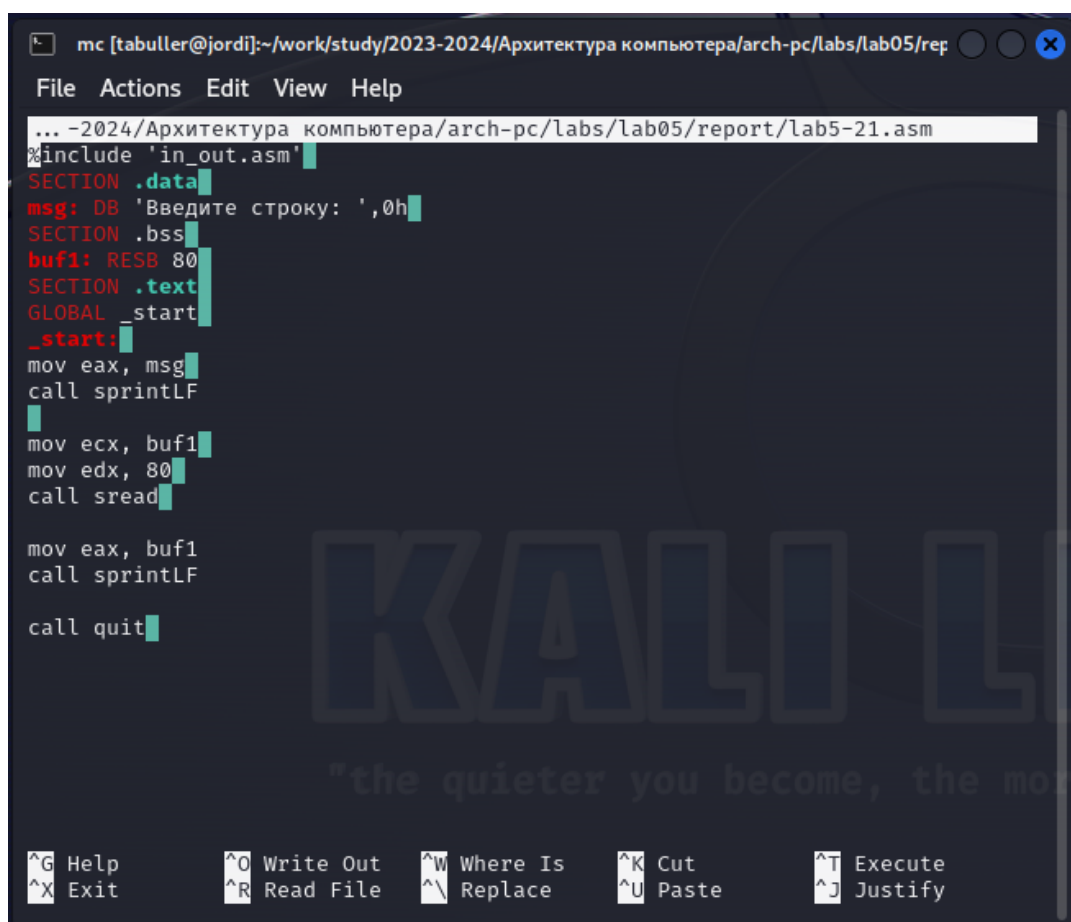
Рис. 3.2: Компиляция обновленной программы

Для того, чтобы после введения строки полученный ввод был показан еще раз, скопируем блок вывода текста за одним исправлением: вместо переменной msg

теперь нужно вывести полученный ввод (он сохраняется в переменную `buf1`). при компиляции программы можно видеть, что это работает именно так, как и было задумано.

3.2 Задание 11:

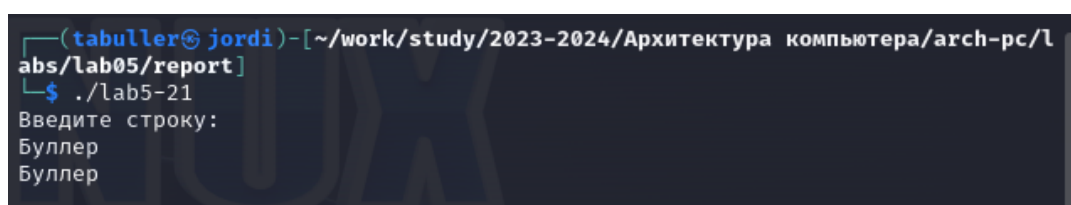
Создайте копию файла `lab5-2.asm`. Исправьте текст программы с использование подпрограмм из внешнего файла `in_out.asm`, так чтобы она работала по следующему алгоритму: * вывести приглашение типа “Введите строку:”; * ввести строку с клавиатуры; * вывести введённую строку на экран. Получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите свою фамилию.



```
mc [tabuller@jordi]:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05/rep
File Actions Edit View Help
... -2024/Архитектура компьютера/arch-pc/labs/lab05/report/lab5-21.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 sprintf
mov ecx, buf1
mov edx, 80
call sread

mov eax, buf1
call sprintf
call quit
```

Рис. 3.3: Исправленный код программы



```
(tabuller@jordi)-[~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05/report]
$ ./lab5-21
Введите строку:
Буллер
Буллер
```

Рис. 3.4: Компиляция обновленной программы

Для того, чтобы после введения строки полученный ввод был показан еще раз, в этот раз просто еще один раз вызовем команду `sprintf`. При компиляции программы можно видеть, что это работает именно так, как и было задумано.

4 Вывод

При выполнении лабораторной работы были приобретены практические навыки работы в Midnight Commander и освоены инструкции языка ассемблера `mov` и `int`.