

Отчёта по лабораторной работе №5

Дисциплина: архитектура компьютера

Нурыева Гулсолтан НКАбд-02-25

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
4.1	открытие MC	8
4.2	Подключения внешнего файла	11
5	Задание для самостоятельной работы	14
5.1	Первая работа	14
5.2	Вторая работа	15
6	Выводы	17
	Список литературы	18

Список иллюстраций

4.1	Открывание MC	8
4.2	Создания папки в каталоге	9
4.3	Создания файла	9
4.4	Редактирования файла	10
4.5	Проверка файла	10
4.6	оттранслирования, компоновка, ввеления ФИО	11
4.7	Копирования файла в нужный каталог	12
4.8	Редактирования файла	12
4.9	Проверка файла	13
4.10	Изменения sprintLF на sprint	13
	srintLF и sprint	13
4.11	Разница между	
5.1	Исправления текста lab5-1-1	15
5.2	Создания исполняемого файла и проверка файла lab5-1-1	15
5.3	Исправления текста lab5-2-2	16
5.4	Создания исполняемого файла и проверка файла lab5-2-2	1

1 Цель работы

Целью данной лабораторной работы является приобретение практических навыков работы в Midnight Commander, освоение инструкций языка ассемблера `mov` и `int`.

2 Задание

1. Основы работы с Midnight Commander
2. Структура программы на языке ассемблера NASM
3. Подключение внешнего файла
4. Выполнение заданий для самостоятельной работы

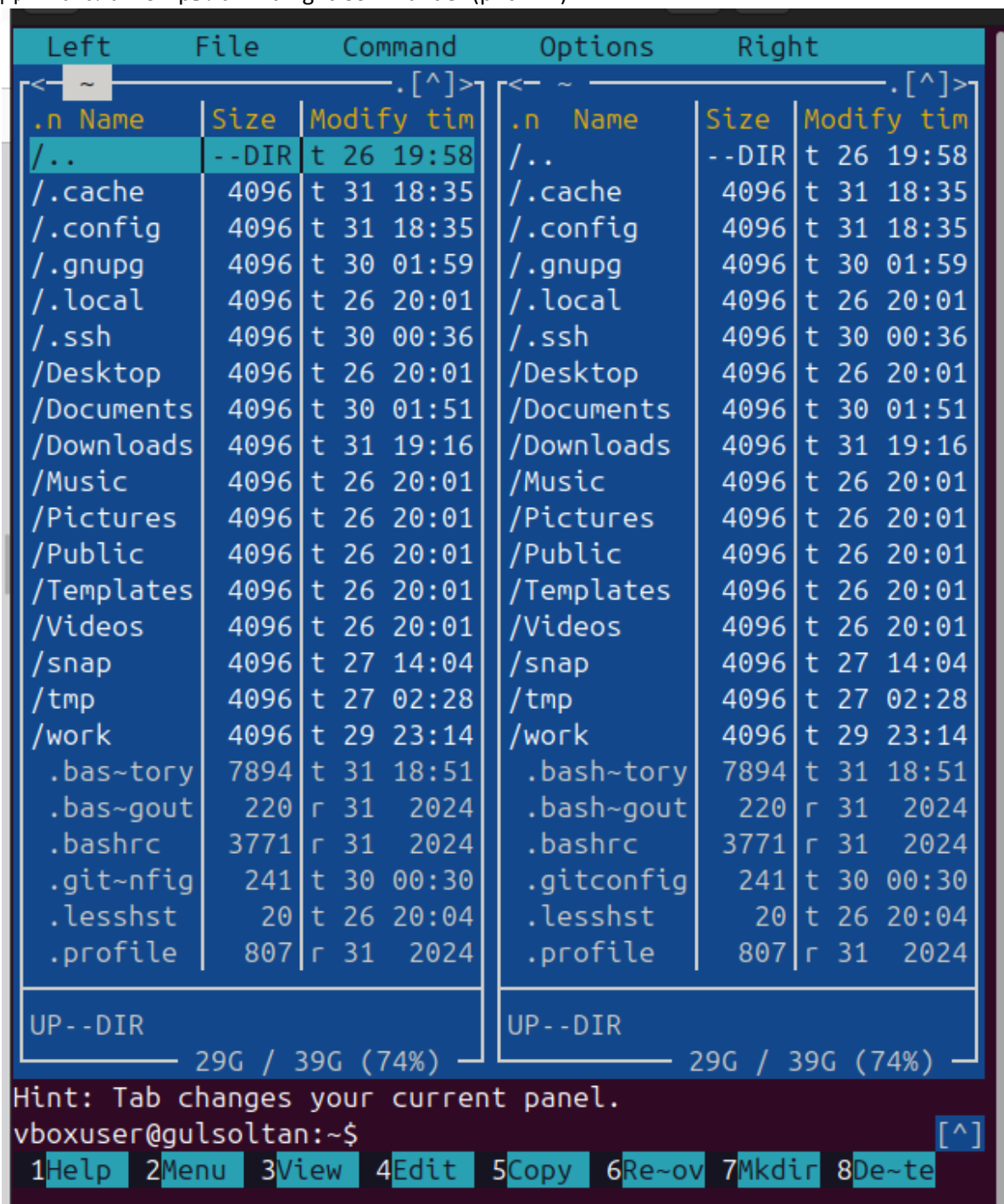
3 Теоретическое введение

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. Следующие комбинации клавиш облегчают работу с Midnight Commander: • Tab используется для переключения между панелями; • ↑ и ↓ используется для навигации, Enter для входа в каталог или открытия файла (если в файле расширений mc.ext заданы правила связи определённых расширений файлов с инструментами их запуска или обработки); • Ctrl + u (или через меню Команда > Переставить панели) меняет местами содержимое правой и левой панелей; • Ctrl + o (или через меню Команда > Отключить панели) скрывает или возвращает панели Midnight Commander, за которыми доступен для работы командный интерпретатор оболочки и выводимая туда информация. • Ctrl + x + d (или через меню Команда > Сравнить каталоги) позволяет сравнить содержимое каталогов, отображаемых на левой и правой панелях. Дополнительную информацию о Midnight Commander можно получить по команде man mc

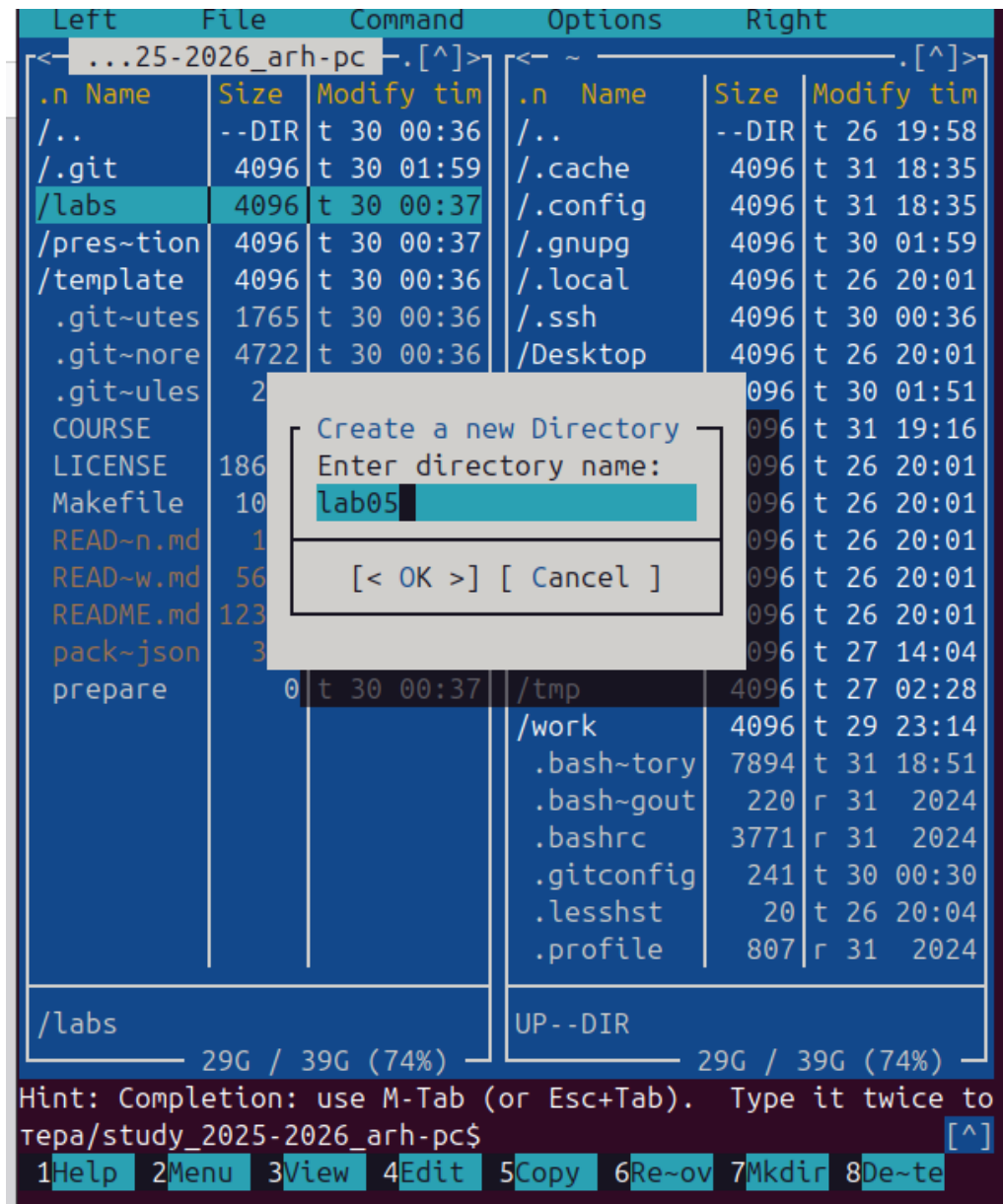
4 Выполнение лабораторной работы

4.1 открытие МС

Для начала я открыла Midnight Commander (рис. 4.1).



##Создание папки Потом перешла на каталог созданная в 4 лабораторной работе
work/arch-pc и там создала новую папку с помощью F7



##Создание файла

После я создала файл lab5-1.asm с помощью команды touch в созданной папке

Left	File	Command	Options	Right
<-- ...6_arh-pc/lab05 --.[^]>			<-- ~ --.[^]>	
.n Name	Size	Modify tim	.n Name	Size Modify tim
/..	--DIR	t 31 19:53	/..	--DIR t 26 19:58
lab5~.asm	0	t 31 19:56	/.cache	4096 t 31 18:35
			/.config	4096 t 31 18:35
			/.gnupg	4096 t 30 01:59
			/.local	4096 t 26 20:01
			/.ssh	4096 t 30 00:36
			/Desktop	4096 t 26 20:01
			/Documents	4096 t 30 01:51
			/Downloads	4096 t 31 19:16
			/Music	4096 t 26 20:01
			/Pictures	4096 t 26 20:01
			/Public	4096 t 26 20:01
			/Templates	4096 t 26 20:01
			/Videos	4096 t 26 20:01
			/snap	4096 t 27 14:04
			/tmp	4096 t 27 02:28
			/work	4096 t 29 23:14
			.bash~tory	7894 t 31 18:51
			.bash~gout	220 r 31 2024
			.bashrc	3771 r 31 2024
			.gitconfig	241 t 30 00:30
			.lessht	20 t 26 20:04
			.profile	807 r 31 2024
UP--DIR			UP--DIR	
29G / 39G (74%)			29G / 39G (74%)	
int: Completion: use M-Tab (or Esc+Tab). Type it twice to				
epa/study_2025-2026_arh-pc/lab05\$				
1Help 2Menu 3View 4Edit 5Copy 6Re~ov 7Mkdir 8De~te				

##Редактирование файла С помощью клавиши F4 я открыла файл и внес редактирования как было указанаю Потом сохранила и закрыла файл

```

/home/vb~5-1.asm [-M--] 20 L:[ 6+29 35/ 35] *(243[*])[X]
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
;----- системный вызов `read` -----
; После вызова инструкции 'int 80h' программа будет ожидать
; строки, которая будет записана в переменную 'buf1' размер
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 рабо
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра

```

1Help 2Save 3Mark 4Re~ac 5Copy 6Move 7Se~ch 8De~te

С помощью клавиши F3 проверила что файл содержит текст которую я написала

```

/home/vboxus~5/lab5-1.asm      1754/2431      72%
;-----
; 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'
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
1Help 2Un~ap 3Quit 4Hex 5Goto 6 7Se~ch 8Raw

```

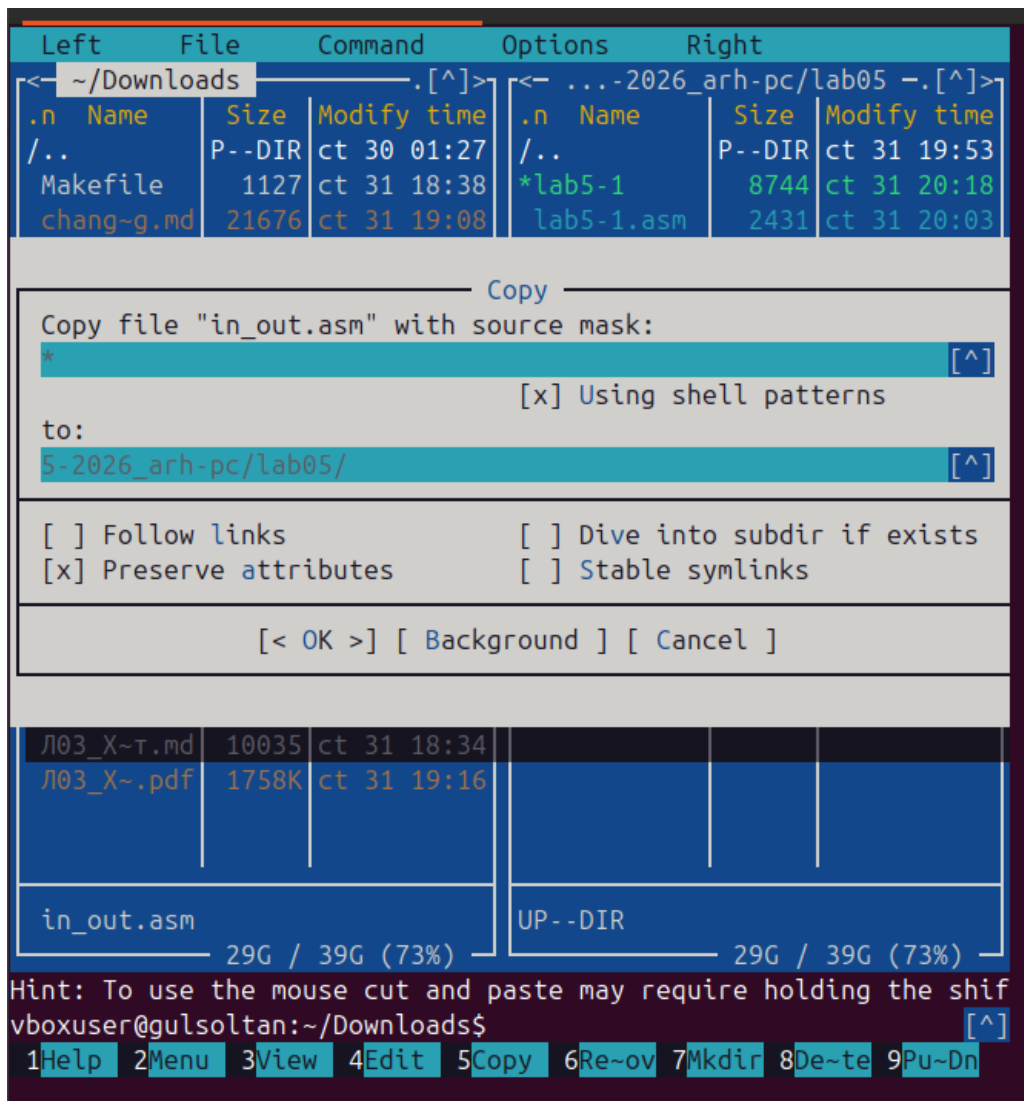
##Оттранслирования текста в объектный файл. Компоновка файла Оттранслировала текст программы lab5-1.asm в объектный файл.Потом выполнила компоновку объектного файла и запустила получившийся исполняемый файл. На строку 'Введите строку:' написала свое

ФИО

```
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/lab05$ nasm -f elf lab5-1.asm
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/lab05$ ./lab5-1
Введите строку:
Нурыева Гулсолтан
```

4.2 Подключения внешнего файла

Для начала скачал файл in_out.asm со страницы курса в ТУИС. Когда скачала этот файл лежал в каталоге “Загрузки”. Поэтому я в другой панели открыла каталог “Загрузки” и с помощью клавиши F5 я скопировал файл в тот каталог в котором лежал lab5-1.asm

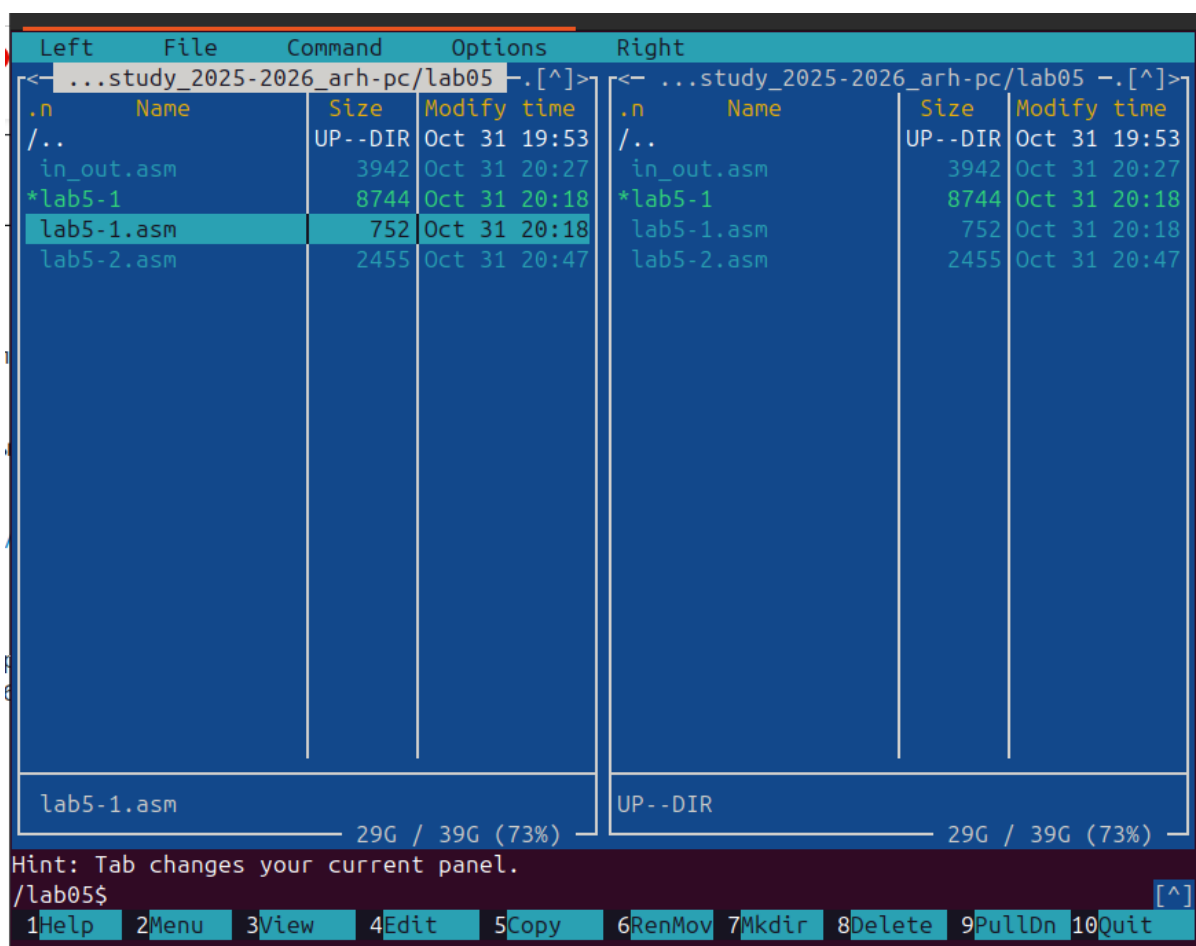


Потом исправила текст программы в файле с использованием подпрограммы врешнего файла in_out.asm.

```
mc [vboxuser@gulsoltan]:~/work/study/2025-2026/... x vboxuser@gulsoltan: ~/work/study/2025-2026/Апх... x
/home/vboxu~/lab5-1.asm [-M--] 20 L:[ 1+ 5 6/ 38] *(357 /2455b) 0034 0x022 [*][X]
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
;----- Объявление переменных -----
%include "in_out.asm"

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
;----- системный вызов 'read' -----
; После вызова инструкции 'int 80h' программа будет ожидать ввода
; строки, которая будет записана в переменную 'buf1' размером 80 байт
1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn 10Quit
```

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



После этого я создала исполняемый файл и проверила его работу

Потом сменила `sprintf` на `sprint`

```
1 ;-----  
2 ; Программа вывода сообщения на экран и ввода строки с клавиатуры  
3 ;-----  
4 %include 'in_out.asm' ; подключение внешнего файла  
5 SECTION .data ; Секция инициализированных данных  
6 msg: DB 'Введите строку: ',0h ; сообщение  
7 SECTION .bss ; Секция не инициализированных данных  
8 buf1: RESB 80 ; Буфер размером 80 байт  
9 SECTION .text ; Код программы  
10 GLOBAL _start ; Начало программы  
11 _start: ; Точка входа в программу  
12 mov eax, msg ; запись адреса выводимого сообщения в `EAX`  
13 call printf ; вызов подпрограммы печати сообщения  
14 mov ecx, buf1 ; запись адреса переменной в `EAX`  
15 mov edx, 80 ; запись длины вводимого сообщения в `EBX`  
16 call read ; вызов подпрограммы ввода сообщения  
17 call quit ; вызов подпрограммы завершения
```

Разница между первым исполняемым файлом и вторым в том, что запуск первого запрашивает ввод с новой строки, а программа, которая выполняется при запуске второго, запрашивает ввод без переноса на новую строку, потому что в этом заключается различия между подпрограммами `sprintf` и `sprint`.


```
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/lab05$ nasm -f elf lab5-2.asm
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/lab05$ ./lab5-2
Введите строку: Нурыева Гулсолтан
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/lab05$ nasm -f elf lab5-2.asm
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/lab05$ ./lab5-2
Введите строку:
Нурыева Гулсолтан
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/lab05$
```

5 Задание для самостоятельной работы

5.1 Первая работа

Для начало я создала копию файла lab5-1.asm и исправила текст программы чтобы она работала по алгоритму:

- Вывести приглашения типа “Введите строку”

- Ввести строку с клавиатуры
- Вывести введенную строку на экран

```
...5-2026/Архитектура компьютера/study_2025-2026_arh-pc/lab05/lab5-3-1.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,4
mov ebx,1
mov ecx,buf1
mov edx,80
int 80h
mov eax,1
mov ebx,0

[ Read 27 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
```

Потом создала исполняемый файл и проверила его работу

```
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/lab05$ nasm -f elf lab5-3-1.asm
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/lab05$ ld -m elf_i386 -o lab5-3-1 lab5-3-1.o
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/lab05$ ./lab5-3-1
Введите строку:
Нурьева Гулсолтан
Нурьева Гулсолтан
vboxuser@gulsoltan:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_arh-pc/lab05$
```

5.2 Вторая работа

Для начала я создала копия файла lab5-2.asm. Потом исправила текст программы с использованием подпрограммы 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

```
...5-2026/Архитектура компьютера/study_2025-2026_arh-pc/lab05/lab5-3-2.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
```

Потом создала исполняемый файл и проверила его работу

```
vboxuser@gulsooltan:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a  
rh-pc/lab05$ nasm -f elf lab5-3-2.asm  
vboxuser@gulsooltan:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a  
rh-pc/lab05$ ld -m elf_i386 -o lab5-3-2 lab5-3-2.o  
vboxuser@gulsooltan:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a  
rh-pc/lab05$ ./lab5-3-2  
Введите строку: Нурыева Гулсолтан  
Нурыева Гулсолтан  
vboxuser@gulsooltan:~/work/study/2025-2026/Архитектура компьютера/study_2025-2026_a  
rh-pc/lab05$
```

6 Выводы

При выполнении данной лабораторной работы я приобрела практические навыки работы в Midnight Commander, а также освоила инструкции языка ассемблера mov и int.

Список литературы

1. Лабораторная работа №6