

Лабораторная работа №5

Архитектура вычислительных систем

Чулкова Валерия

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	6
4	Выполнение лабораторной работы	7
5	Самостоятельная работа	14
6	Выводы	17
	Список литературы	18

Список иллюстраций

4.1	mc	7
4.2	mc	8
4.3	mc	9
4.4	mc	10
4.5	mc	11
4.6	mc	11
4.7	mc	12
4.8	mc	13
5.1	mc	14
5.2	mc	15
5.3	mc	16
5.4	mc	16
5.5	mc	16

1 Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

2 Задание

1. Создайте копию файла lab6-1.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.
2. Получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите свою фамилию.
3. Создайте копию файла lab6-2.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.
4. Создайте исполняемый файл и проверьте его работу.

3 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы. Например, в табл. 3.1 приведено краткое описание стандартных каталогов Unix.

Таблица 3.1: Описание некоторых каталогов файловой системы GNU Linux

Имя каталога	Описание каталога
/	Корневая директория, содержащая всю файловую систему
/bin	Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям
/etc	Общесистемные конфигурационные файлы и файлы конфигурации установленных программ
/home	Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя
/media	Точки монтирования для сменных носителей
/root	Домашняя директория пользователя root
/tmp	Временные файлы
/usr	Вторичная иерархия для данных пользователя

Более подробно об Unix см. в [1–6].

4 Выполнение лабораторной работы

1. Открываем Midnight Commander.

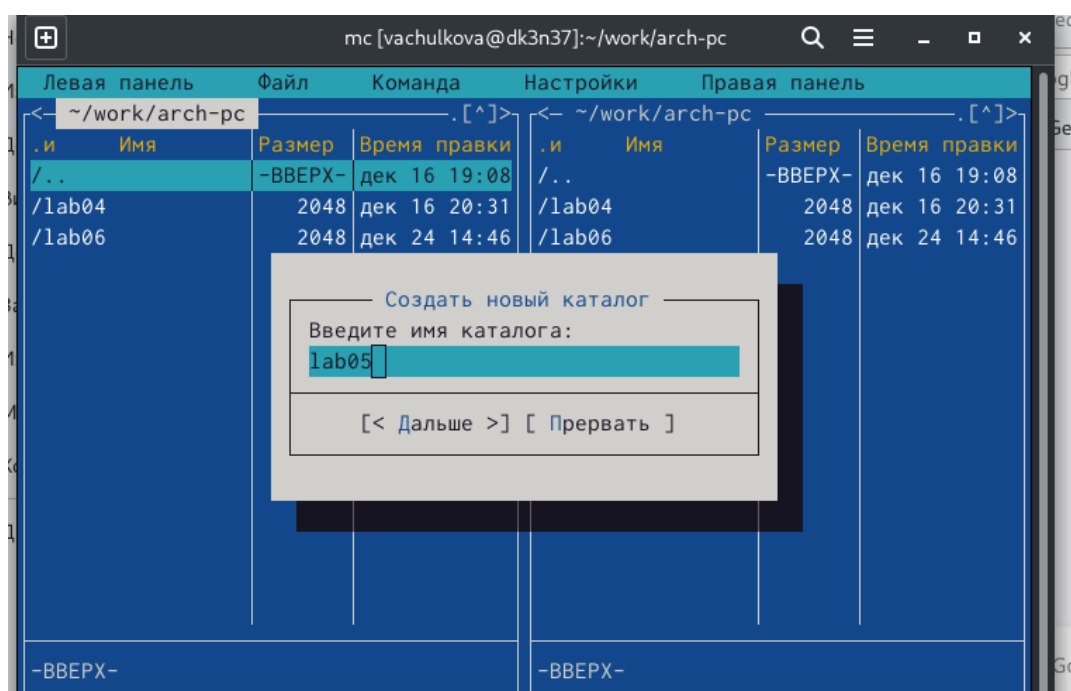


Рис. 4.1: mc

2. Пользуясь клавишами перейдем в каталог `~/work/archpc` созданный при выполнении лабораторной работы №4. С помощью функциональной клавиши F7 создаём папку `lab06` и переходим в созданный каталог

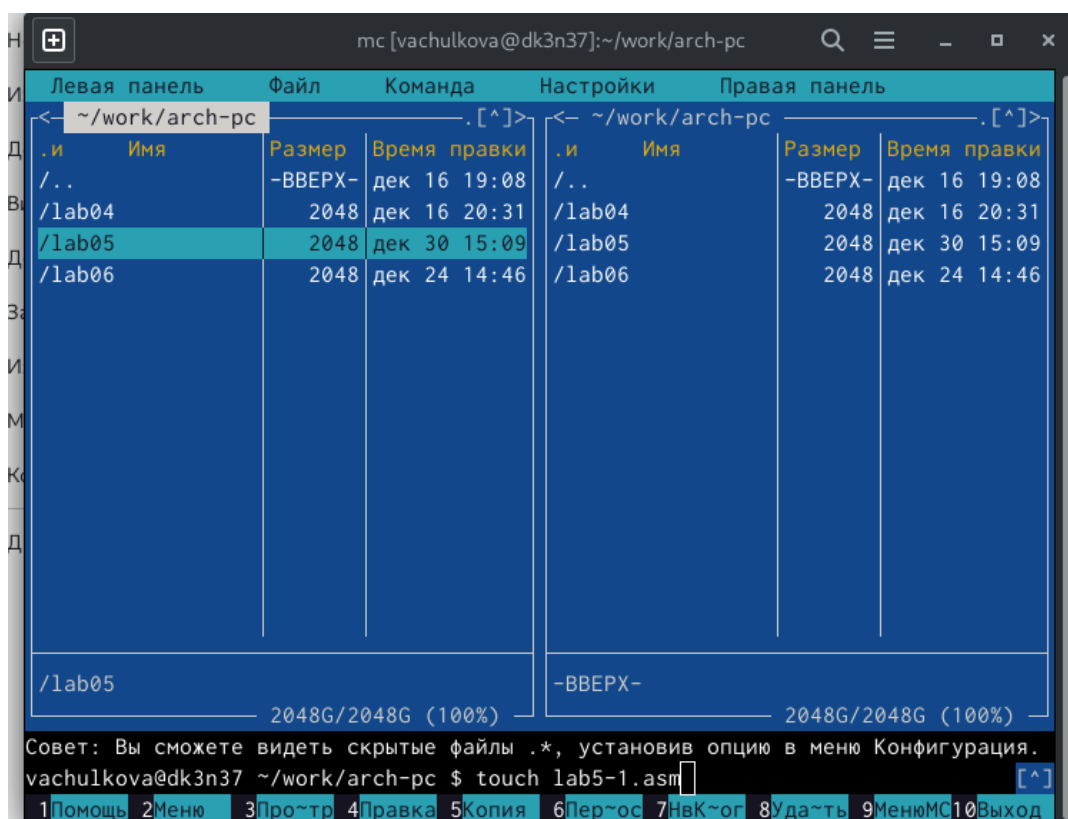
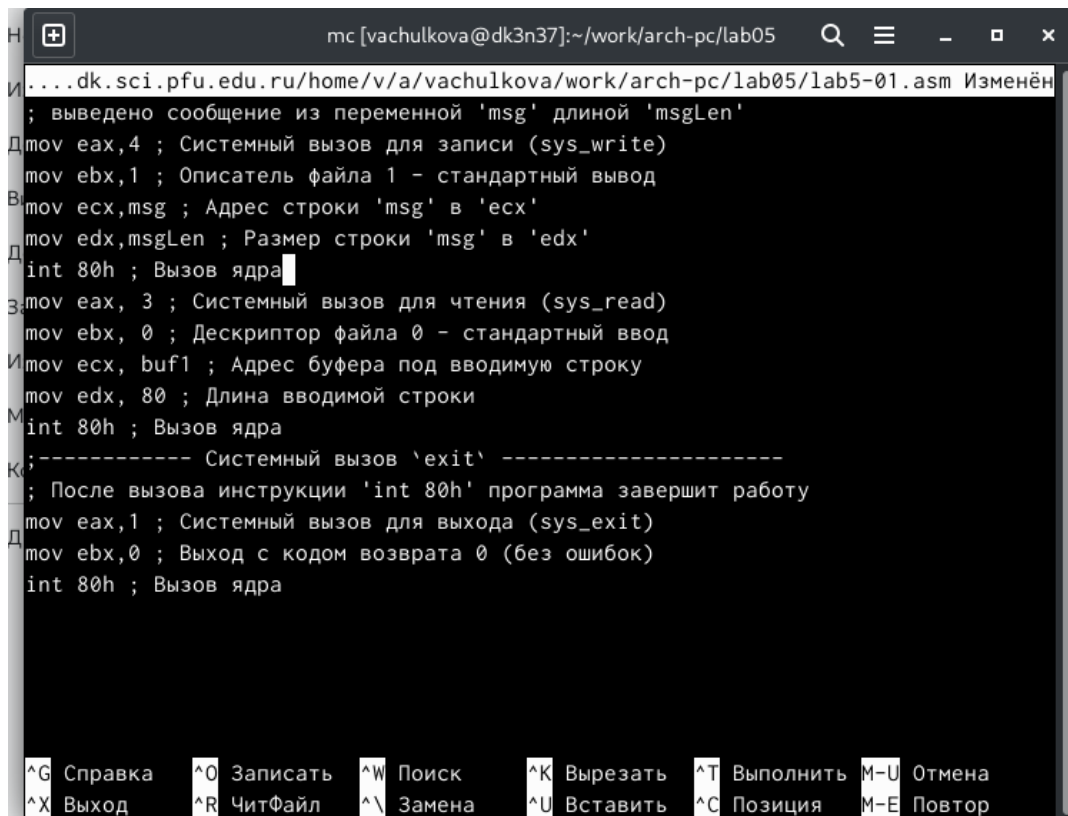


Рис. 4.2: mc

3. Пользуясь строкой ввода и командой `touch` создайте файл `lab6-1.asm`. С помощью функциональной клавиши F4 откроем файл `lab5-1.asm` для редактирования во встроенном редакторе. Как правило в качестве встроенного редактора Midnight Commander используется редакторы `nano` или `mcedit`. Вводим текст программы из листинга 5.1, сохраняем изменения и закрываем файл. С помощью функциональной клавиши F3 открываем файл `lab5-1.asm` для просмотра.



```
...dk.sci.pfu.edu.ru/home/v/a/vachulkova/work/arch-pc/lab05/lab5-01.asm Изменён
; выведено сообщение из переменной 'msg' длиной 'msgLen'
Д mov eax,4 ; Системный вызов для записи (sys_write)
В mov ebx,1 ; Описатель файла 1 - стандартный вывод
Д mov ecx,msg ; Адрес строки 'msg' в 'ecx'
Д mov edx,msgLen ; Размер строки 'msg' в 'edx'
Д int 80h ; Вызов ядра
З mov eax, 3 ; Системный вызов для чтения (sys_read)
И mov ebx, 0 ; Дескриптор файла 0 - стандартный ввод
И mov ecx, buf1 ; Адрес буфера под вводимую строку
М mov edx, 80 ; Длина вводимой строки
М int 80h ; Вызов ядра
К ;----- Системный вызов 'exit' -----
; После вызова инструкции 'int 80h' программа завершит работу
Д mov eax,1 ; Системный вызов для выхода (sys_exit)
Д mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
Д int 80h ; Вызов ядра

^G Справка    ^O Записать   ^W Поиск      ^K Вырезать   ^T Выполнить  M-U Отмена
^X Выход      ^R ЧитФайл   ^\ Замена     ^U Вставить   ^C Позиция    M-E Повтор
```

Рис. 4.3: mc

4. С помощью функциональной клавиши F4 откройте файл lab5-1.asm для редактирования во встроенном редакторе. Как правило в качестве встроенного редактора Midnight Commander используется редакторы nano или mcedit . Вводим текст программы из листинга 5.1 , сохраняем изменения и закрываем файл. С помощью функциональной клавиши F3 открываем файл lab5-1.asm для просмотра. Убедимся, что файл содержит текст программы.

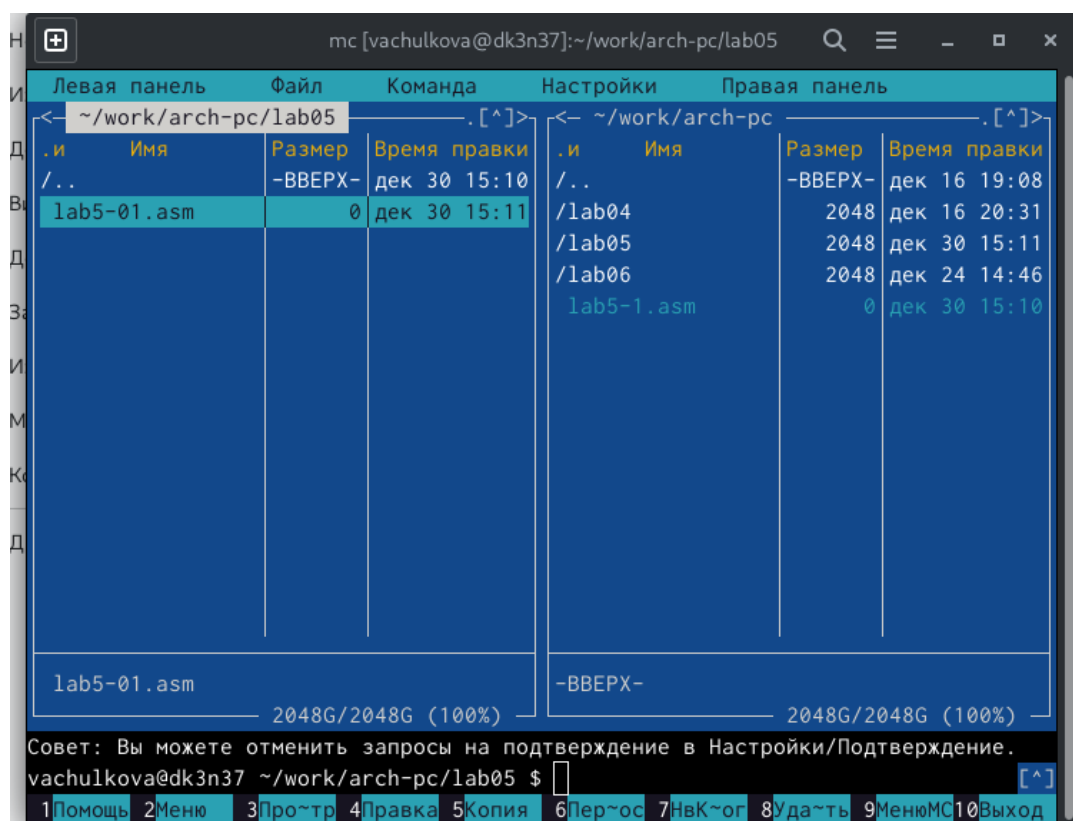
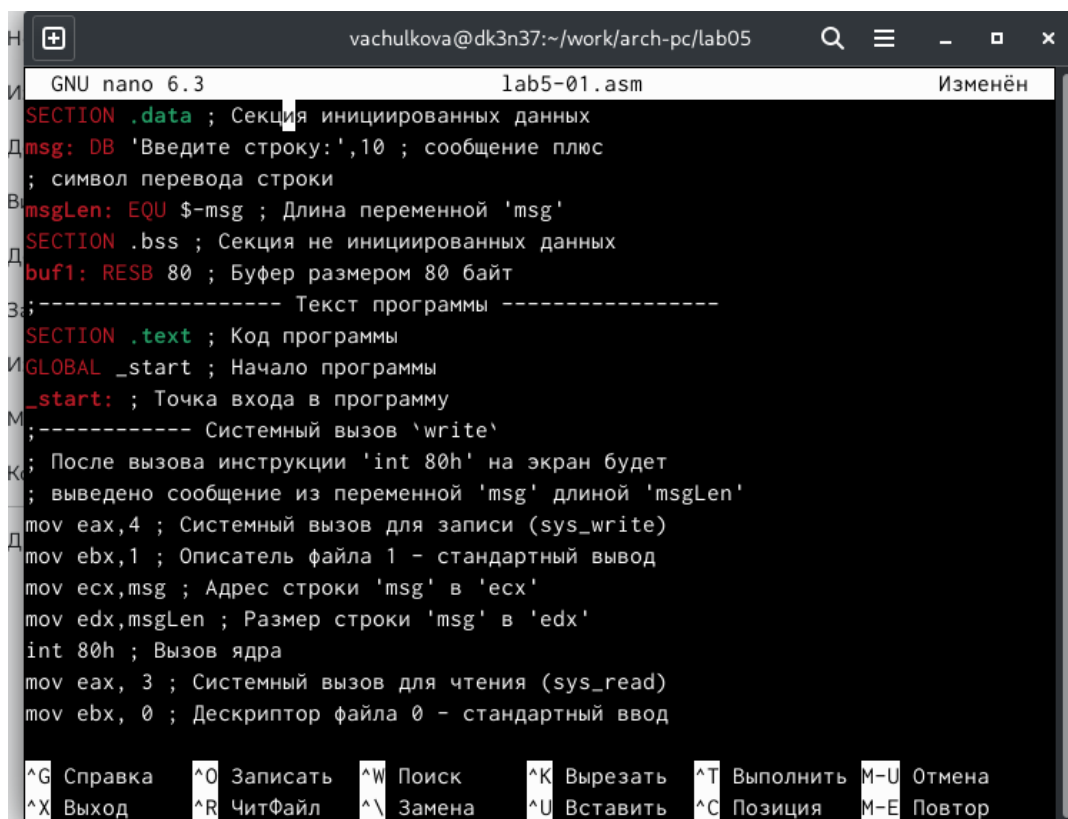


Рис. 4.4: mc

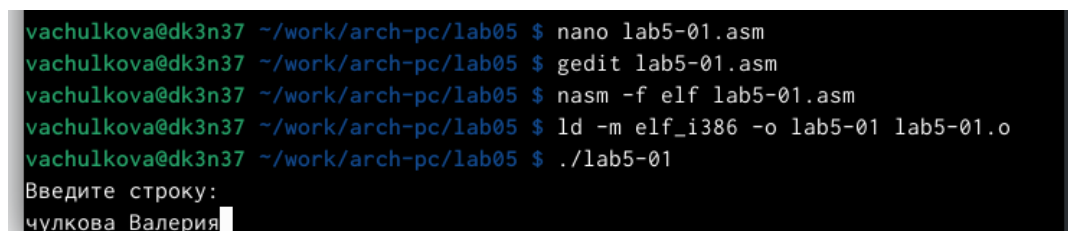


```
GNU nano 6.3 lab5-01.asm Изменён
SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не иницированных данных
buf1: RESB 80 ; Буфер размером 80 байт
;----- Текст программы -----
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
mov eax, 3 ; Системный вызов для чтения (sys_read)
mov ebx, 0 ; Дескриптор файла 0 - стандартный ввод

^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить M-U Отмена
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^C Позиция M-E Повтор
```

Рис. 4.5: mc

5. 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 ввожу свои ФИО.

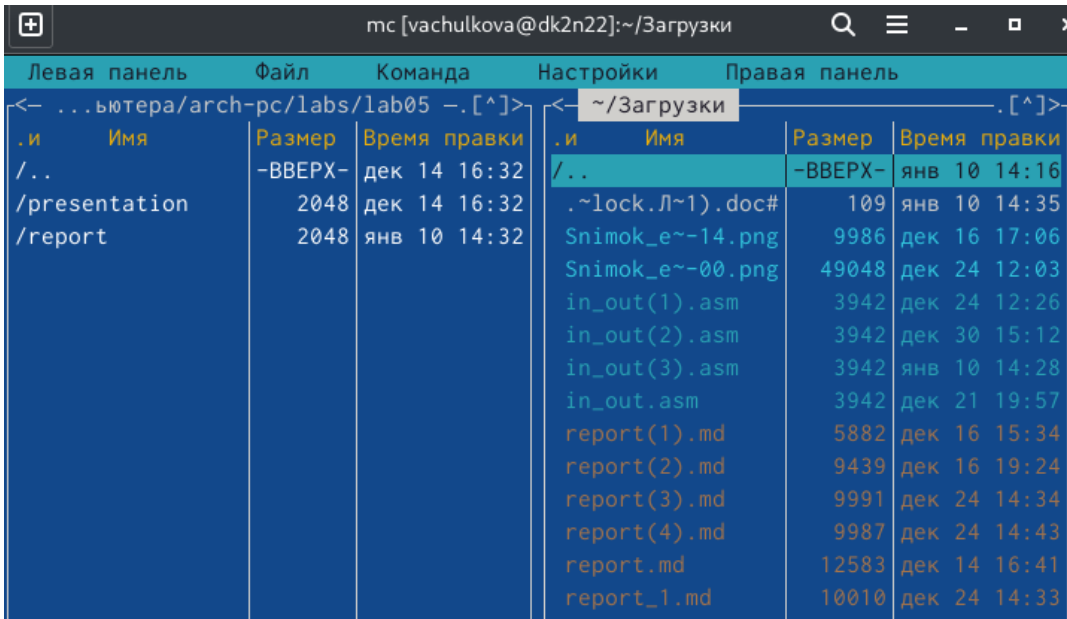


```
vachulkova@dk3n37 ~/work/arch-pc/lab05 $ nano lab5-01.asm
vachulkova@dk3n37 ~/work/arch-pc/lab05 $ gedit lab5-01.asm
vachulkova@dk3n37 ~/work/arch-pc/lab05 $ nasm -f elf lab5-01.asm
vachulkova@dk3n37 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-01 lab5-01.o
vachulkova@dk3n37 ~/work/arch-pc/lab05 $ ./lab5-01
Введите строку:
чулкова Валерия
```

Рис. 4.6: mc

5. Скачиваю файл in_out.asm со страницы курса в ТУИС. 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.



Левая панель			Правая панель		
Файл	Команда	Настройки	Файл	Команда	Настройки
<- ...ьютера/arch-pc/labs/lab05 -. [^]>			<- ~/Загрузки -. [^]>		
.и	Имя	Размер	Время правки	.и	Имя
/..	-ВВЕРХ-	дек 14 16:32		/..	-ВВЕРХ-
/presentation	2048	дек 14 16:32		..lock.Л~1).doc#	109
/report	2048	янв 10 14:32		Snimok_e~-14.png	9986
				Snimok_e~-00.png	49048
				in_out(1).asm	3942
				in_out(2).asm	3942
				in_out(3).asm	3942
				in_out.asm	3942
				report(1).md	5882
				report(2).md	9439
				report(3).md	9991
				report(4).md	9987
				report.md	12583
				report_1.md	10010

Рис. 4.7: mc

Исправляю текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in_out.asm в соответствии с листингом 5.2. Создаю исполняемый файл и проверяю его работу.

```
%include 'in_out.asm'
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 sprintf ; 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
```

Имя файла для записи: <c/lab05/lab5-2.asm

^G Справка	M-D Формат DOS	M-A Доп. в начало	M-B Резерв. копия
^C Отмена	M-M Формат Mac	M-P Доп. в конец	^T Обзор

Рис. 4.8: mc

5 Самостоятельная работа

1. 1. Создаем копию файла lab5-1.asm. Внесем изменения в программу), так чтобы
 - вывести приглашение типа “Введите строку:”;
 - ввести строку с клавиатуры;
 - вывести введенную строку на экран. Копируем файл.

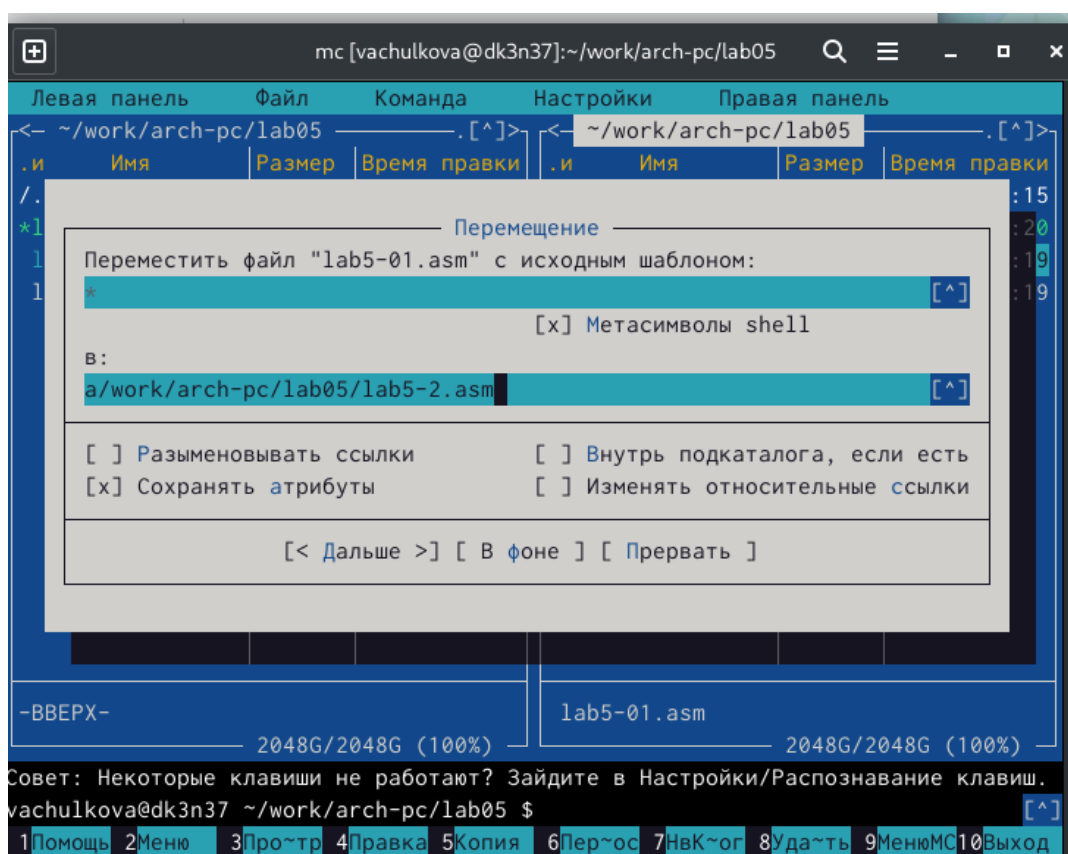
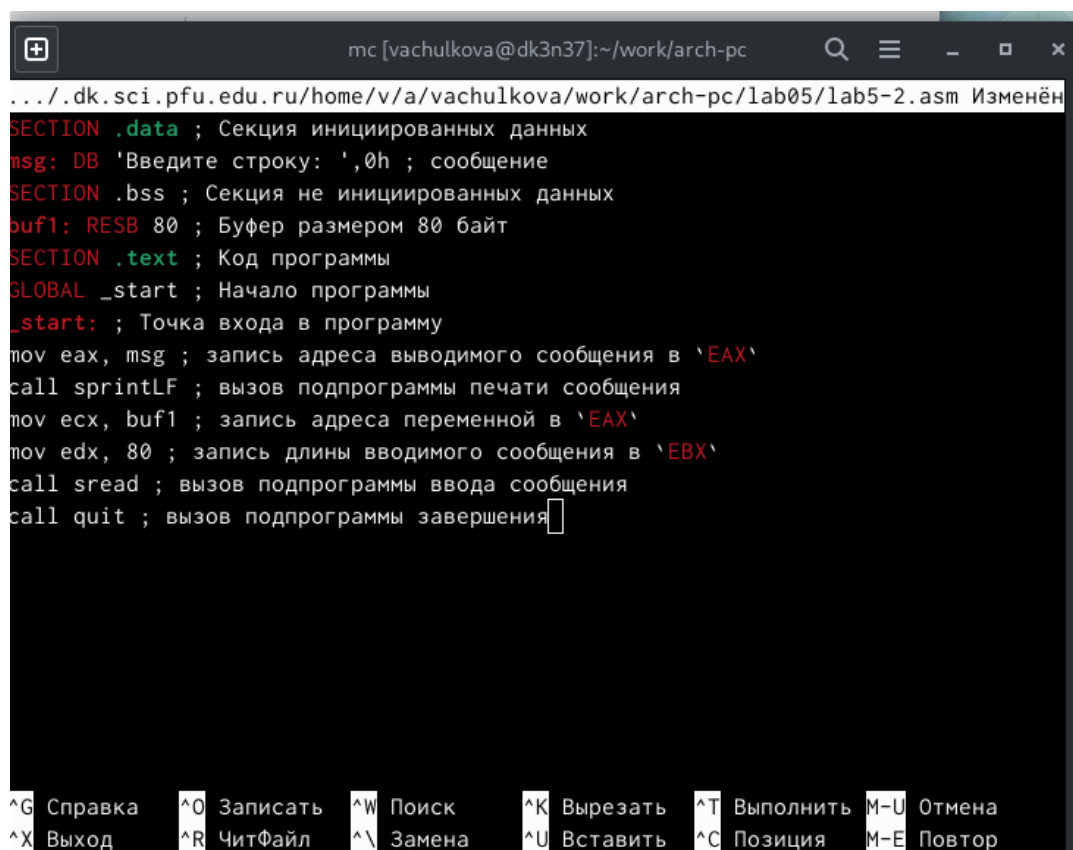


Рис. 5.1: mc

2. Исходный код

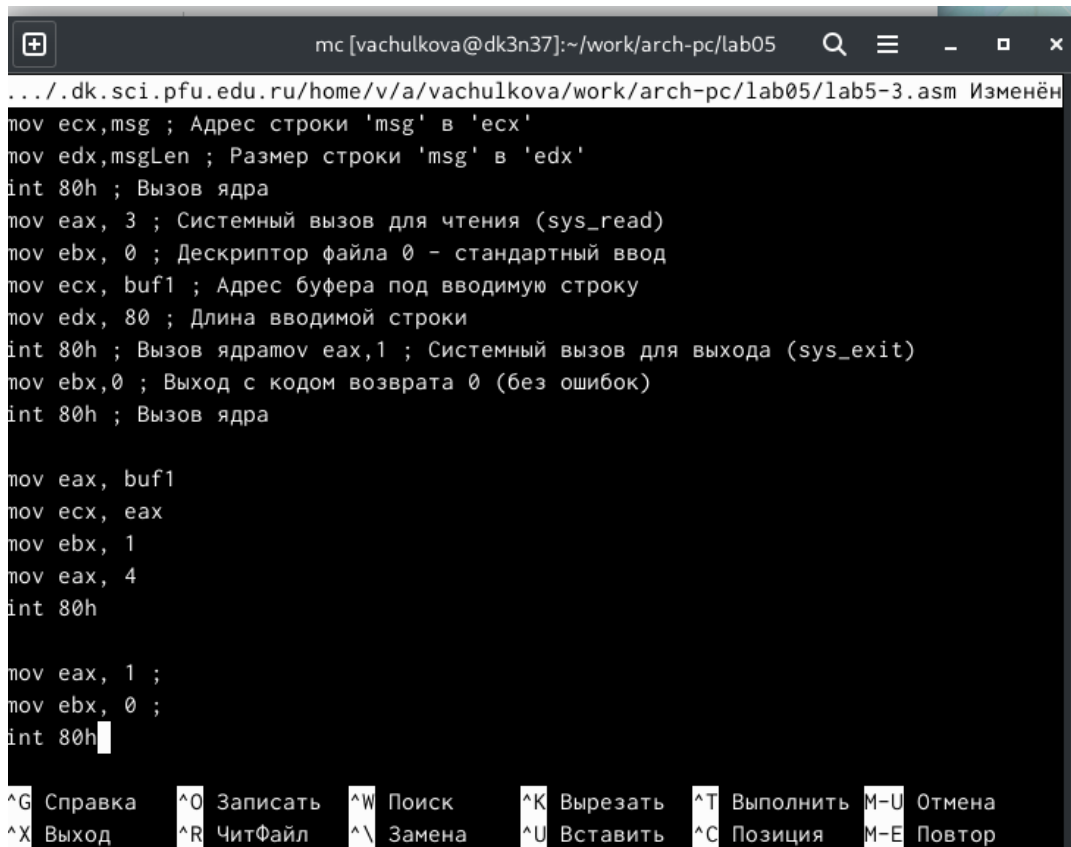


```
.../.dk.sci.pfu.edu.ru/home/v/a/vachulkova/work/arch-pc/lab05/lab5-2.asm Изменён
SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не иницированных данных
buf1: RESB 80 ; Буфер размером 80 байт
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
mov eax, msg ; запись адреса выводимого сообщения в `EAX`
call sprintf ; 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 M-U Отмена
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^C Позиция M-E Повтор

Рис. 5.2: mc

3. Дописываем 4 строки после `call sread` вызывающие функцию `sprintf`. Создаем исполняемый файл и проверьте его работу.



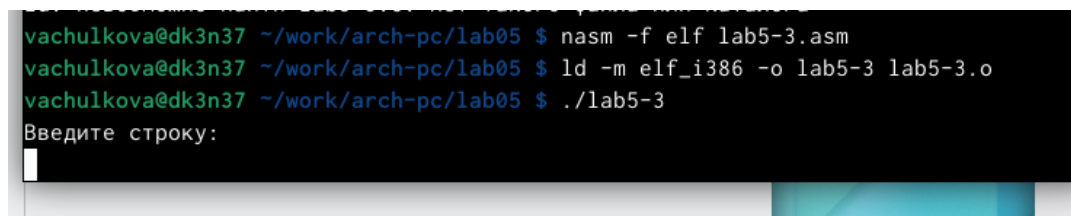
```
.../.dk.sci.pfu.edu.ru/home/v/a/vachulkova/work/arch-pc/lab05/lab5-3.asm Изменён
mov ecx,msg ; Адрес строки 'msg' в 'ecx'
mov edx,msgLen ; Размер строки 'msg' в 'edx'
int 80h ; Вызов ядра
mov eax, 3 ; Системный вызов для чтения (sys_read)
mov ebx, 0 ;Descriptor файла 0 - стандартный ввод
mov ecx, buf1 ; Адрес буфера под вводимую строку
mov edx, 80 ; Длина вводимой строки
int 80h ; Вызов ядра
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра

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

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

^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить M-U Отмена
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^C Позиция M-E Повтор

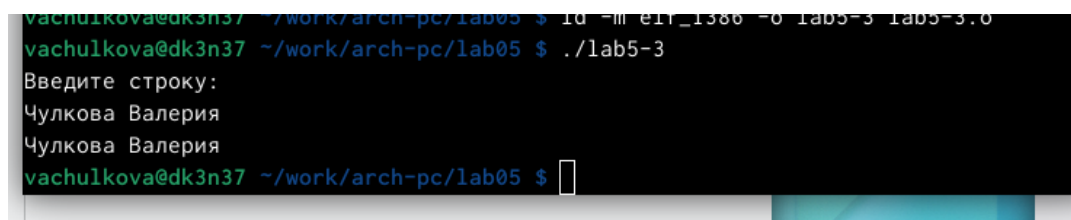
Рис. 5.3: mc



```
vachulkova@dk3n37 ~/work/arch-pc/lab05 $ nasm -f elf lab5-3.asm
vachulkova@dk3n37 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-3 lab5-3.o
vachulkova@dk3n37 ~/work/arch-pc/lab05 $ ./lab5-3
Введите строку:

```

Рис. 5.4: mc



```
vachulkova@dk3n37 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-3 lab5-3.o
vachulkova@dk3n37 ~/work/arch-pc/lab05 $ ./lab5-3
Введите строку:
Чулкова Валерия
Чулкова Валерия
vachulkova@dk3n37 ~/work/arch-pc/lab05 $
```

Рис. 5.5: mc

6 Выводы

В ходе выполнения данной лабораторной работы были приобретены практические навыки работы в Midnight Commander. Были освоены инструкции языка ассемблера `mov` и `int`.

Список литературы

::: {#refs}:

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016.
URL: <https://www.gnu.org/software/bash/manual/>.
2. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.