

Лабораторная работа №6

Щетинин Даниил Николаевич

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
4	Задание для самостоятельной работы	8
5	Выводы	14

Список иллюстраций

4.1	мс, создание каталога	9
4.2	файл lab6-1.asm	10
4.3	исполнение файла	10
4.4	перемещение файла	11
4.5	копирование файла с новым именем	11
4.6	работа файла lab6-2	11
4.7	lab6-2, sprint	12
4.8	код lab6-11	12
4.9	код lab6-22	13
4.10	исполнение программ	13

1 Цель работы

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

2 Задание

Используя Midnight Commander (MC) и `mov` и `int` сделать базовые программы

3 Выполнение лабораторной работы

Шаг 1

Открываем Мс с помощью команды `mc` в терминале, перейдём в рабочий каталог и создадим каталог `lab06`, где мы будем проводить дальнейшую работу, также создадим (с помощью команды `touch`) файл `lab6-1.asm`, редактируем его (рис. 4.1)

Шаг 2

вводим в `lab6-1.asm` текст программы

(рис. 4.2)

помощью клавиш F2, F3 сохраним изменения, проверим их наличие

Шаг 3

скомпилируем исполняемый файл и запустим получившийся файл:

(рис. 4.3)

Шаг 4

Скопируем файл `in_out.asm` в каталог `arch-rc` с помощью клавиши F5

(рис. 4.4)

Скопируем файл `lab6-1.asm` как `lab6-2.asm` с помощью клавиши F6

(рис. 4.5)

Шаг 5

вводим в `lab6-2.asm` текст программы

(рис. 4.6)

Выполним компоновку объектного файла и запустим получившийся исполняемый файл, а также заменим подпрограмму `sprintLF` на `sprint`:

(рис. 4.7)

во втором файле отсутствует функция перевода строки после вывода сообщения на экран, поэтому введённое сообщение будет на одном уровне с “Введите строку:”

4 Задание для самостоятельной работы

Шаг 1

Копируем файл lab6-1.asm как lab6-11.asm и изменим его для того, чтобы на экран выводилась введённая строка:

(рис. 4.8)

Последний параграф отвечает за:

- 1 Системный вызов для записи
- 2 Стандартный вывод
- 3 Адрес строки buf1 (ранее введённое сообщение)
- 4 Длину строки

После вызова инструкции `int 80h` на экране будет сообщение buf1, что и является введённым текстом

(рис. 4.10)

Шаг 2

скопируем lab6-2 как lab6-22:

(рис. 4.9)

Здесь мы вписываем наше сообщение в регистр `eax` и используем подпрограмму `sprint` для того чтобы вывести на экран введённое сообщение (buf1)

(рис. 4.10)

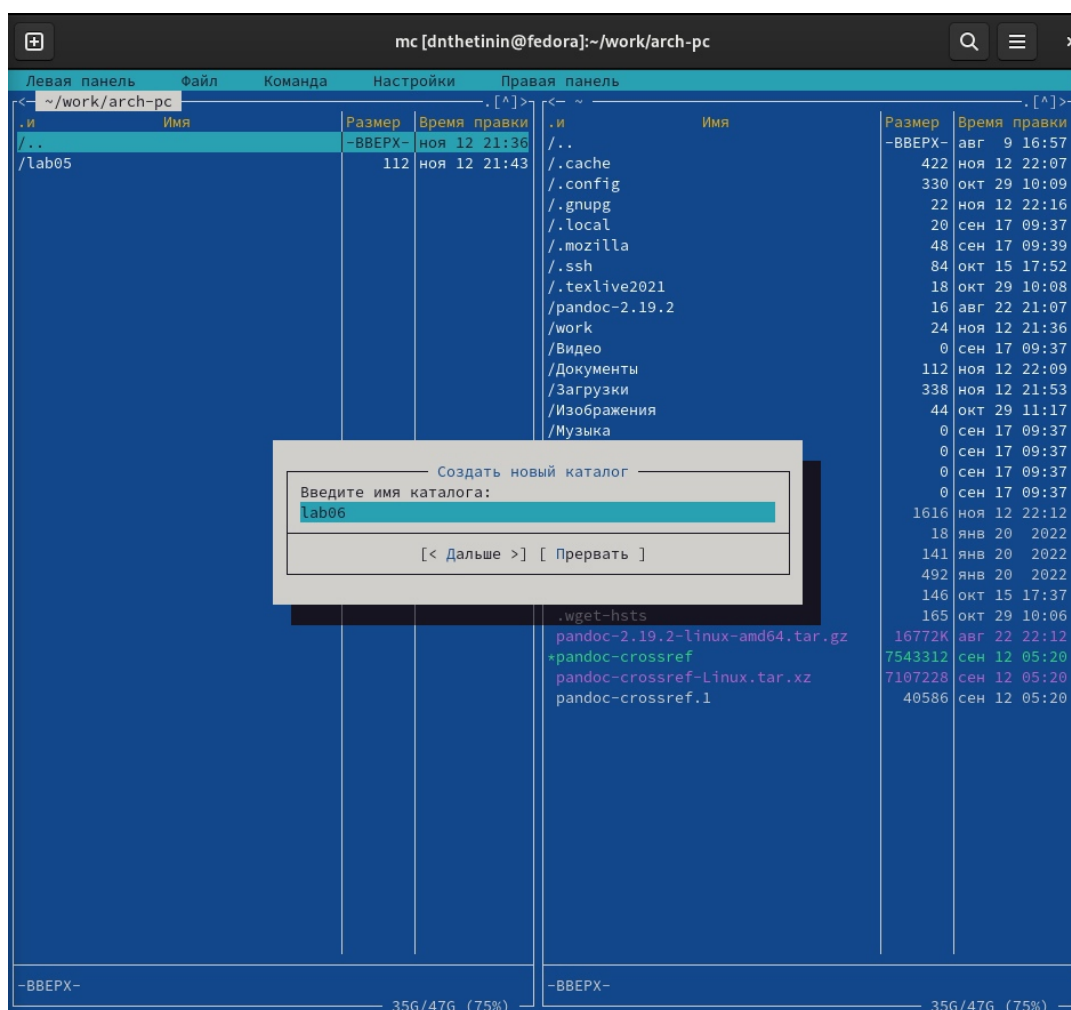


Рис. 4.1: mc, создание каталога

```

lab6-1.asm      [----]  0 L:[ 1+21 22/ 22] *(268 / 268b) <EOF>
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.

```

Рис. 4.2: файл lab6-1.asm

```

[dnthetinin@fedora ~]$ mc

[dnthetinin@fedora lab06]$ touch lab6-1.asm

[dnthetinin@fedora lab06]$ nasm -f elf lab6-1.asm

[dnthetinin@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o

[dnthetinin@fedora lab06]$ ./lab6-1
Введите строку:
Щетинин Даниил Николаевич

```

Рис. 4.3: исполнение файла

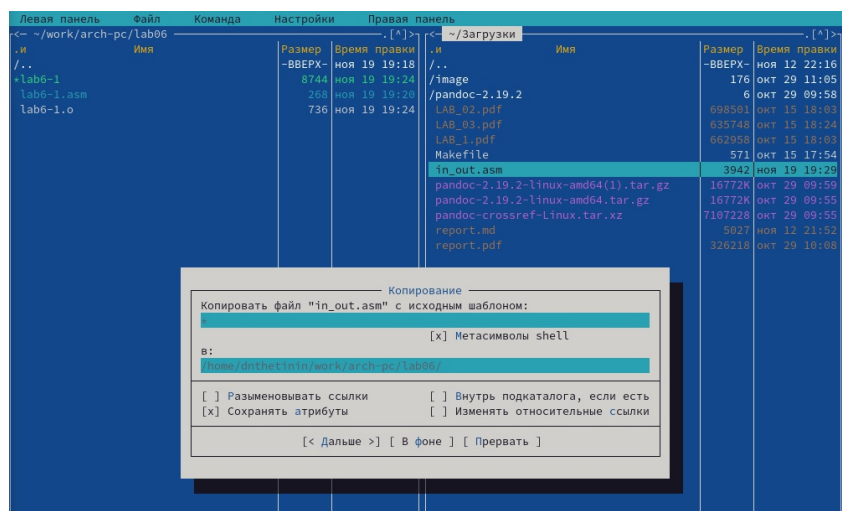


Рис. 4.4: перемещение файла

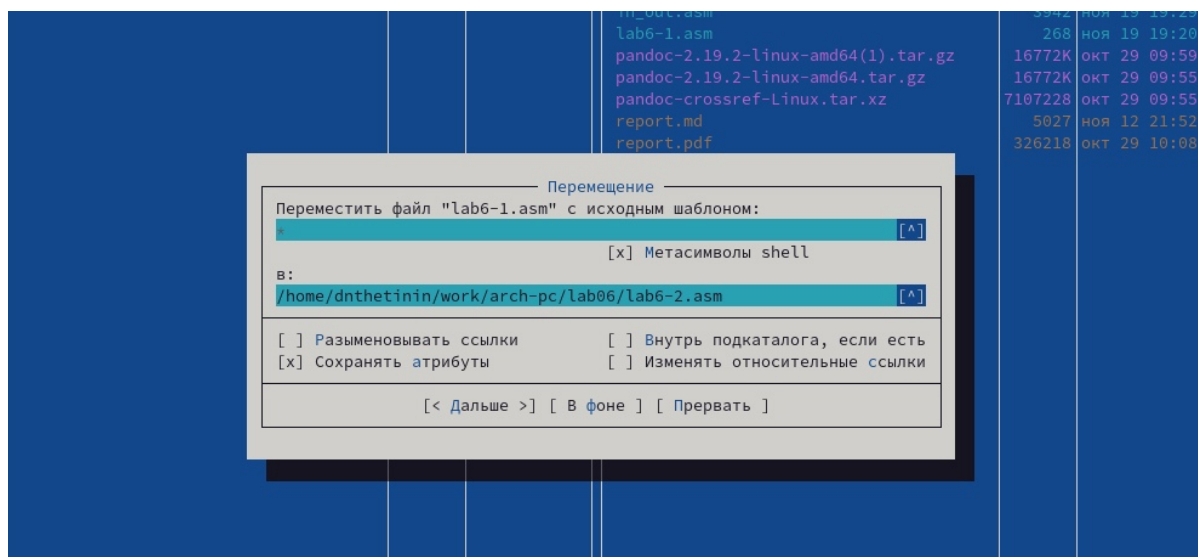


Рис. 4.5: копирование файла с новым именем

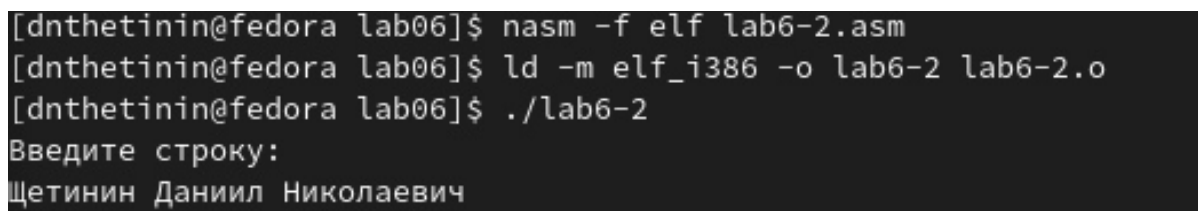


Рис. 4.6: работа файла lab6-2

```
[dnthetinin@fedora lab06]$ nasm -f elf lab6-2.asm
[dnthetinin@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[dnthetinin@fedora lab06]$ ./lab6-2
Введите строку: Щетинин Даниил Николаевич
```

Рис. 4.7: lab6-2, sprint

```
lab6-11.asm      [----]  0 L:[ 1+ 0  1/ 32] *(0  / 352b) 0083 0x053
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. 4.8: код lab6-11

```
lab6-22.asm      [----]  0 L:[ 1+ 0  1/ 20] *(0  / 256b) 0010 0x00A

%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.
```

Рис. 4.9: код lab6-22

```
[dnthetinin@fedora lab06]$ nasm -f elf lab6-11.asm
[dnthetinin@fedora lab06]$ ld -m elf_i386 -o lab6-11 lab6-11.o
[dnthetinin@fedora lab06]$ ./lab6-11
Введите строку:
Щетинин Даниил
Щетинин Даниил
[dnthetinin@fedora lab06]$ nasm -f elf lab6-22.asm
[dnthetinin@fedora lab06]$ ld -m elf_i386 -o lab6-22 lab6-22.o
[dnthetinin@fedora lab06]$ ./lab6-22
Введите строку: Щетинин Даниил Николаевич
Щетинин Даниил Николаевич
```

Рис. 4.10: исполнение программ

5 Выводы

Я смог использовать tc для навигации по системе, а также языки ассемблера `mov` и `int` для написания программ