

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: Павличенко Родион Андреевич

Группа: НПИбд-02-24

МОСКВА

2024 г.

Цель работы.

Цель работы: Ознакомиться с программой Midnight commander и освоить написание программ на языке ассемблера с помощью инструкций mov и int.

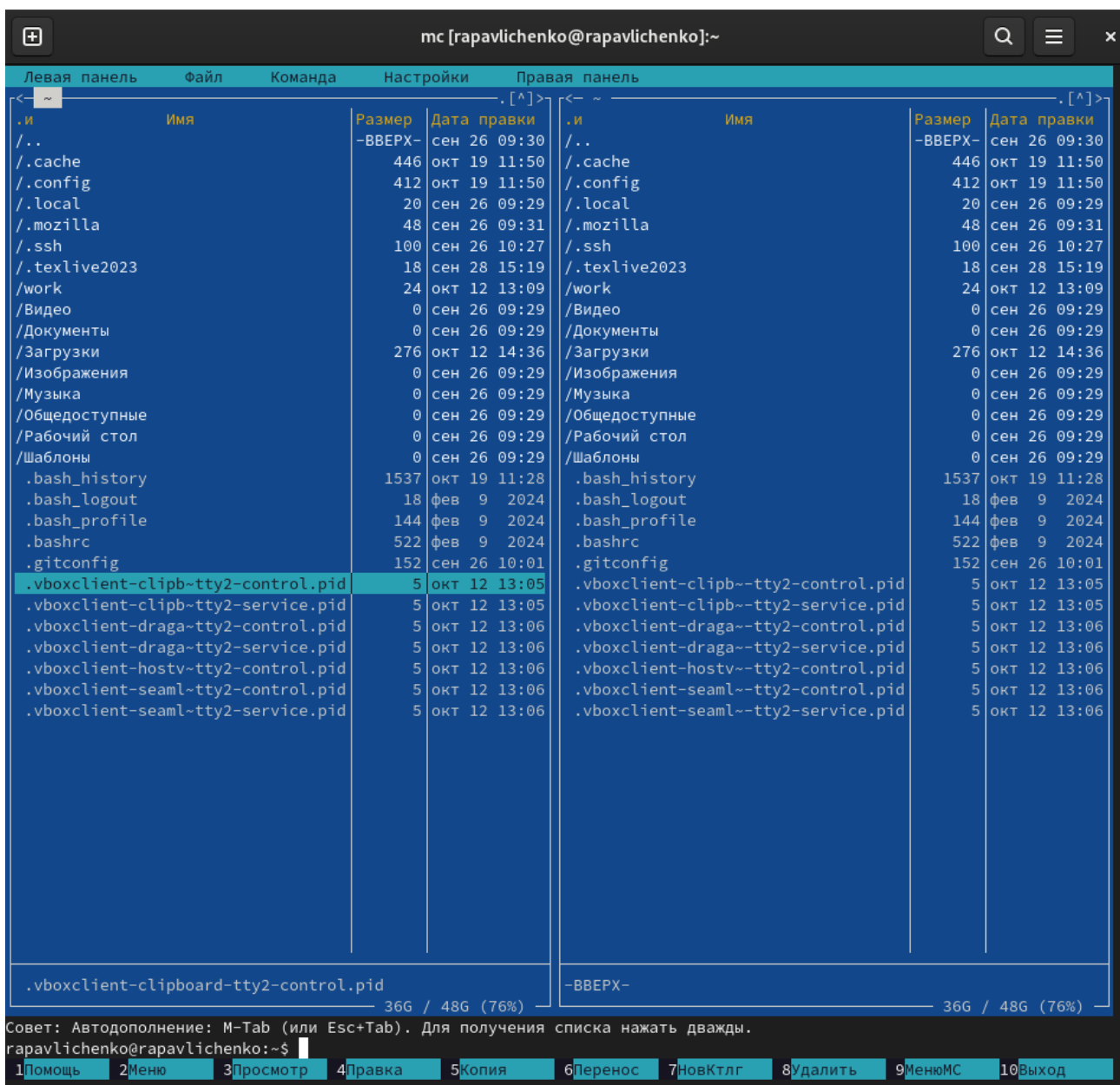
Выполнение лабораторной работы

- 1) Для начала выполнения лабораторной работы нам необходимо открыть Midnight commander с помощью команды mc

```
rapavlichenko@rapavlichenko:~$ mc
```

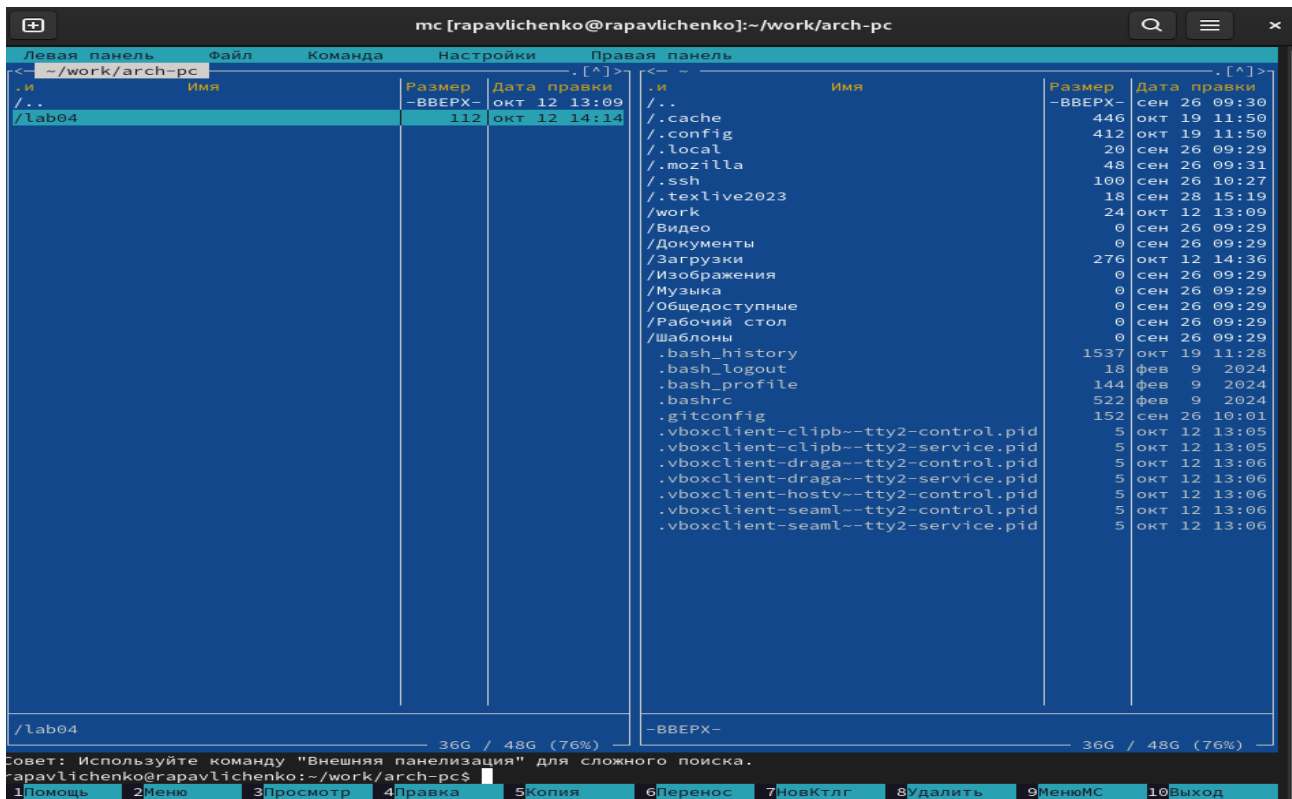
Запуск Midnight commander

- 2) После ввода команды мы увидим такой интерфейс



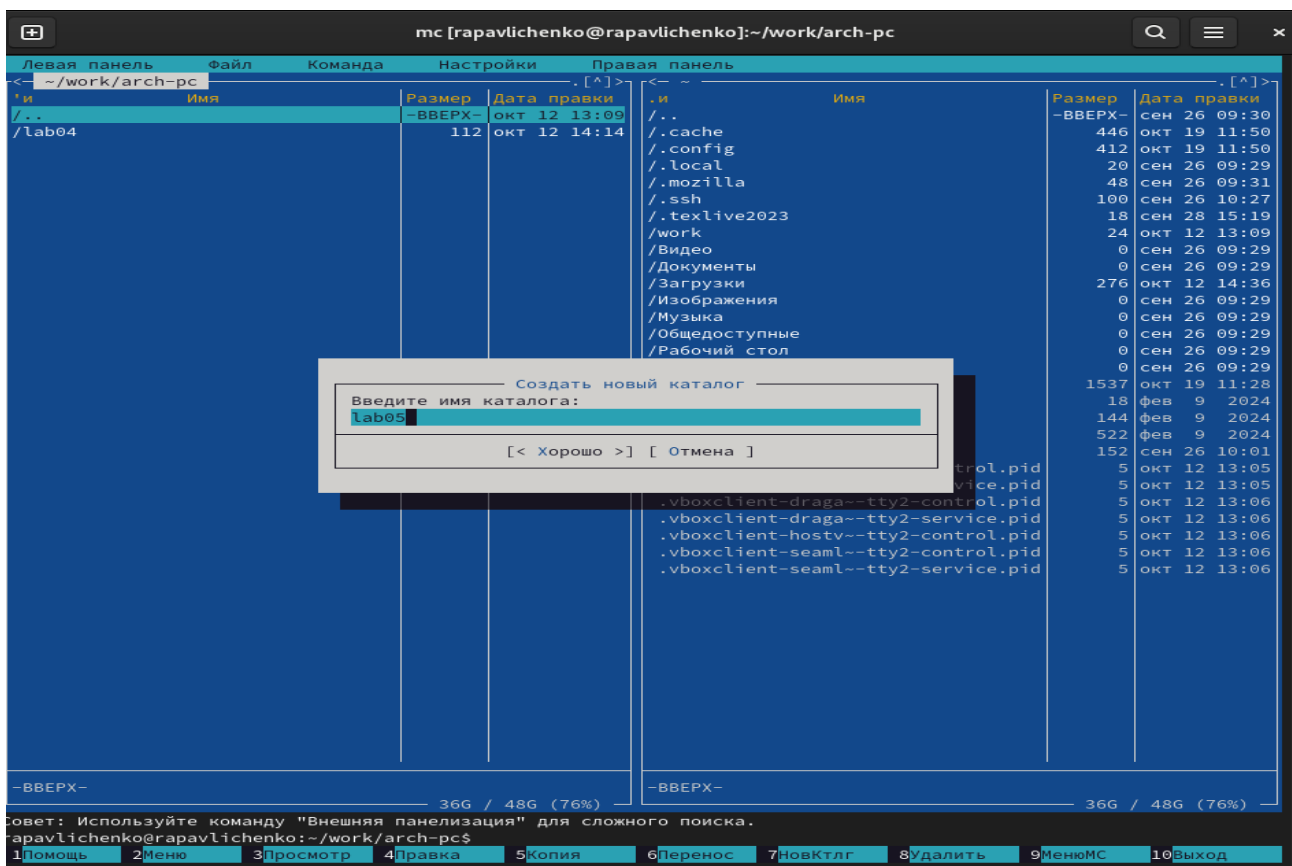
Интерфейс midnight commander

3) С помощью стрелок и клавиши Enter перейдём в каталог ~/work/arch-pc



Переход в нужный каталог (~/work/arch-pc)

4) Создадим папку lab05 с помощью клавиши F7



Создание папки

5) Теперь с помощью команды touch создадим файл lab5-1.asm

The screenshot shows the mc file manager interface. The left pane displays the current directory `~/work/arch-pc/lab05` with a table of files and folders. The right pane shows the contents of the selected file `lab5-1.asm`, which is empty. The command bar at the bottom shows the command `touch lab5-1.asm` being executed. The status bar at the bottom indicates the disk usage: 36G / 48G (76%).

Имя	Размер	Дата правки
..	-BBERX-	окт 19 12:04

Имя	Размер	Дата правки
..	-BBERX-	сен 26 09:30
/.cache	446	окт 19 11:50
/.config	412	окт 19 11:50
/.local	20	сен 26 09:29
/.mozilla	48	сен 26 09:31
/.ssh	100	сен 26 10:27
/.texlive2023	18	сен 28 15:19
/work	24	окт 12 13:09
/Видео	0	сен 26 09:29
/Документы	0	сен 26 09:29
/Загрузки	276	окт 12 14:36
/Изображения	0	сен 26 09:29
/Музыка	0	сен 26 09:29
/Общедоступные	0	сен 26 09:29
/Рабочий стол	0	сен 26 09:29
/Шаблоны	0	сен 26 09:29
.bash_history	1537	окт 19 11:28
.bash_logout	18	фев 9 2024
.bash_profile	144	фев 9 2024
.bashrc	522	фев 9 2024
.gitconfig	152	сен 26 10:01
.vboxclient-clipb--tty2-control.pid	5	окт 12 13:05
.vboxclient-clipb--tty2-service.pid	5	окт 12 13:05
.vboxclient-draga--tty2-control.pid	5	окт 12 13:06
.vboxclient-draga--tty2-service.pid	5	окт 12 13:06
.vboxclient-hostv--tty2-control.pid	5	окт 12 13:06
.vboxclient-seaml--tty2-control.pid	5	окт 12 13:06
.vboxclient-seaml--tty2-service.pid	5	окт 12 13:06

Совет: %D/%T распространяется на отмеченные файлы в противоположном каталоге.
 rapavlichenko@rapavlichenko:~/work/arch-pc/lab05\$ touch lab5-1.asm

1Помощь 2Меню 3Просмотр 4Правка 5Копия 6Перенос 7НовКтлг 8Удалить 9МенюМС 10Выход

Создание файла lab5-1.asm с помощью команды touch прямо в mc

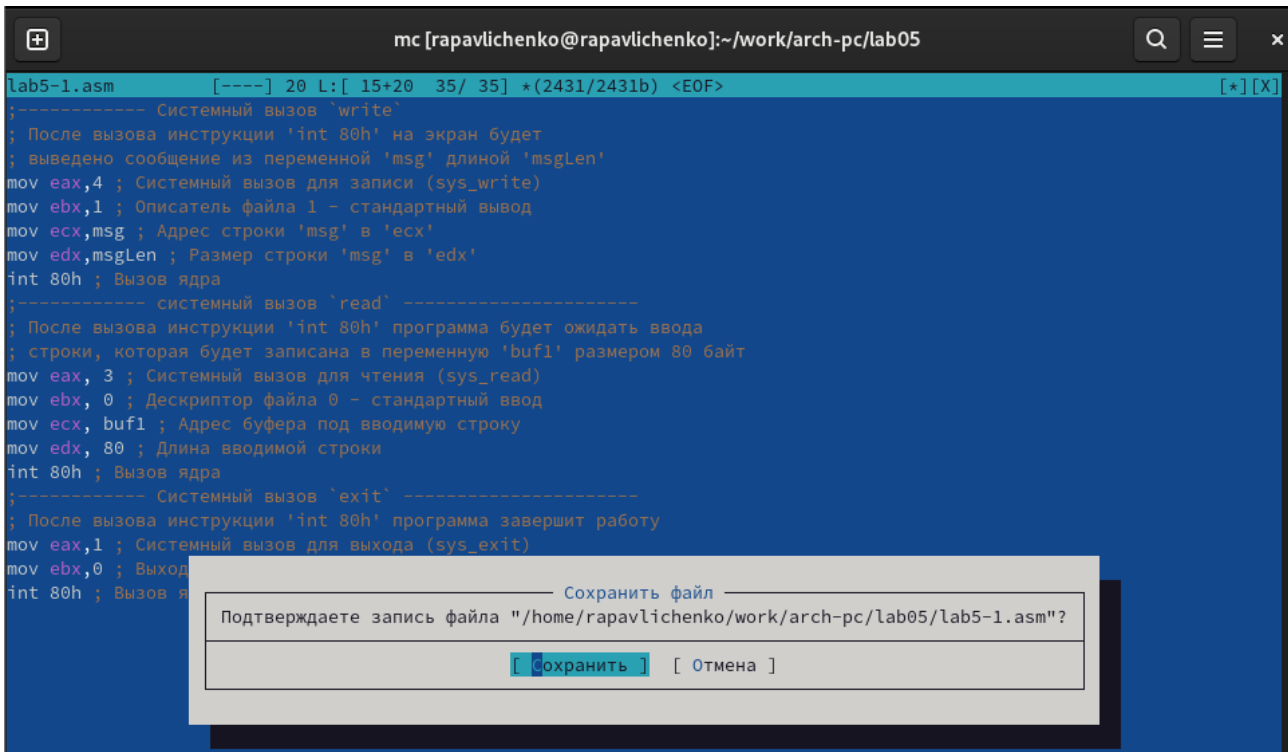
6) Теперь с помощью клавиши F4 откроем только что созданный файл и отредактируем файл, поместим в него следующий код

The screenshot shows the mc file manager interface with the file `lab5-1.asm` open for editing. The editor displays assembly code for a program that writes a message to the screen, reads input from the user, and then exits. The code is as follows:

```
[ -M-- ] 20 L: [ 15+20 35/ 35 ] *(2431/2431b) <EOF>
--- Системный вызов 'write'
Зова инструкции 'int 80h' на экран будет
сообщение из переменной 'msg' длиной 'msgLen'
; Системный вызов для записи (sys_write)
; Описатель файла 1 - стандартный вывод
g ; Адрес строки 'msg' в 'ecx'
gLen ; Размер строки 'msg' в 'edx'
Вызов ядра
--- системный вызов 'read' -----
Зова инструкции 'int 80h' программа будет ожидать ввода
которая будет записана в переменную 'buf1' размером 80 байт
; Системный вызов для чтения (sys_read)
; Deskриптор файла 0 - стандартный ввод
uf1 ; Адрес буфера под вводимую строку
0 ; Длина вводимой строки
Вызов ядра
--- Системный вызов 'exit' -----
Зова инструкции 'int 80h' программа завершит работу
; Системный вызов для выхода (sys_exit)
; Выход с кодом возврата 0 (без ошибок)
Вызов ядра
```

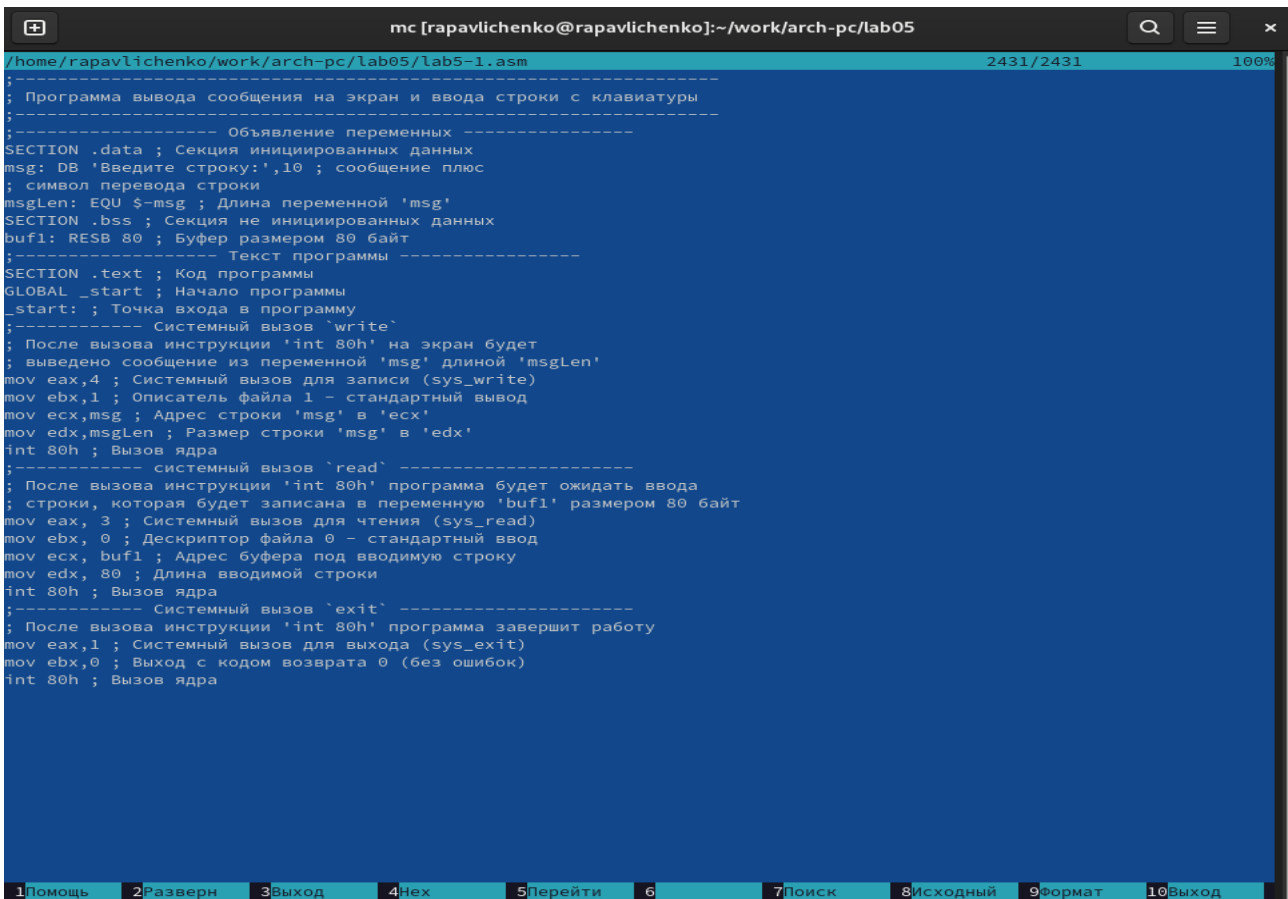
Редактирование файла lab5-1.asm

7) Теперь сохраним его клавишей F2



Сохранение файла lab5-1.asm

8) С помощью функциональной клавиши F3 откроем файл lab5-1.asm для просмотра. Убедимся, что файл содержит текст программы.



Проверка успешного редактирования

9) Теперь скомпилируем его

Совет: Для смены каталога во время набора команды нажмите М-с (быстрая смена).

```
rapavlichenko@rapavlichenko:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
```

1Помощь 2Меню 3Просмотр 4Правка 5Копия 6Перенос 7НовКтлг 8Удалить 9МенюМС 10Выход

Компиляция файла с помощью nasm

10) И соберём

Совет: Для смены каталога во время набора команды нажмите М-с (быстрая смена).

```
rapavlichenko@rapavlichenko:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
```

1Помощь 2Меню 3Просмотр 4Правка 5Копия 6Перенос 7НовКтлг 8Удалить 9МенюМС 10Выход

Сборка исполняемого файла с помощью ld

11) После этого запустим получившийся исполняемый файл

```
rapavlichenko@rapavlichenko:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
```

Запуск исполняемого файла

12) Теперь введём ФИО

```
rapavlichenko@rapavlichenko:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Павличенко Родион Андреевич
```

Взаимодействие с программой

13) После нажатия Enter программа завершится и ничего не произойдёт. Теперь скачаем файл in_out.asm и откроем папку с ним в правой панели

mc [rapavlichenko@rapavlichenko]:~/Загрузки

Левая панель		Правая панель	
Имя	Размер	Имя	Размер
lab5-1	8744	in_out.asm	3942
lab5-1.asm	2431	pandoc-crossref-Linux.tar.xz	8778656
lab5-1.o	752	pandoc-crossref.1	43994
		report.docx	1349618
		Павличенко_ОТЧЕТ_ЛБ3.doc	313344
		Павличенко_ОТЧЕТ_ЛБ3.pdf	880267

Совет: Если ваш терминал без функц. клавиш, вам поможет ESC+цифра.

```
rapavlichenko@rapavlichenko:~/Загрузки$
```

1Помощь 2Меню 3Просмотр 4Правка 5Копия 6Перенос 7НовКтлг 8Удалить 9МенюМС 10Выход

Открытие папки с файлом in_out.asm в правой панели

14) Скопируем его в нашу рабочую папку с помощью F6

Перемещение

Переместить файл "in_out.asm" с исходным шаблоном:
*
[x] Метасимволы shell

В:
/home/rapavlichenko/work/arch-pc/lab05/

☐ Разыменовывать ссылки

☐ Внутрь подкаталога, если есть

☒ Сохранять атрибуты

☐ Изменять относительные ссылки

[< Хорошо >]

[В фоне]

[Отмена]

Копирование файла с помощью F6

15) Теперь сделаем копию файла lab5-1.asm с помощью команды F5. Назовём копию lab5-2.asm

Перемещение

Переместить файл "lab5-1.asm" с исходным шаблоном:
* [^]
[x] Метасимволы shell

В:
/home/rapavlichenko/work/arch-pc/lab05/lab5-2.asm [^]

☐ Разыменовывать ссылки

☐ Внутрь подкаталога, если есть

☒ Сохранять атрибуты

☐ Изменять относительные ссылки

[< Хорошо >]

[В фоне]

[Отмена]

Копирование файла с помощью F5

16) Теперь наша папка выглядит следующим образом

~ /work/arch-pc/lab05		. [^]>	
.и	Имя	Размер	Дата правки
/..		-ВВЕРХ-	окт 19 12:04
	in_out.asm	3942	окт 19 12:39
*	lab5-1	8744	окт 19 12:35
	lab5-1.asm	2431	окт 19 12:29
	lab5-1.o	752	окт 19 12:33
	lab5-2.asm	2431	окт 19 12:29

Текущий вид рабочей папки

17) Откроем в текстовом редакторе файл lab5-2.asm и напишем туда следующий код

```
mc [rapavlichenko@rapavlichenko]:~/work/arch-pc/lab05
lab5-2.asm [-M--] 54 L: [ 1+14 15/ 17] *(1076/1224b) 0010 0x00A [*] [X]
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не инициализированных 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
```

Редактирование файла lab5-2.asm

18) После чего создадим исполняемый файл с помощью nasm и ld

```
rapavlichenko@rapavlichenko:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
rapavlichenko@rapavlichenko:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
rapavlichenko@rapavlichenko:~/work/arch-pc/lab05$ ./lab5-2
```

Создание исполняемого файла

19) Запустим созданный файл и введем ФИО

```
rapavlichenko@rapavlichenko:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Павличенко Родион Андреевич
```

Запуск исполняемого файла

20) Он работает также, как и файл lab5-1, но использует для работы сторонний файл. Попробуем теперь вместо команды sprintf использовать просто команду sprint

```
mc [rapavlichenko@rapavlichenko]:~/work/arch-pc/lab05
lab5-2.asm [-M--] 11 L: [ 1+12 13/ 17] *(847 /1222b) 0032 0x020 [*] [X]
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не инициализированных 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 sprint ; 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
```

Изменение файла lab5-2.asm

21) Точно также соберём исполняемый файл и запустим его

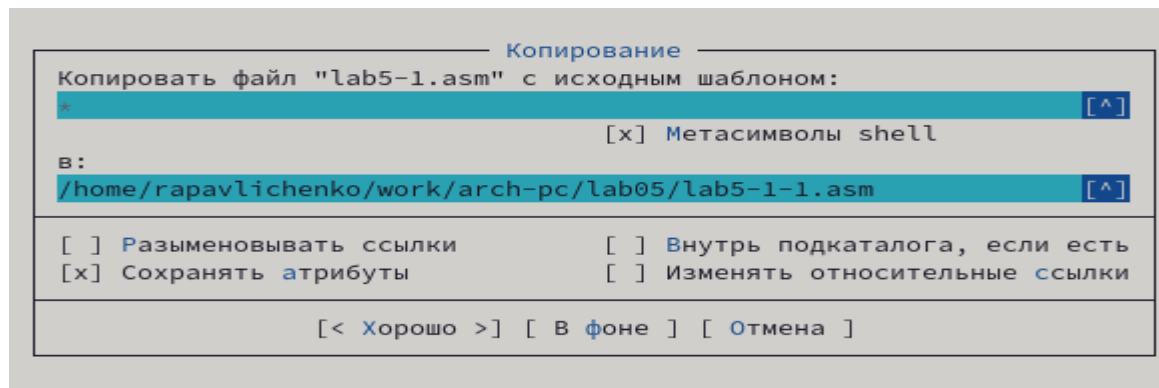
```
rapavlichenko@rapavlichenko:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Павличенко Родион Андреевич
```

Запуск изменённого файла

Как мы видим, теперь нет переноса на следующую строку. Этим и отличаются команды `sprintLF` от `sprint`. Первая добавляет перенос после текста, а вторая нет

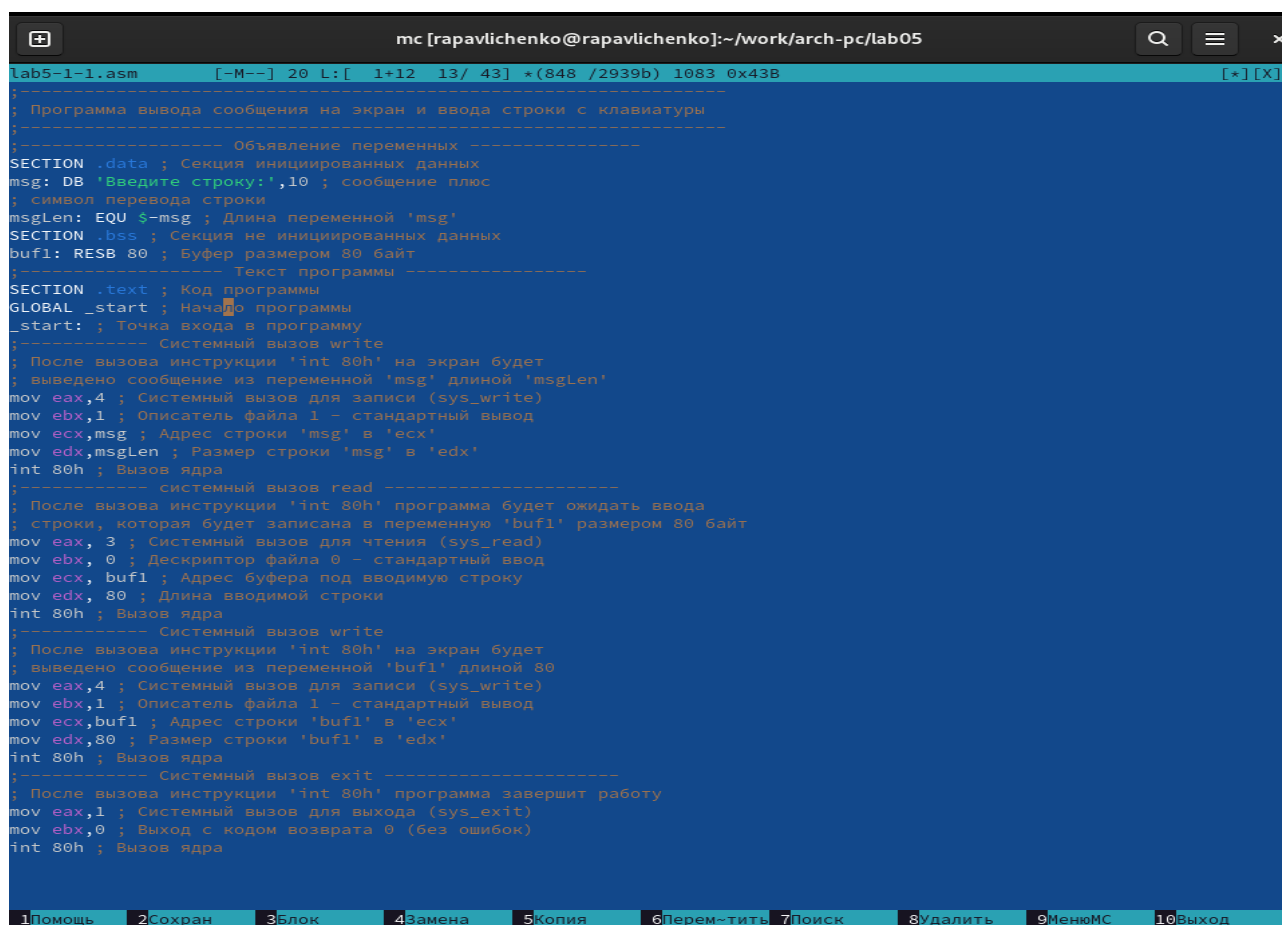
Выполнение задания для самостоятельной работы

1) Теперь создадим с помощью F5 копию файла `lab5-1.asm`



Создание копии файла `lab5-1.asm`

2) Изменим копию так, чтобы она выводила тот текст, который получила на ввод. Для этого перед системным вызовом `exit` вставим текст с системным вызовом `write`. Он очень похож на системный вызов `write`, который уже был в коде, но есть несколько отличий. Так, мы перемещаем адрес строки `buf1` в `ecx` и размер строки `buf1` (80) в `edx`



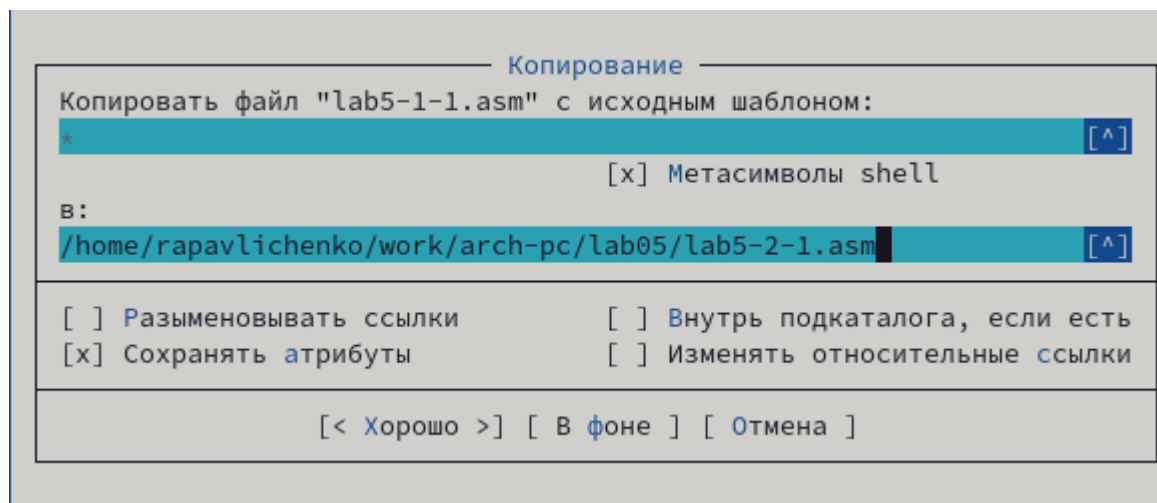
Изменение файла `lab5-1-1.asm`

3) Сохраним изменения и создадим исполняемый файл. Запустим его и проверим, что всё работает.

```
rapavlichenko@rapavlichenko:~/work/arch-pc/lab05$ nasm -f elf lab5-1-1.asm
rapavlichenko@rapavlichenko:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1-1 lab5-1-1.o
rapavlichenko@rapavlichenko:~/work/arch-pc/lab05$ ./lab5-1-1
Введите строку:
Павличенко Родион Андреевич
Павличенко Родион Андреевич
```

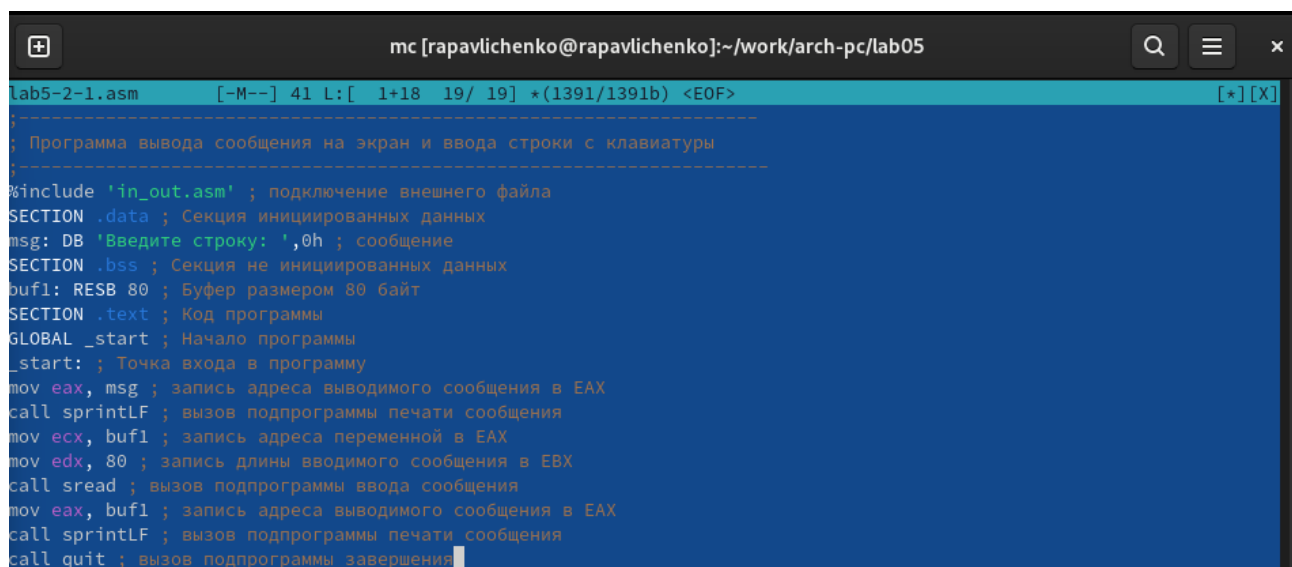
Создание исполняемого файла. Проверка работы программы

4) Теперь создадим с помощью F5 копию файла lab5-2.asm



Создание копии файла lab5-2.asm

5) Теперь сделаем так, чтобы этот код также выводил тот текст, что получит на ввод. Для этого перед последней строкой добавим строчку, которая записывает в eax адрес buf1, а также строчку, которая вызывает подпрограмму sprintf



Изменение файла lab5-2-1.asm

б)Теперь создадим исполняемый файл, запустим программу и убедимся, что она работает.

```
rapavlichenko@rapavlichenko:~/work/arch-pc/lab05$ nasm -f elf lab5-2-1.asm
rapavlichenko@rapavlichenko:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2-1 lab5-2-1.o
rapavlichenko@rapavlichenko:~/work/arch-pc/lab05$ ./lab5-2-1
Введите строку:
Павличенко Родион Андреевич
Павличенко Родион Андреевич
```

Создание исполняемого файла, проверка работы программы

Выводы

В результате выполнения работы были получены навыки работы с Midnight commander, а также навыки написания простых программ ввода-вывода на языке ассемблера