

# **Лабораторная работа №6**

**Архитектура компьютера**

Косолапов Матвей Эдуардович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>5</b>	<b>Выводы</b>	<b>16</b>
	<b>Список литературы</b>	<b>17</b>

## Список иллюстраций

4.1	Midnight Commander . . . . .	8
4.2	Каталог arch-pc . . . . .	9
4.3	Создание папки lab06 . . . . .	9
4.4	Папка lab06 . . . . .	10
4.5	Файл lab6-1.asm . . . . .	10
4.6	Сохранение изменений файла . . . . .	11
4.7	Проверка наличия содержимого файла . . . . .	11
4.8	Результат работы программы . . . . .	12
4.9	Файл in_out.asm . . . . .	12
4.10	Создание копии файла lab6-1.asm . . . . .	12
4.11	Изменение текста программы для работы с подпрограммами файла in_out.asm . . . . .	13
4.12	Проверяем работу файла . . . . .	13
4.13	Проверяем работу файла с изменением . . . . .	13
4.14	Изменение файла lab6-cop.asm под требуемые задачи . . . . .	14
4.15	Результат выполнения программы . . . . .	14
4.16	Изменение файла lab6-cop2.asm под требуемые задачи . . . . .	14
4.17	Результат выполнения программы . . . . .	15

## **Список таблиц**

# 1 Цель работы

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

## **2 Задание**

### **3 Теоретическое введение**

# 4 Выполнение лабораторной работы

1. Откроем Midnight Commander (рис. 4.1):

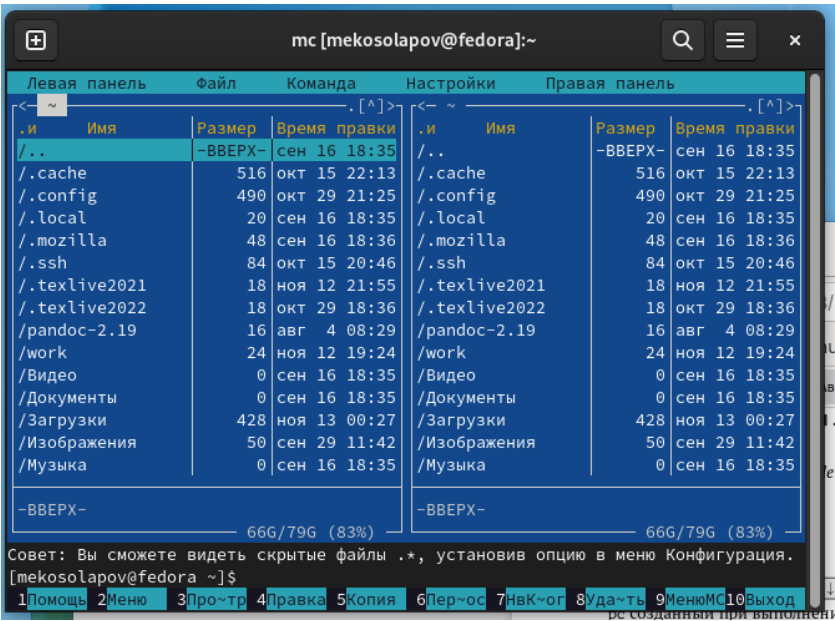


Рис. 4.1: Midnight Commander

2. Переходим в каталог ~/work/arch-pc (рис. 4.2):



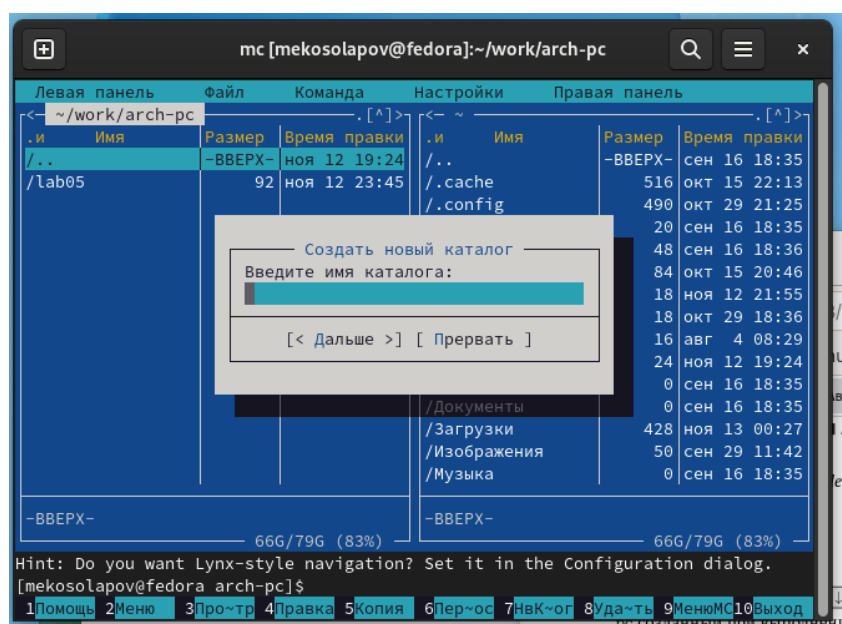


Рис. 4.2: Каталог arch-pc

3. Создаём папку lab06 и переходим в неё (рис. 4.3 - рис. 4.4):

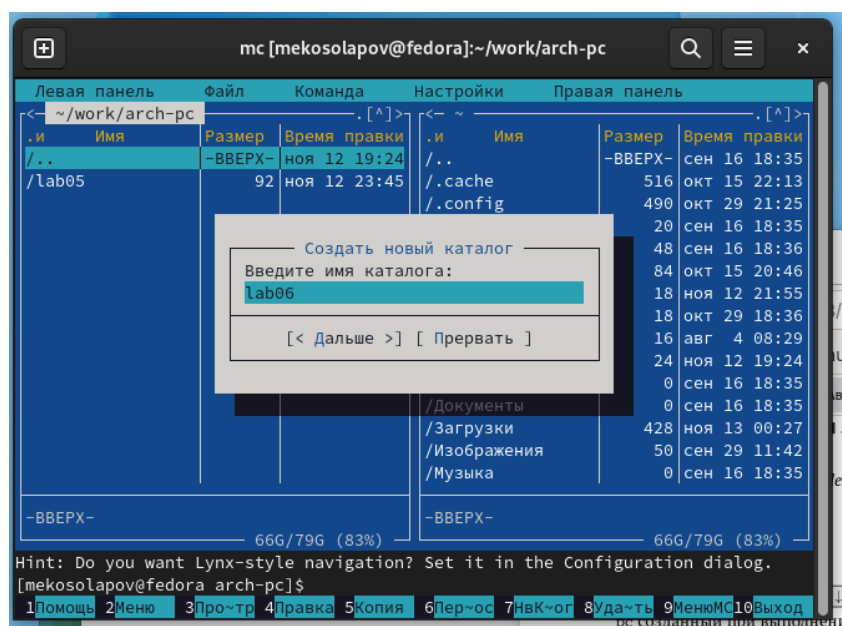


Рис. 4.3: Создание папки lab06

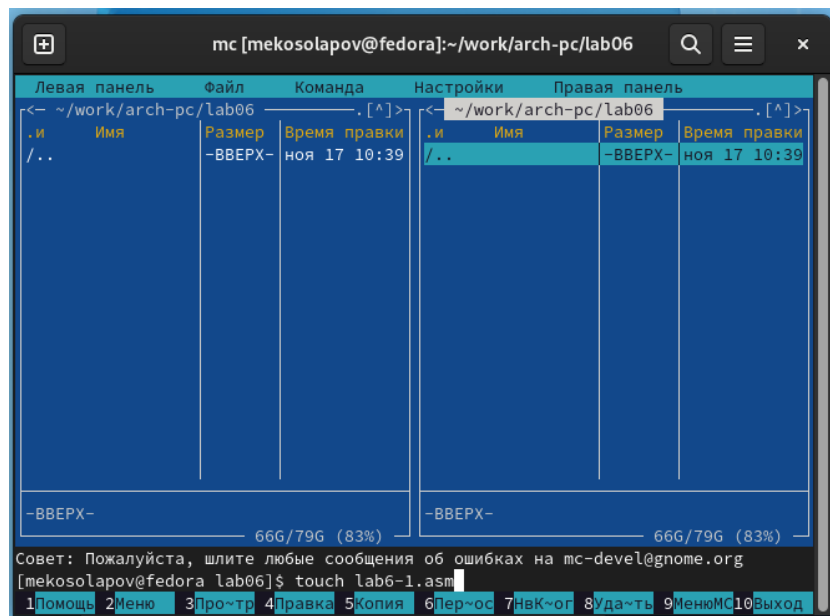


Рис. 4.4: Папка lab06

4. Создаём файл lab6-1.asm (рис. 4.5):

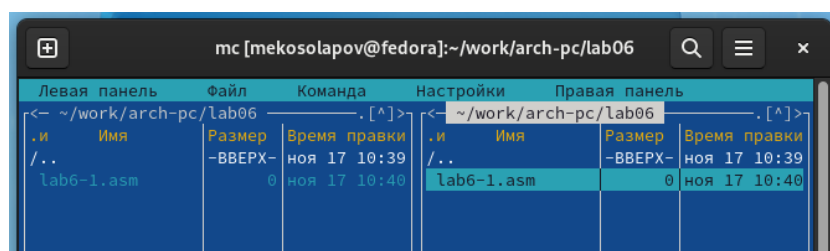


Рис. 4.5: Файл lab6-1.asm

5. Открываем созданный файл, вводим в него предложенную программу, сохраняем и закрываем (рис. 4.6):

```

lab6-1.asm      [-M--] 61 L: [ 15+18  33/ 36] *(2282/2463b) 0010 0x00A  [*][X]
;----- 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

Сохранить файл
Подтверждаете запись файла "/home/mekosolapov/work/arch-pc/lab06/lab6-1.asm"?

[ Сохранить ] [ Прервать ]

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
;----- Системный вызов 'exit' -----
; После вызова инструкции 'int 80h' программа завершит работу
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра
1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перейти 7Поиск 8Вдаться 9Меню 10Выход

```

Рис. 4.6: Сохранение изменений файла

6. Откроем файл и проверим, что файл содержит текст программы (рис. 4.7):

```

/home/mekosolapov/work/~ch-pc/lab06/lab6-1.asm      2463/2463      100%
;----- 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
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
;----- Системный вызов 'exit' -----
; После вызова инструкции 'int 80h' программа завершит работу
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра
1Помощь 2Разрыв 3Выход 4Нех 5Перейти 6 7Поиск 8Исходный 9Формат 10Выход

```

Рис. 4.7: Проверка наличия содержимого файла

7. Переводим текст программы в исполняемый файл. Запускаем файл и вводим своё ФИО (рис. 4.8)

```
[mekosolapov@fedora lab06]$ nasm -f elf lab6-1.asm
lab6-1.asm:26: warning: label alone on a line without a colon might be in error
[-w+label-orphan]
[mekosolapov@fedora lab06]$ ls
lab6-1.asm  lab6-1.o
[mekosolapov@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[mekosolapov@fedora lab06]$ ls
lab6-1  lab6-1.asm  lab6-1.o
[mekosolapov@fedora lab06]$ ./lab6-1
Введите строку:
Косолапов Матвей Эдуардович
[mekosolapov@fedora lab06]$
```

Рис. 4.8: Результат работы программы

9. Скачиваем файл `in_out.asm`, переносим файл в каталог с программой (рис. 4.9)

The image shows a file manager window with a dark theme. At the top, there's a title bar with a home icon, the path 'mc [mekosolapov@fedora]:~/work/arch-pc/lab06', a search icon, and a menu icon. Below the title bar, there are two panels. The left panel has a header with 'Левая панель', 'Файл', 'Команда', 'Настройки', and 'Правая панель'. It shows a list of files: '..', '..', 'in\_out.asm', '\*lab6-1', 'lab6-1.asm', and 'lab6-1.o'. The file 'in\_out.asm' is highlighted. The right panel shows a similar list of files: '..', '..', 'lab6-1.o', 'lab6-1.asm', '\*lab6-1', and 'in\_out.asm'. The file 'in\_out.asm' is also highlighted. The columns in both panels are 'Имя', 'Размер', and 'Время правки'.

Рис. 4.9: Файл `in_out.asm`

10. Создаём копию файла `lab6-1.asm` с именем `lab6-2.asm` (рис. 4.10)

Копирование

Копировать файл "lab6-1.asm" с исходным шаблоном:

[x] Метасимволы shell

в:

/home/mekosolapov/lab6-2.asm

[ ] Разыменовывать ссылки

[x] Сохранять атрибуты

[ ] Внутрь подкаталога, если есть

[ ] Изменять относительные ссылки

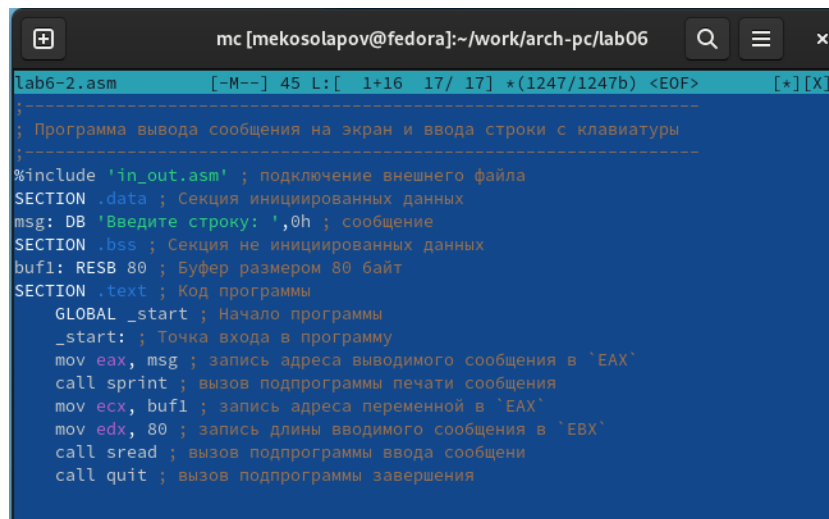
[< Дальше >]

[ В фоне ]

[ Прервать ]

Рис. 4.10: Создание копии файла `lab6-1.asm`

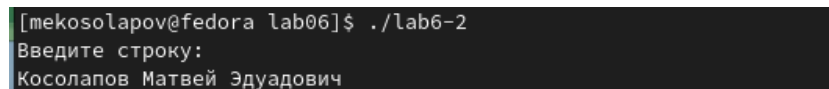
11. Меняем текст программы для использования подпрограмм из внешнего файла `in_out.asm` (рис. 4.11):



```
lab6-2.asm [-M--] 45 L: [ 1+16 17/ 17] *(1247/1247b) <EOF> [*][X]
;
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;
-----
%include 'in_out.asm' ; подключение внешнего файла
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 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
```

Рис. 4.11: Изменение текста программы для работы с подпрограммами файла in\_out.asm

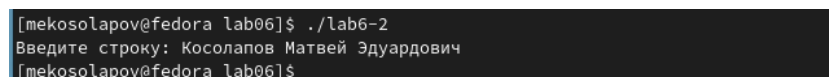
12. Создаём исполняемый файл и проверяем его работу (рис. 4.12):



```
[mekosolapov@fedora lab06]$ ./lab6-2
Введите строку:
Косолапов Матвей Эдуардович
```

Рис. 4.12: Проверяем работу файла

13. Меняем подпрограмму sprintLF на sprint в тексте программы, смотрим результат. (рис. 4.13):



```
[mekosolapov@fedora lab06]$ ./lab6-2
Введите строку: Косолапов Матвей Эдуардович
[mekosolapov@fedora lab06]$
```

Рис. 4.13: Проверяем работу файла с изменением

Разница в том, что с sprintLF ввод данных осуществляется на новой строке, а в sprint на той же.

14. Создаём копию файла lab6-1.asm с именем lab6-cop.asm, меняем его, чтобы он выводил введённые данные (рис. 4.14):

```

байт→
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
;-----Вызов принт-----
mov eax,4
mov ebx,1
mov ecx,buf1
mov edx,80
int 80h
;----- Системный вызов 'exit' -----
; После вызова инструкции 'int 80h' программа завершит работу
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра

```

Рис. 4.14: Изменение файла lab6-cop.asm под требуемые задачи

15. Создаём исполнительный файл и проверяем его работу (рис. 4.15):

```

[mekosolapov@fedora lab06]$ ./lab6-cop
Введите строку:
Косолапов Матвей Эдуардович
Косолапов Матвей Эдуардович
[mekosolapov@fedora lab06]$

```

Рис. 4.15: Результат выполнения программы

16. Создаём копию файла lab6-2.asm с именем lab6-cop2.asm, меняем его, чтобы он выводил введённые данные (рис. 4.16):

```

; 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
mov eax, buf1 ; запись адреса конечного выводимого сообщения в 'EAX'
call sprint ; вызов вывода сообщения
call quit ; вызов подпрограммы завершения

```

Рис. 4.16: Изменение файла lab6-cop2.asm под требуемые задачи

17. Создаём исполняемый файл и проверяем его работу (рис. 4.17):

```
[mekosolapov@fedora lab06]$ nasm -f elf lab6-cop2.asm
[mekosolapov@fedora lab06]$ ld -m elf_i386 -o lab6-cop2 lab6-cop2.o
[mekosolapov@fedora lab06]$ ls
in_out.asm  lab6-1.o    lab6-2.o    lab6-cop2.asm  lab6-cop.o
lab6-1      lab6-2     lab6-cop    lab6-cop2.o
lab6-1.asm  lab6-2.asm lab6-cop2   lab6-cop.asm
[mekosolapov@fedora lab06]$ ./lab6-cop2
Введите строку: Косолапов Матвей Эдуардович
Косолапов Матвей Эдуардович
[mekosolapov@fedora lab06]$
```

Рис. 4.17: Результат выполнения программы

18. Отлично! Всё работает прекрасно!

## 5 Выводы

В ходе данной лабораторной работы я научился писать программы в Midnight Commander.



## **Список литературы**