

Отчёт по лабораторной работе №4

дисциплина: Архитектура вычислительных систем

Мосолов Александр Денисович

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Программа вывода сообщения на экран и ввода строки с клавиатуры	6
2.2	Подключение внешнего файла in_out.asm	10
2.3	Задание для самостоятельной работы	13
3	Выводы	16

Список иллюстраций

2.1	Используем команду mc	6
2.2	Структура Midnight Commander	6
2.3	Каталог arch-pc в Midnight Commander	7
2.4	Создаём папку lab05	7
2.5	Создаём файл lab5-1.asm	8
2.6	Выбираем редактор	8
2.7	Текст программы из листинга 5.1	9
2.8	Использования функциональной клавиши F3	9
2.9	Объектный файл	10
2.10	Используем команды для запуска программы	10
2.11	Перемещаем скачанный файл	11
2.12	Каталог lab05	11
2.13	Создаем копию файла с помощью F5	12
2.14	Текст программы lab5-2	12
2.15	Используем подпрограмму sprintLF	13
2.16	Используем подпрограмму sprint	13
2.17	Текст программы my_cod.asm	13
2.18	Текст программы my_cod-2.asm	14
2.19	Запускаем программу my_cod	14
2.20	Запускаем программу my_cod-2	15
2.21	Просмотр каталога lab05	15
2.22	Загружаем файлы на github	15

Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

2.1 Программа вывода сообщения на экран и ввода строки с клавиатуры

С помощью команды *mc* открываем *Midnight Commander*

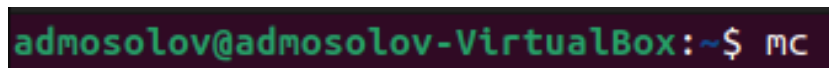


Рис. 2.1: Используем команду mc

С помощью клавиш *вверх*, *вниз* можно перемещаться по меню, переходить в каталоги с помощью *Enter*.

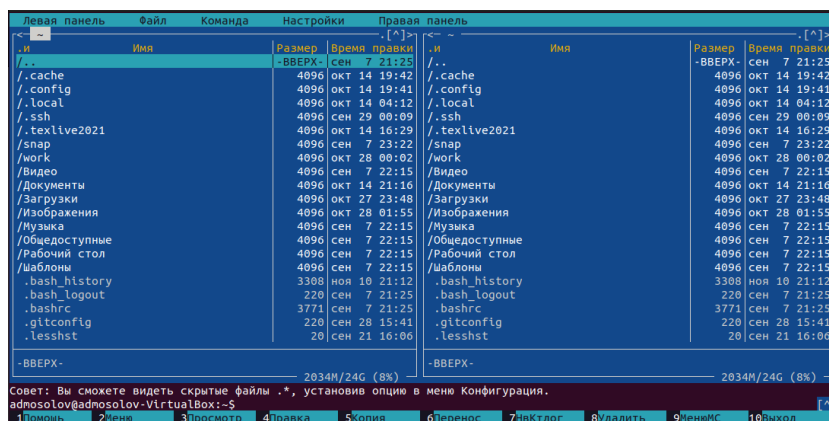


Рис. 2.2: Структура Midnight Commander

Переходим в каталог *~/work/arch-pc*.

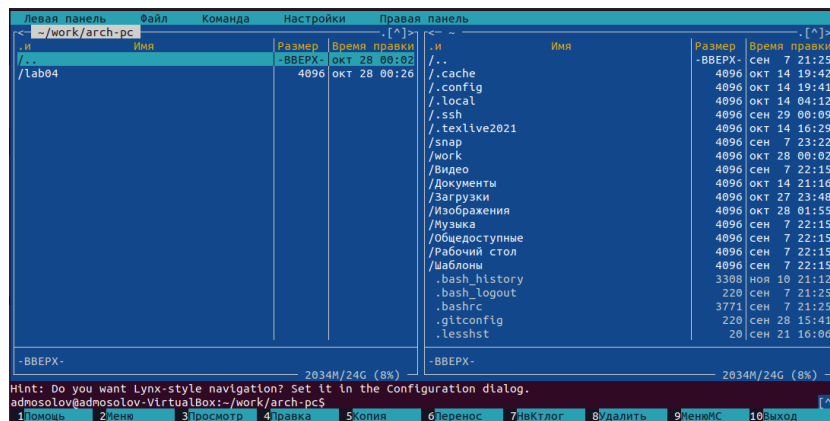


Рис. 2.3: Каталог arch-pc в Midnight Commander

С помощью функциональной клавиши *F7* создаём папку *lab05* и переходим в созданный каталог.

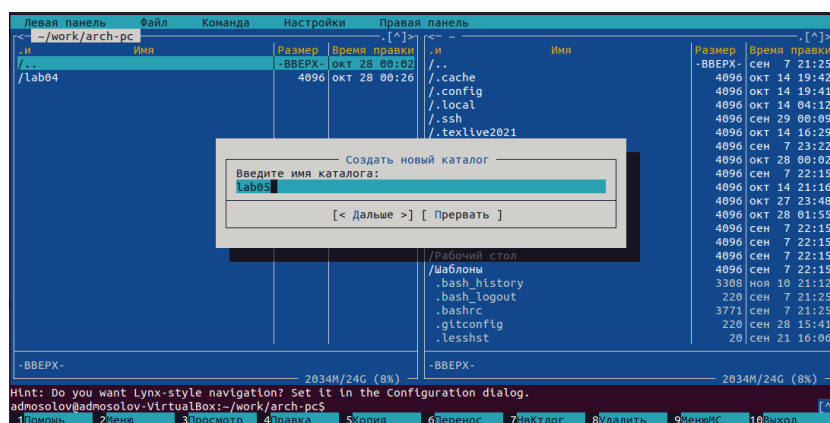


Рис. 2.4: Создаём папку lab05

Пользуясь строкой ввода и командой *touch* создаём файл *lab5-1.asm*, убеждаемся в *Midnight Commander*, что файл был создан.

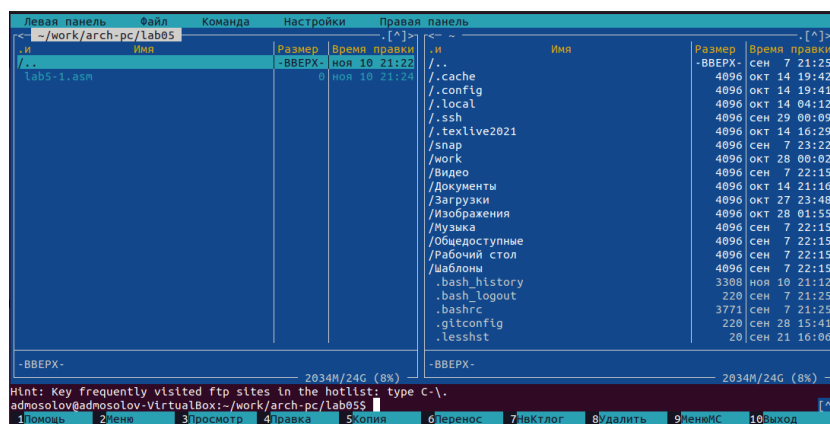


Рис. 2.5: Создаём файл lab5-1.asm

С помощью функциональной клавиши *F4* открываем файл *lab5-1.asm* для редактирования во встроенном редакторе. Выбираем в качестве основного редактора *nano* с помощью клавиши *1* в терминале.

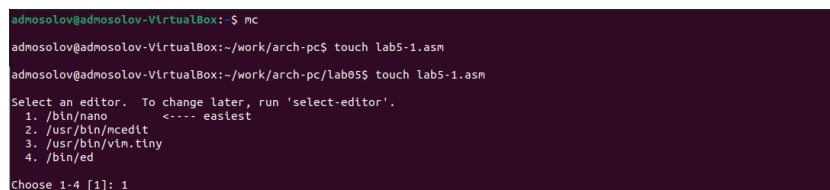


Рис. 2.6: Выбираем редактор

Вводим текст программы из листинга 5.1, сохраняем (*Ctrl+o*) изменения и закрываем файл (*Ctrl+x*).


```

GNU nano 6.2 /home/admosolov/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

```

Рис. 2.7: Текст программы из листинга 5.1

С помощью функциональной клавиши *F3* открываем файл *lab5-1.asm* для просмотра. Убеждаемся, что файл содержит текст программы.

```

/home/admosolov/work/arch-pc/lab05/lab5-1.asm 362/362 100%
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

```

Рис. 2.8: Использование функциональной клавиши *F3*

Транслируем текст программы *lab5-1.asm* в объектный файл для этого используем команду *nasm -f elf lab5-1.asm*. Выполняем компоновку объектного файла с помощью команды *ld -m elf_i386 -o lab5-1 lab5-1.o* и запускаем получившийся исполняемый файл, используя команду *./lab5-1*. Программа выводит строку *‘Введите строку:’* и ожидает ввода с клавиатуры. На запрос вводим ФИО.

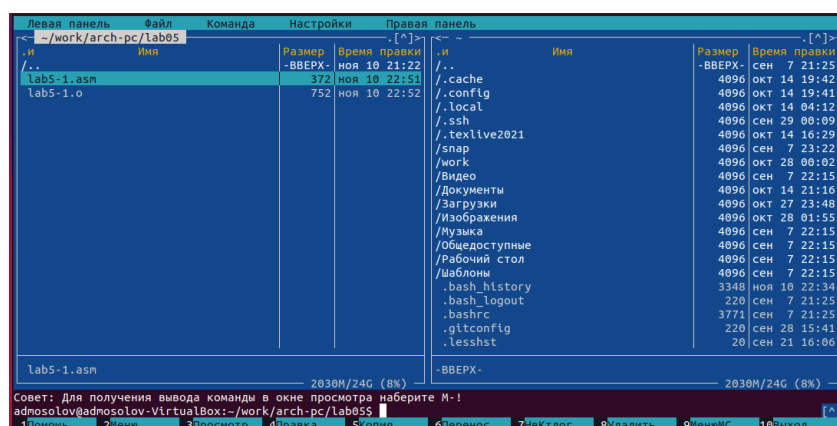


Рис. 2.9: Объектный файл

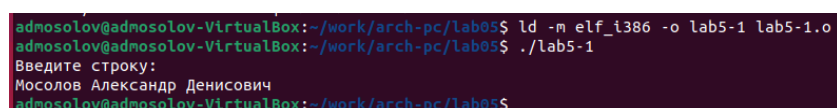


Рис. 2.10: Используем команды для запуска программы

2.2 Подключение внешнего файла in_out.asm

Для упрощения написания программ часто встречающиеся одинаковые участки кода (такие как, например, вывод строки на экран или выход из программы) можно оформить в виде подпрограмм и сохранить в отдельные файлы, а во всех нужных местах поставить вызов нужной подпрограммы. Это позволяет сделать основную программу более удобной для написания и чтения. Для вызова подпрограммы из внешнего файла используется инструкция *call*.

Скачиваем файл *in_out.asm* со страницы курса в ТУИС. Переносим его в папку с программой.

В одной из панелей *mc* открываем каталог с файлом *lab5-1.asm*. В другой панели каталог со скаченным файлом *in_out.asm* (для перемещения между панелями используем *Tab*). Копируем файл *in_out.asm* в каталог с файлом *lab5-1.asm* с помощью функциональной клавиши *F5*.

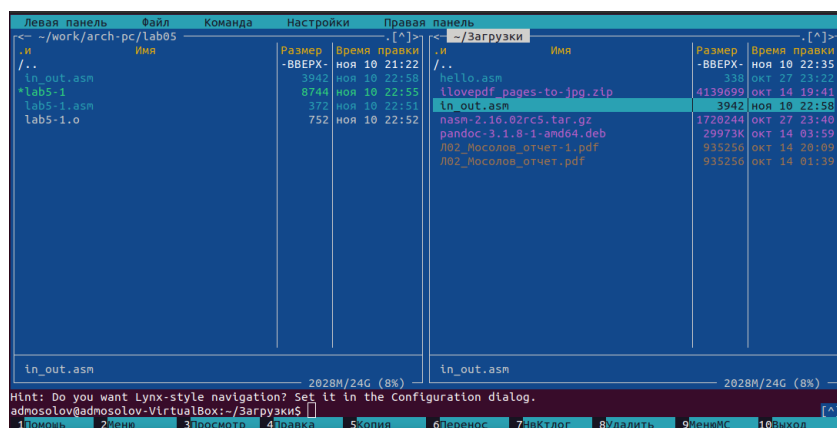


Рис. 2.11: Перемещаем скачанный файл

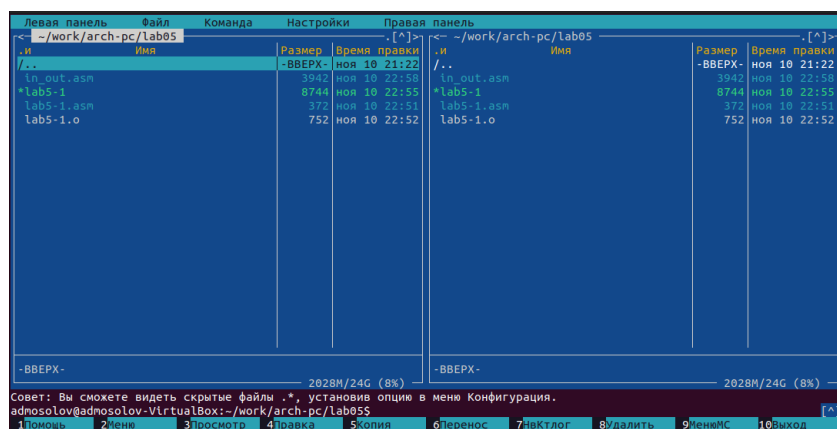


Рис. 2.12: Каталог lab05

С помощью функциональной клавиши `F5` создаём копию файла `lab5-1.asm` с именем `lab5-2.asm`. Выделяем файл `lab5-1.asm`, нажимаем клавишу `F6`, вводим имя файла `lab5-2.asm` и нажимаем клавишу `Enter`.

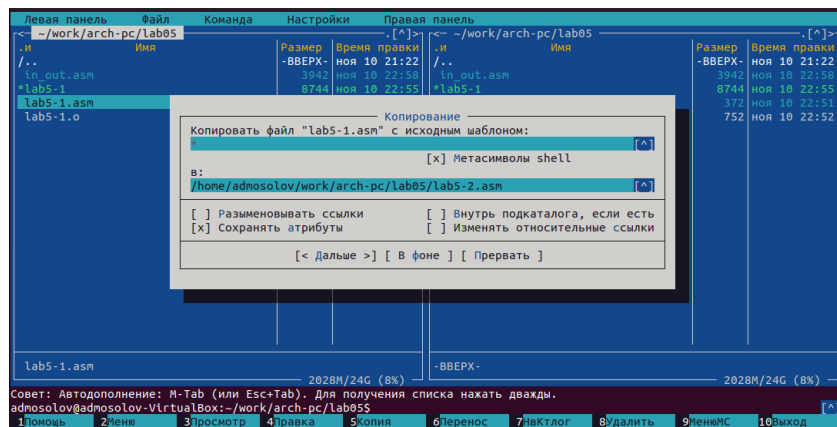


Рис. 2.13: Создаем копию файла с помощью F5

Исправляем текст программы в файле *lab5-2.asm* с использованием подпрограмм из внешнего файла *in_out.asm* (используем подпрограммы *sprintLF*, *sread* и *quit*) в соответствии с листингом 5.2.

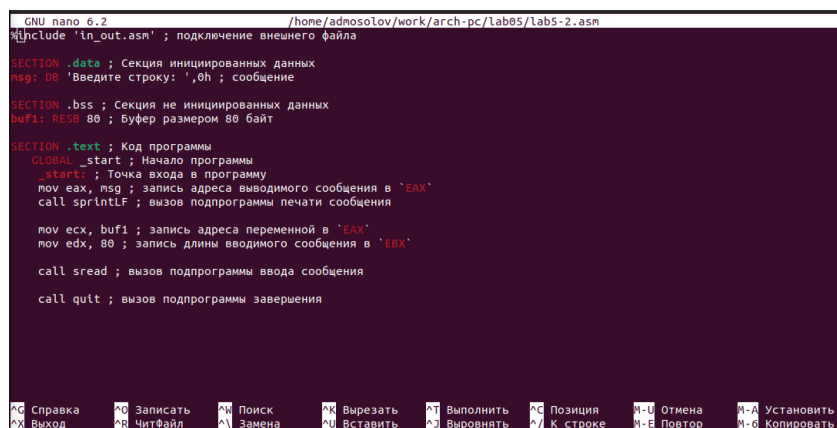


Рис. 2.14: Текст программы lab5-2

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

Разница заключается в том, что при использовании подпрограммы *sprintLF* при печати строк программа автоматически переводила курсор на следующую строку, а при *sprint* курсор смещается вправо и не переходит на следующую строку.

```
admosolov@admosolov-VirtualBox:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Александр Мосолов Денисович
```

Рис. 2.15: Используем подпрограмму `sprintLF`

```
admosolov@admosolov-VirtualBox:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
admosolov@admosolov-VirtualBox:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
admosolov@admosolov-VirtualBox:~/work/arch-pc/lab05$ ls
in_out.asm lab5-1 lab5-1.asm lab5-1.o lab5-2 lab5-2.asm lab5-2.o
admosolov@admosolov-VirtualBox:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Александр Мосолов Денисович
admosolov@admosolov-VirtualBox:~/work/arch-pc/lab05$
```

Рис. 2.16: Используем подпрограмму `sprint`

2.3 Задание для самостоятельной работы

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.

Создадим в домашнем каталоге папку *for* для выполнения этого задания. Перенесём туда файл *in_out.asm*. Назовём файл с программой *my_cod.asm*. Напишем текст программы.

```
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
```

Рис. 2.17: Текст программы *my_cod.asm*

Создайте копию файла *lab5-1.asm*. Внесите изменения в программу с использованием 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.

Назовём файл с программой *my_cod-2.asm*. Напишем текст программы.



```
GNU nano 6.2 /home/admosolov/for/my_cod-2.asm
#include 'in_out.asm'

SECTION .data
msg: DB 'Введите строку',10

SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:
    mov eax, msg
    call sprint

    mov ecx, buf1
    mov edx, 80

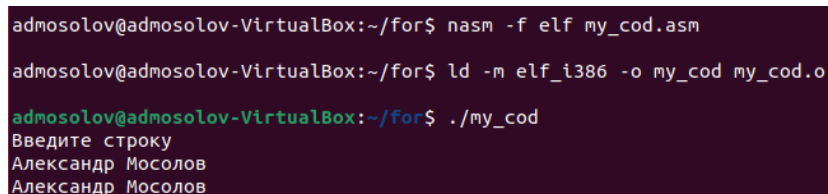
    call sread

    mov eax, buf1
    call sprint

    call quit
```

Рис. 2.18: Текст программы *my_cod-2.asm*

Транслируем текст программы (*my_cod.asm*). Для трансляции используем: *nasm -f elf my_cod.asm*. Выполняем компоновку с помощью команды *ld -m elf_i386 -o my_cod my_cod.o* и запускаем получившийся исполняемый файл, используя команду *./my_cod*.



```
admosolov@admosolov-VirtualBox:~/for$ nasm -f elf my_cod.asm
admosolov@admosolov-VirtualBox:~/for$ ld -m elf_i386 -o my_cod my_cod.o
admosolov@admosolov-VirtualBox:~/for$ ./my_cod
Введите строку
Александр Мосолов
Александр Мосолов
```

Рис. 2.19: Запускаем программу *my_cod*

Аналогичные команды используем для запуска программы *my_cod-2*.

```
admosolov@admosolov-VirtualBox:~/for$ nasm -f elf my_cod-2.asm
admosolov@admosolov-VirtualBox:~/for$ ld -m elf_i386 -o my_cod-2 my_cod-2.o
admosolov@admosolov-VirtualBox:~/for$ ./my_cod-2
Введите строку
Мосолов Александр
Мосолов Александр
admosolov@admosolov-VirtualBox:~/for$
```

Рис. 2.20: Запускаем программу my_cod-2

Переименовываем все файлы с именем *my_cod* на *lab5-1*, а с именем *my_cod-2* на *lab5-2* и переносим все файлы из каталога *for* в каталог *~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab05*. Удостоверимся в том, что правильно перенесли файлы с помощью команды *ls*.

```
admosolov@admosolov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ cd labs/lab05
admosolov@admosolov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab05$ ls
in_out.asm lab5-1 lab5-1.asm lab5-1.o lab5-2 lab5-2.asm lab5-2.o presentation report
```

Рис. 2.21: Просмотр каталога lab05

Загружаем файлы на *github*.

```
admosolov@admosolov-VirtualBox:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git add .
git commit -am 'feat(main): add files lab-5'
git push
```

Рис. 2.22: Загружаем файлы на github

3 Выводы

В ходе выполнения лабораторной работы были приобретены практические навыки работы в *Midnight Commander*, помимо этого были освоены инструкции языка ассемблера *mov* и *int*.