

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: Волчкова Елизавета

Дмитриевна

Группа: НКАбд-01-24

МОСКВА

2024 г.

Введение.

1.	Теоретическое введение	3
2.	Цель работы	4
3.	Задание самостоятельной и лабораторной работы	5
4.	Выполнение самостоятельной и лабораторной работы	6
5.	Вывод	
6.	Список литературы	
7.		

Теоретическое введение.

Основы работы с Midnight Commander Midnight Commander (или просто mc) — это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управлению файловой системой, т.е. mc является файловым менеджером.

Midnight Commander позволяет сделать работу с файлами более удобной и наглядной.

Для активации оболочки Midnight Commander достаточно ввести в командной строке mc и нажать клавишу Enter.

В Midnight Commander используются функциональные клавиши F1 — F10, к которым привязаны часто выполняемые операции.

Цель работы.

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

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 введенную строку на экран.

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

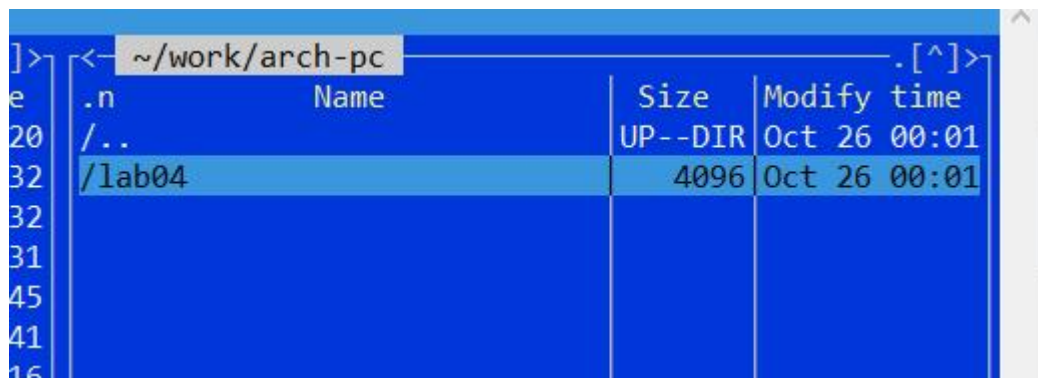
3. Создайте копию файла lab5-2.asm. Исправьте текст программы с использованием подпрограмм из внешнего файла in_out.asm, так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введенную строку на экран. Не забудьте, подключаемый файл in_out.asm должен лежать в том же каталоге, что и файл с программой, в которой он используется.

4. Создайте исполняемый файл и проверьте его работу.

Выполнение самостоятельной и лабораторной работ.

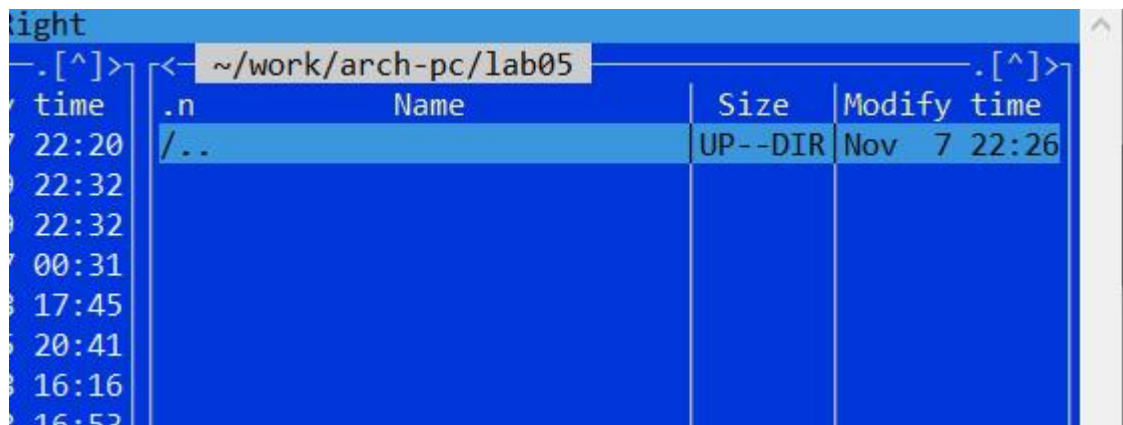
Порядок выполнения лабораторной работы.

1. Сначала открыла Midnight Commander user@dk4n31:~\$ mc
2. Далее я, пользуясь клавишами ↑, ↓ и Enter перешла в каталог ~/work/arch-pc созданный при выполнении лабораторной работы №4.



	.n	Name	Size	Modify time
20		/..	UP--DIR	Oct 26 00:01
32		/lab04	4096	Oct 26 00:01
32				
31				
45				
41				
16				

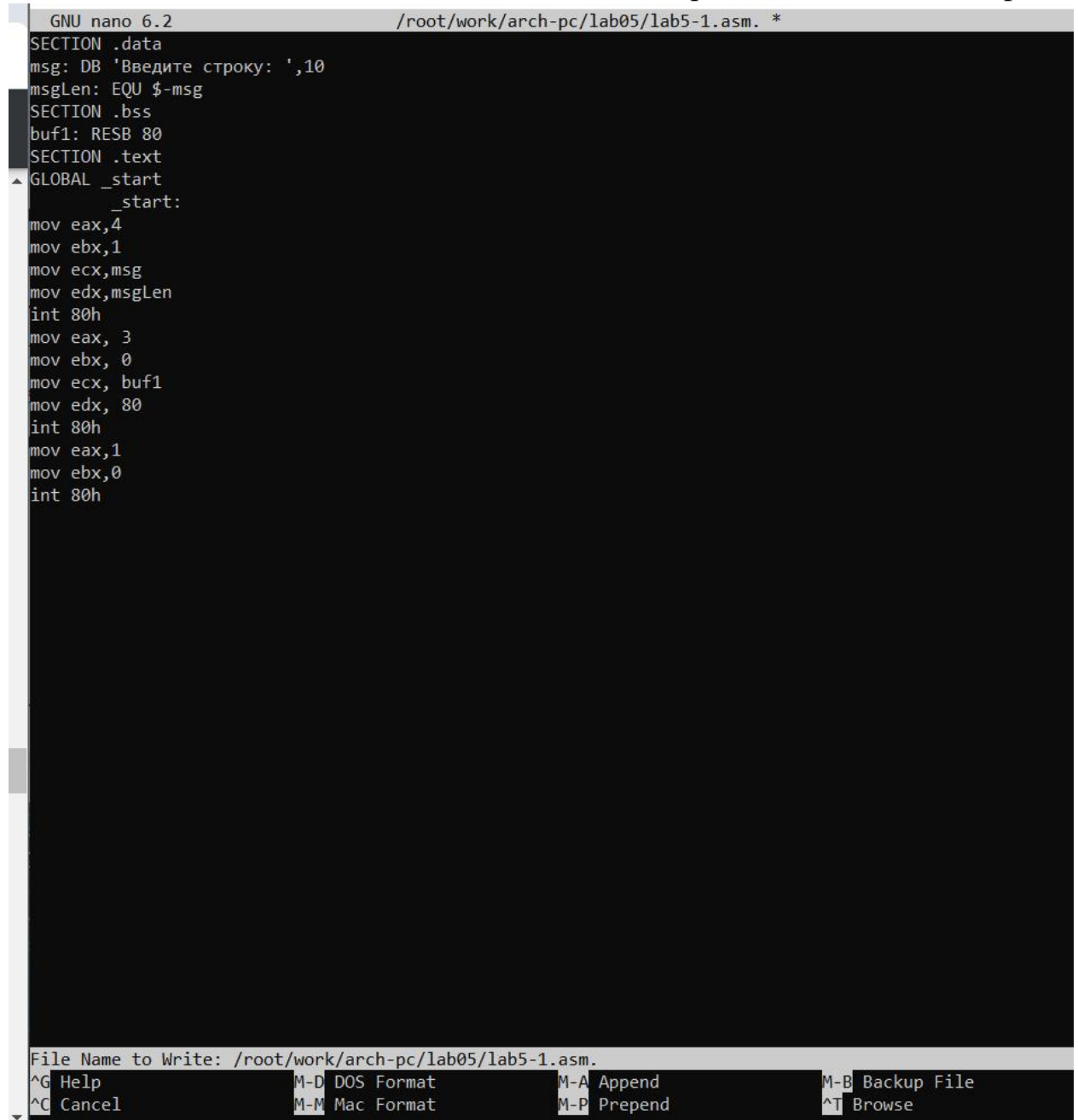
3. С помощью функциональной клавиши F7 создала папку lab05 и перешла в созданный каталог.



	.n	Name	Size	Modify time
22:20		/..	UP--DIR	Nov 7 22:26
22:32				
22:32				
00:31				
17:45				
20:41				
16:16				
16:53				

4. Потом, пользуясь строкой ввода и командой touch создала lab5-1.asm.
5. Затем с помощью функциональной клавиши F4 открыла файл lab5-1.asm для редактирования во встроенном редакторе.

6. После ввела текст программы из листинга 5.1, затем сохранила изменения и закрыла файл.



```
GNU nano 6.2 /root/work/arch-pc/lab05/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

File Name to Write: /root/work/arch-pc/lab05/lab5-1.asm.
^G Help      M-D DOS Format  M-A Append     M-B Backup File
^C Cancel    M-M Mac Format  M-P Prepend    ^T Browse
```

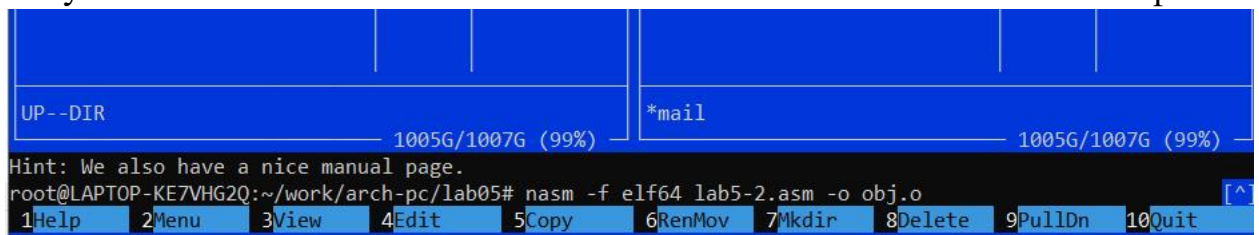
7. После этого с помощью функциональной клавиши F3 открыла файл lab5-1.asm для просмотра и убедилась, что файл содержит текст программы.

```
/root/work/arch-pc/lab05/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
```

8. Оттранслировала текст программы lab5-1.asm в объектный файл.

Далее я выполнила компоновку объектного файла и запустила получившийся исполняемый файл.




```
Left      File      Command      Options      Right
<-- ~      .[^]>      <-- ~/work/arch-pc/lab05      .[^]>
Name      Size      Modify time      Name      Size      Modify time
UP--DIR      UP--DIR      UP--DIR      UP--DIR      UP--DIR      UP--DIR
/..      /..      /..      /..      /..      /..
/.cache      4096      Oct 29 22:32      *.in_out.asm      3942      Nov 7 23:15
/.config      4096      Oct 29 22:32      *lab5-1      8984      Nov 7 23:53
/.git      4096      Sep 27 00:31      lab5-2.asm      278      Nov 7 22:45
/.local      4096      Sep 8 17:45      obj.o      1056      Nov 7 23:44
/LABS      4096      Sep 15 20:41
/Liza      4096      Sep 8 16:16
/lab4      4096      Sep 8 16:53
/labs      4096      Sep 15 20:29
/parentdir      4096      Sep 15 22:11
/parentdir2      4096      Sep 15 22:11
/parentdir3      4096      Sep 15 22:25
/sept26      4096      Sep 26 23:21
/snap      4096      Sep 8 14:54
/temp      4096      Sep 15 20:11
/tmp      4096      Sep 8 16:16
/work      4096      Oct 26 00:01
.bash_history      7418      Nov 7 23:47
.bashrc      3106      Oct 15 2021
.gitconfig      148      Sep 27 00:45
.lessht      20      Sep 27 00:12
.motd_shown      0      Nov 7 22:20
.profile      161      Jul 9 2019
.selected_editor      66      Nov 7 22:29
Liza2006      2675      Sep 27 01:10
Liza2006.pub      593      Sep 27 01:10
hello.asm      0      Oct 25 23:25
test1.txt      0      Sep 15 22:24

UP--DIR      1005G/1007G (99%)      *lab5-1      1005G/1007G (99%)

Hint: We also have a nice manual page.
root@LAPTOP-KE7VHG2Q:~/work/arch-pc/lab05# ld -o lab5-1 obj.o
1Help 2Menu 3View 4Edit 5Copy 6RenMov 7Mkdir 8Delete 9PullDn 10Quit
```

Для вызова подпрограммы из внешнего файла использовала инструкцию call, которая имеет следующий вид call, где function - имя подпрограммы.

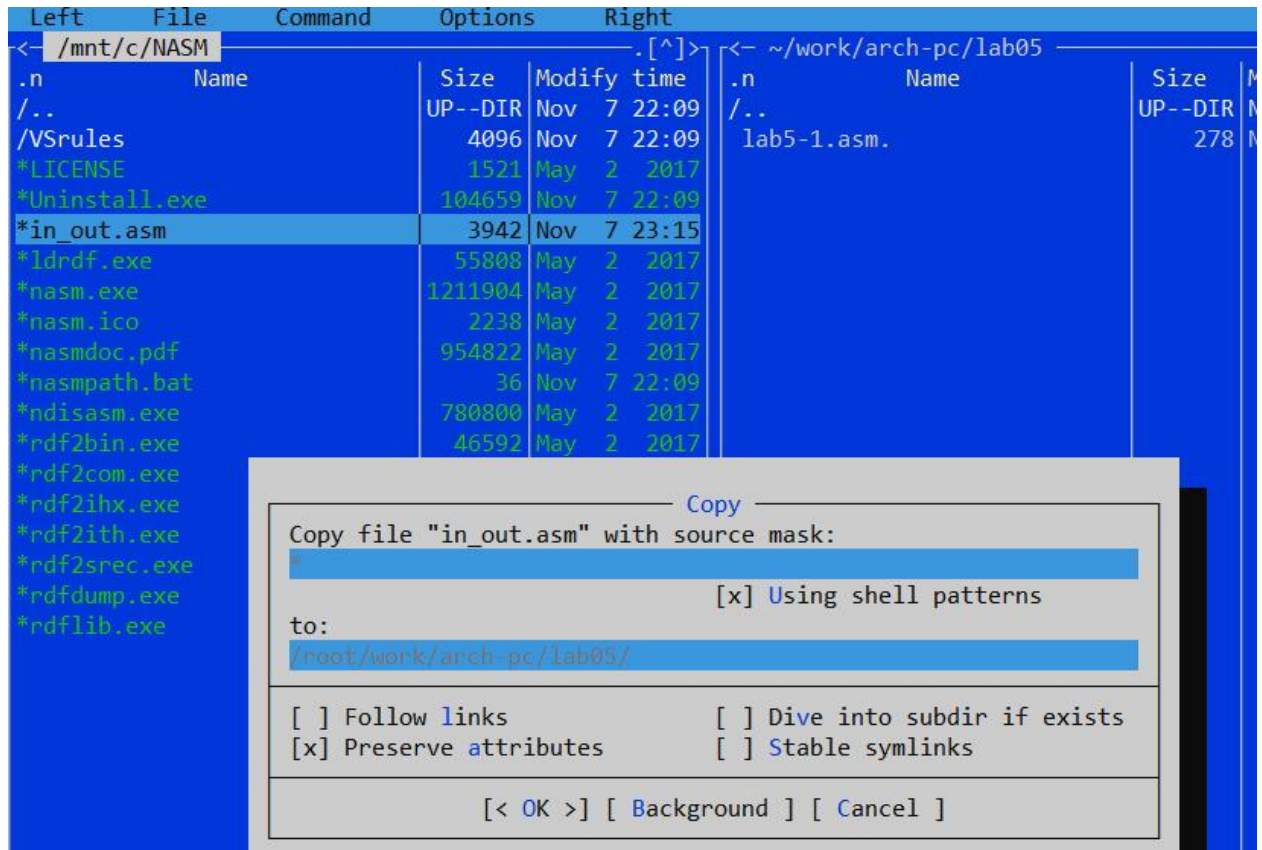
Для выполнения лабораторных работ использовала файл in_out.asm1.

(Подключаемый файл in_out.asm должен лежать в том же каталоге, что и файл с программой, в которой он используется.)

Затем в одной из панелей mc открыла каталог с файлом lab5-1.asm. В другой панели каталог со скачанным файлом in_out.asm.

Скопировала файл in_out.asm в каталог с файлом lab5-1.asm с помощью функциональной клавиши

F5



Дальше я с помощью функциональной клавиши F6 создала копию файла lab5-1.asm с именем lab5-2.asm, вывела файл lab5-1.asm, нажала клавишу F6, ввела имя файла lab5-2.asm и нажала клавишу Enter.

Исправила 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. и создала исполняемый файл и проверила его работу.

Выполнение для самостоятельной работы.

1. В начале я создала копию файла lab5-1.asm. Затем внесла изменения в программу (без использования внешнего файла in_out.asm)

При условии, чтобы она работала по следующему алгоритму:

- вывести приглашение типа “Введите строку:”;
- ввести строку с клавиатуры;
- вывести введенную строку на экран.

2. Затем я получила исполняемый файл и проверила его работу. На приглашение ввести строку ввела свою фамилию-Волчкова.

3. В конце я создала копию файла lab5-2.asm и исправьте текст программы с использование подпрограмм из внешнего файла in_out.asm

При условии, чтобы она работала по следующему алгоритму:

- вывести приглашение типа “Введите строку:”;
- ввести строку с клавиатуры;
- вывести введённую строку на экран.

Вывод.

Целью работой было - приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера mov и int, проделав задания, я разобралась в данной теме.

Список литературы

1. GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb>
2. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>.
3. Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>.
4. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: <https://www.amazon.com/Learningbash-Shell-Programming-Nutshell/dp/0596009658>.

6. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс, 2017.
11. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
12. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.
13. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВПетербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
14. Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-е изд. — М. : МАКС Пресс, 2011. — URL: http://www.stolyarov.info/books/asm_unix.
15. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
16. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science).

