

Отчёт по лабораторной работе 5

Архитектура компьютеров и операционные системы”

Дрожжанова А.Д. НБИбд-01-23

Содержание

1	Цель работы	5
2	Задания	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
4.1	Выполнение заданий для самостоятельной работы.	18
5	Выводы	22
6	Источники	23

Список иллюстраций

4.1	Запуск Midnight Commander	8
4.2	Создание каталога	9
4.3	Создание файла lab05-1.asm	10
4.4	Программа в файле lab05-1.asm	11
4.5	Просмотр файла lab05-1.asm	12
4.6	Запуск программы lab05-1.asm	13
4.7	Копирование файла in_out.asm	14
4.8	Копирование файла lab05-1.asm	15
4.9	Программа в файле lab05-2.asm	16
4.10	Запуск программы lab05-2.asm	16
4.11	Программа в файле lab05-2.asm	17
4.12	Запуск программы lab05-2.asm	17
4.13	Копирование файла lab05-1.asm	18
4.14	Программа в файле lab05-3.asm	19
4.15	Запуск программы lab05-3.asm	20
4.16	Копирование файла lab05-2.asm	20
4.17	Программа в файле lab05-4.asm	21
4.18	Запуск программы lab05-4.asm	21

Список таблиц

1 Цель работы

Целью работы является приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера `mov` и `int`.

2 Задания

1. Освоить возможности Midnight Commander
2. Изучить примеры программ с использованием внешнего файла `in_out.asm`
3. Выполнить задание по программе
4. Подготовить отчет и загрузить на GitHub

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.

Программа на языке ассемблера NASM, как правило, состоит из трёх секций: секция кода программы (SECTION .text), секция инициированных (известных во время компиляции) данных (SECTION .data) и секция неинициализированных данных (тех, под которые во время компиляции только отводится память, а значение присваивается в ходе выполнения программы) (SECTION .bss).

Инструкция языка ассемблера mov предназначена для дублирования данных источника в приёмнике. В общем виде эта инструкция записывается в виде mov dst,src Здесь операнд dst — приёмник, а src — источник

Инструкция языка ассемблера int предназначена для вызова прерывания с указанным номером. В общем виде она записывается в виде int n Здесь n — номер прерывания, принадлежащий диапазону 0–255

4 Выполнение лабораторной работы

1. Открыла Midnight Commander, с помощью клавишь со стрелками и Enter перехожу в каталог ~/work/arch-pc. Далее нажимаю F7 и создаю каталог lab05

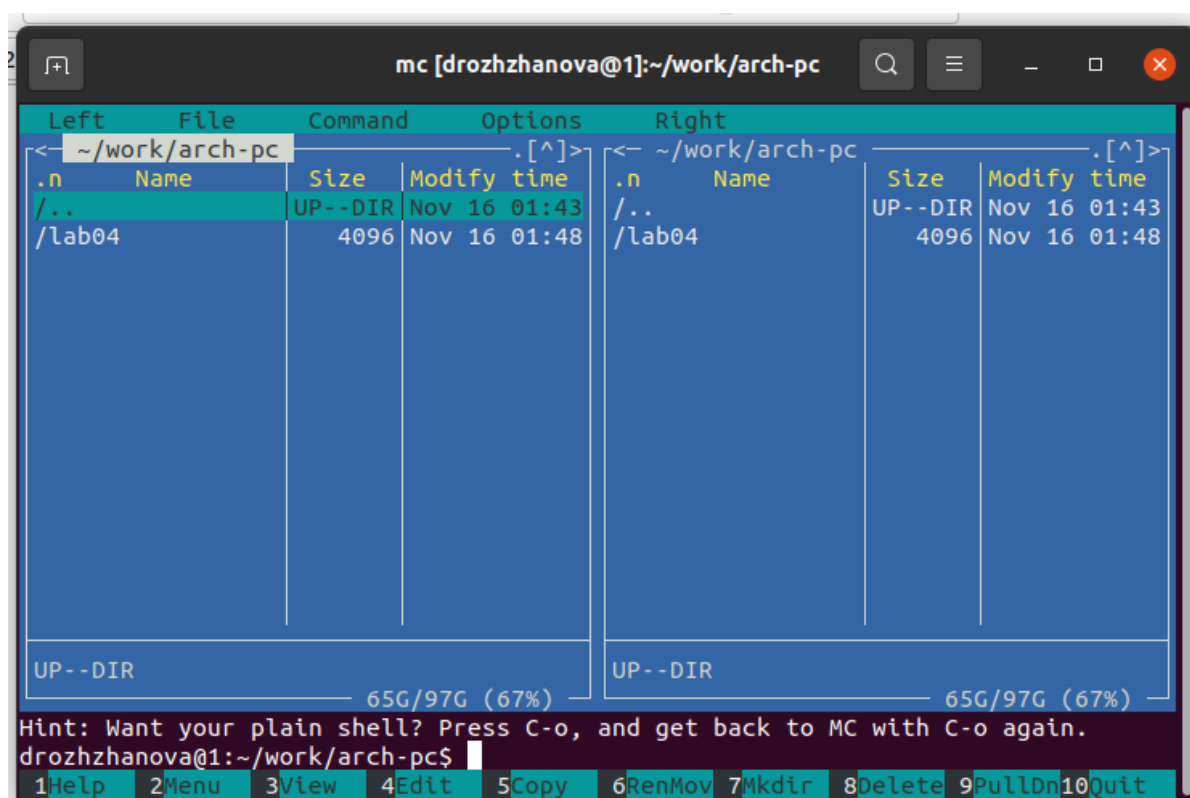


Рис. 4.1: Запуск Midnight Commander

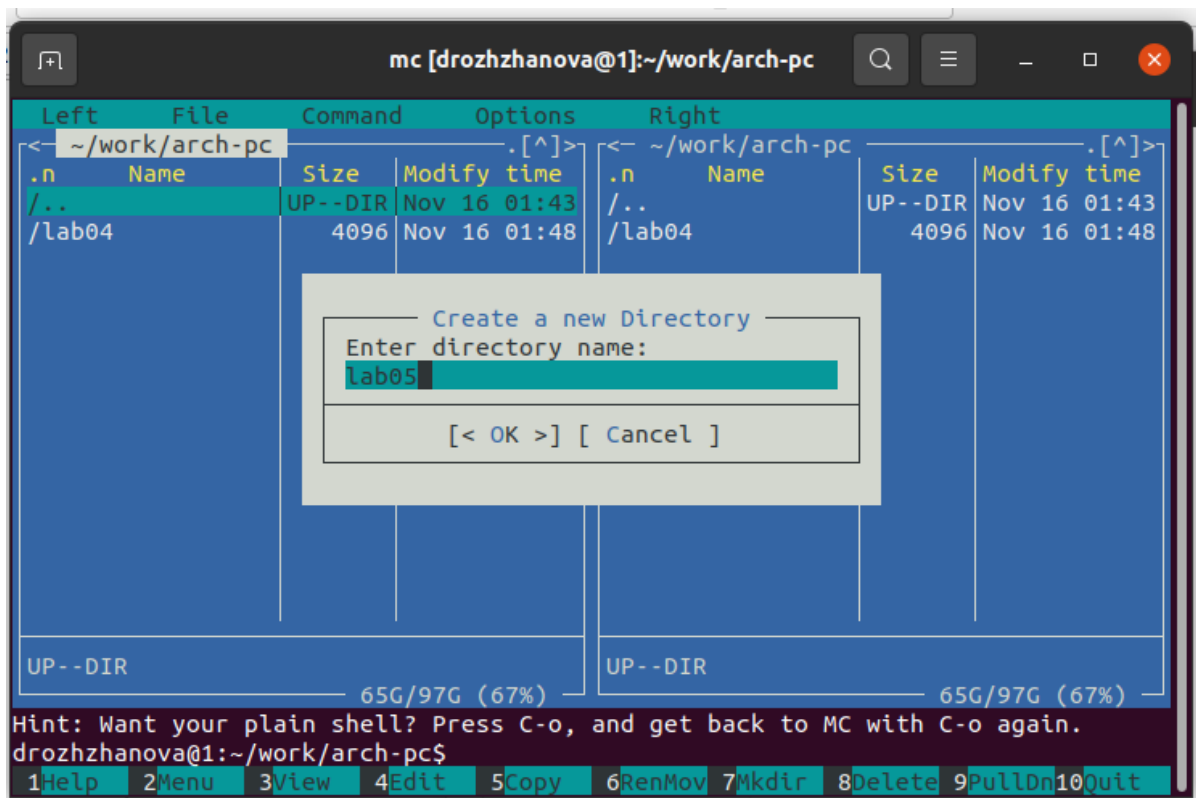


Рис. 4.2: Создание каталога

2. При помощи `touch` создала файл `lab05-1.asm`

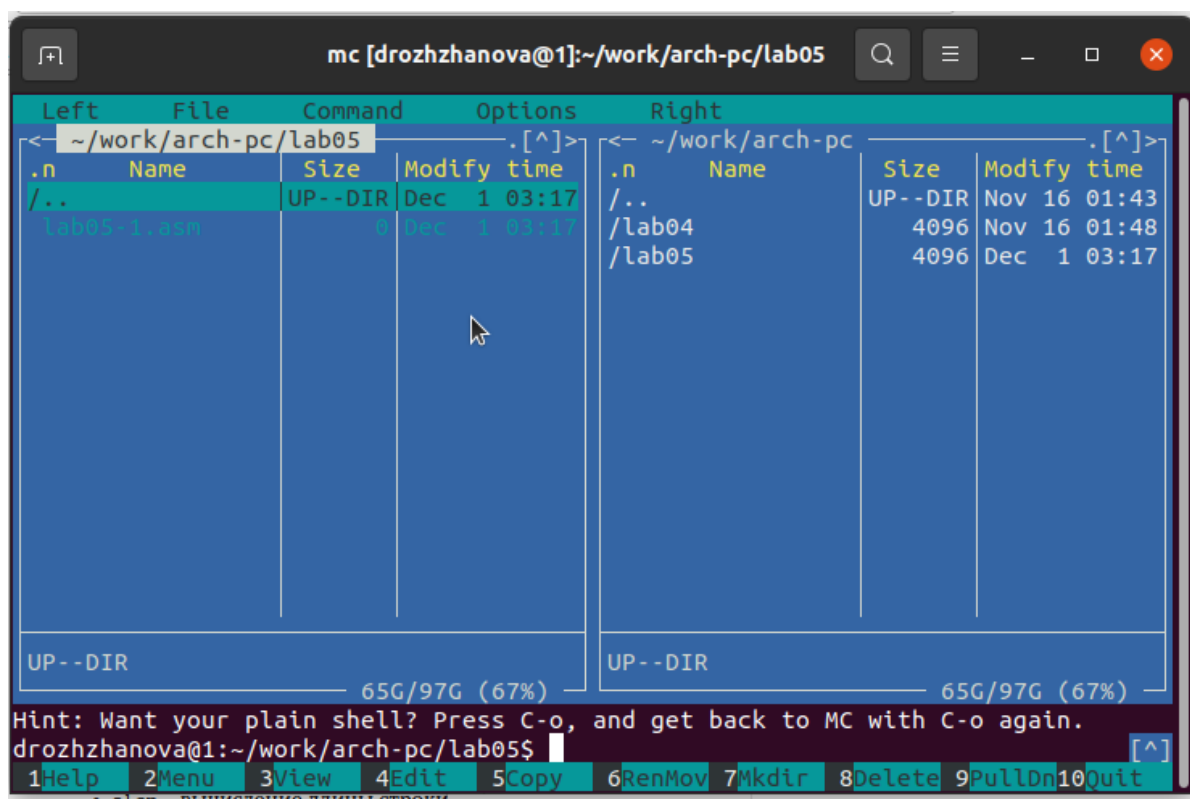


Рис. 4.3: Создание файла lab05-1.asm

3. Открыла файл на редактирование клавишей F4, выбрала редактор mceditor, написала код программы из задания.

```
12 mc [drozhzhanova@1]:~/work/arch-pc/lab05
/home/dr~05-1.asm [----] 0 L:[ 2+21 23/ 23] *(278 / 2
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. 4.4: Программа в файле lab05-1.asm

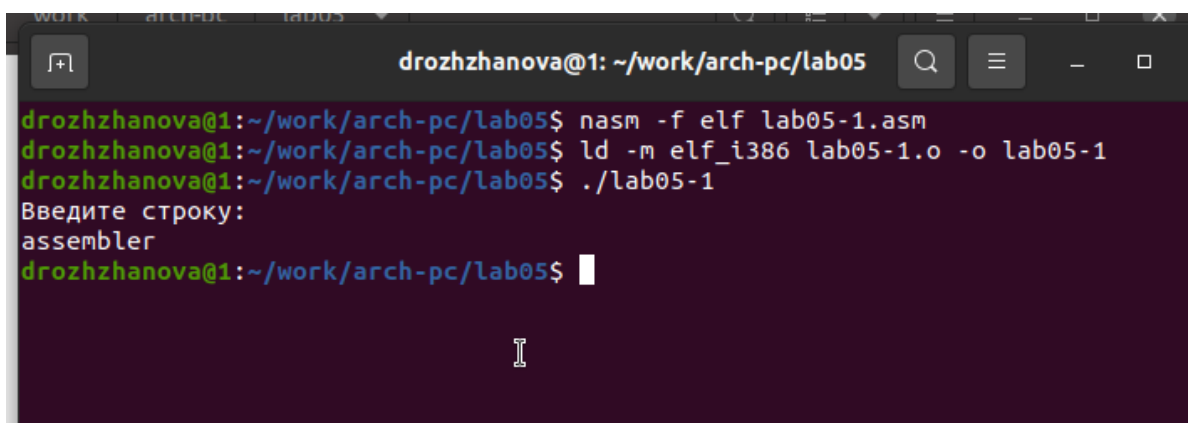
4. Открыла файл на просмотр клавишей F3 и проверила, что он содержит набранный код.

```
12 mc [drozhzhanova@1]:~/work/arch-pc/lab05
/home/drozhzhanova/work~h-pc/lab05/lab05-1.asm 278
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. 4.5: Просмотр файла lab05-1.asm

5. Транслировала файл программы в объектный файл, выполнила компоновку объектного файла, получила исполняемый файл программы и проверила ее работу.

A terminal window with a dark background and light-colored text. The window title is "drozhzhanova@1: ~/work/arch-pc/lab05". The terminal shows the following commands and output:

```
drozhzhanova@1:~/work/arch-pc/lab05$ nasm -f elf lab05-1.asm
drozhzhanova@1:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-1.o -o lab05-1
drozhzhanova@1:~/work/arch-pc/lab05$ ./lab05-1
Введите строку:
assembler
drozhzhanova@1:~/work/arch-pc/lab05$
```

Рис. 4.6: Запуск программы lab05-1.asm

6. 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.

Скачала файл `in_out.asm` и разместила его в рабочем каталоге. Для копирования используется клавиша F5. Для перемещения используется клавиша F6.

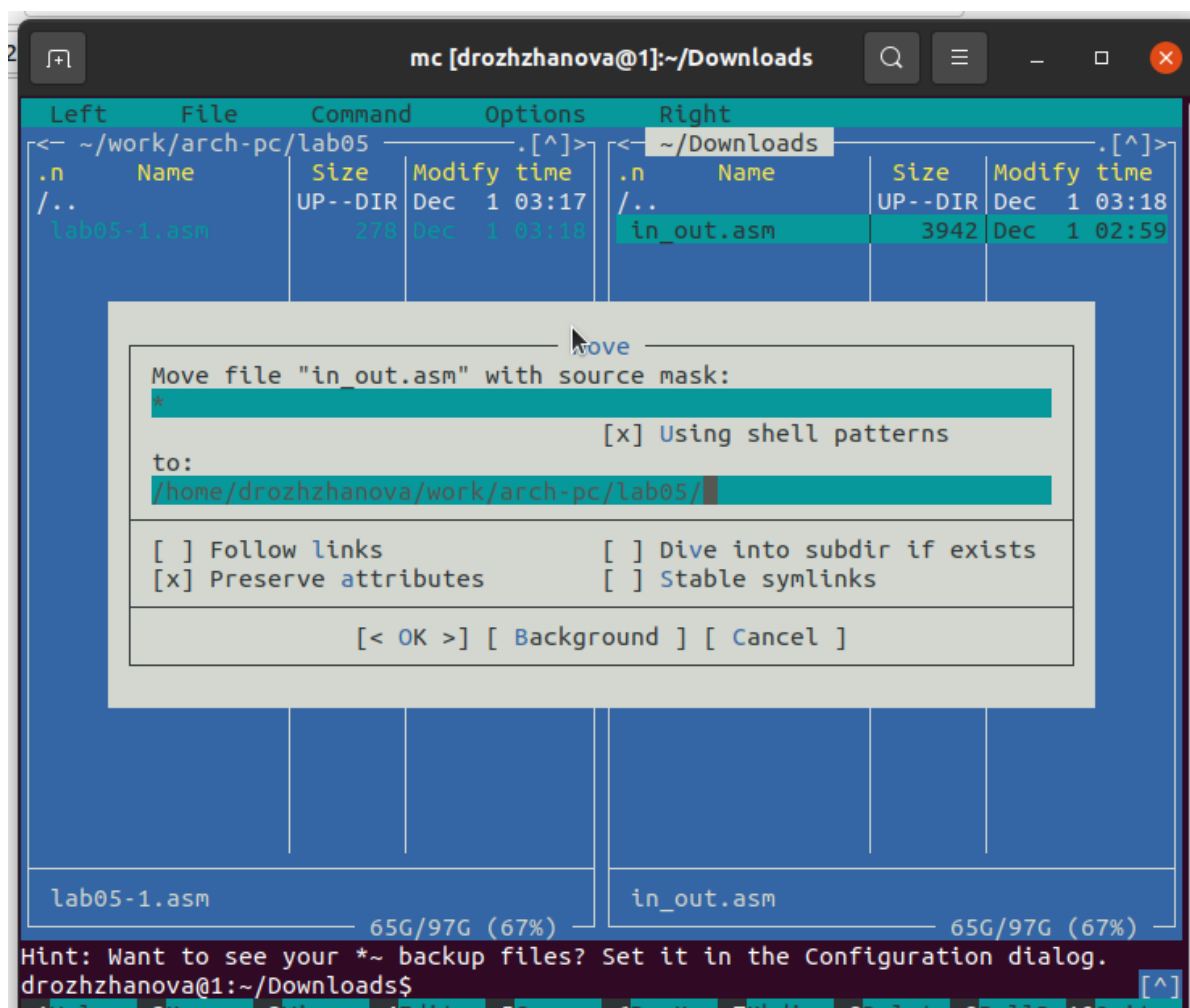


Рис. 4.7: Копирование файла in_out.asm

7. Скопировала lab05-1.asm в lab05-2.asm.

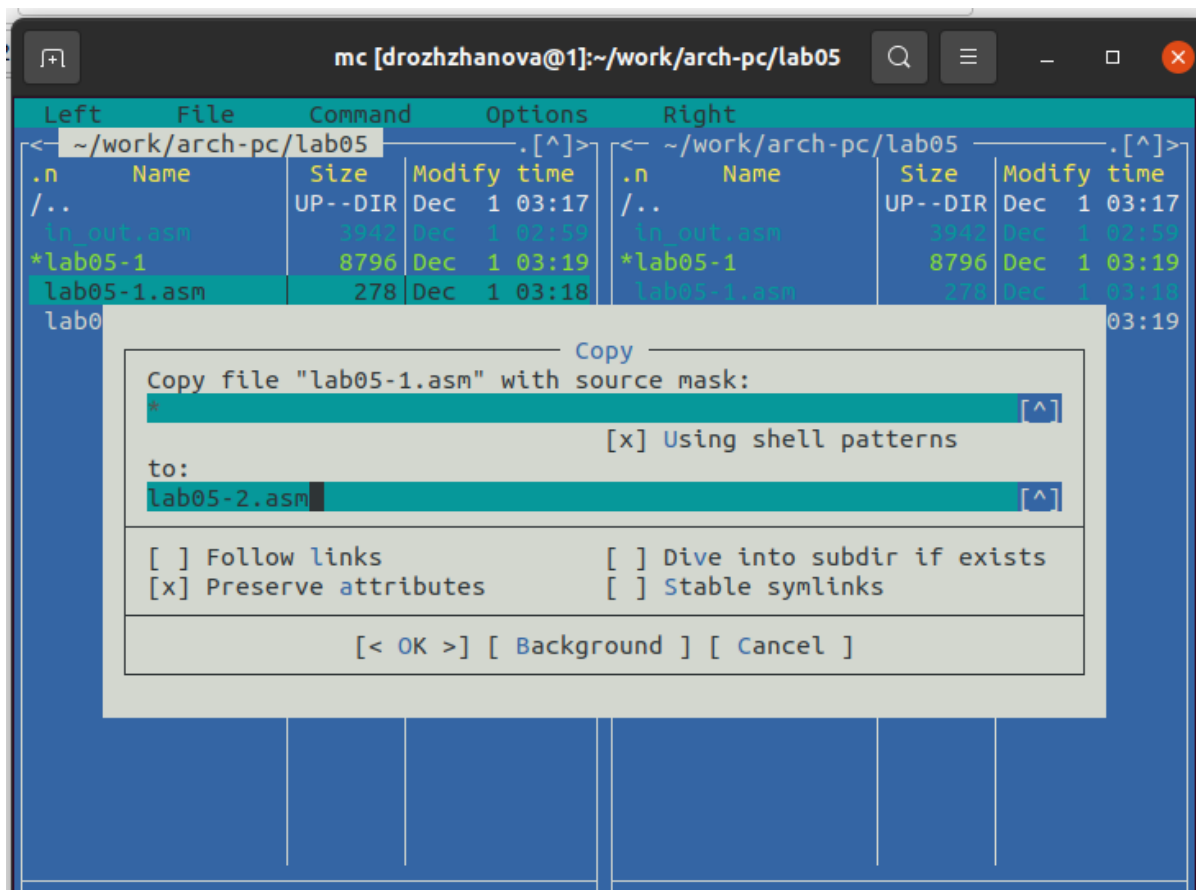


Рис. 4.8: Копирование файла lab05-1.asm

8. Написала код программы lab05-2.asm с использованием подпрограмм из внешнего файла in_out.asm.

```
12 mc [drozhzhanova@1]:~/work/arch-pc/la
/home/dr~05-2.asm [----] 0 L: [ 1+14 15/ 15] *(210
#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 sprintLF
mov ecx, buf1
mov edx, 80
call sread
call quit
```

Рис. 4.9: Программа в файле lab05-2.asm

```
drozhzhanova@1:~/work/arch-pc/lab05$
drozhzhanova@1:~/work/arch-pc/lab05$ nasm -f elf lab05-2.asm
drozhzhanova@1:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2
drozhzhanova@1:~/work/arch-pc/lab05$ ./lab05-2
Введите строку:
assembler
drozhzhanova@1:~/work/arch-pc/lab05$
```

Рис. 4.10: Запуск программы lab05-2.asm

9. В файле lab5-2.asm заменила подпрограмму sprintLF на sprint. Заново собрала исполняемый файл. Теперь после вывода строки она не завершается символом перехода на новую строку.


```
12 mc [drozhzhanova@1]:~/work/arch-  
/home/dr~05-2.asm [----] 11 L:[ 1+ 9 10/ 15]  
%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  
call quit
```

Рис. 4.11: Программа в файле lab05-2.asm

```
assembler  
drozhzhanova@1:~/work/arch-pc/lab05$ nasm -f elf lab05-2.asm  
drozhzhanova@1:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2  
drozhzhanova@1:~/work/arch-pc/lab05$ ./lab05-2  
Введите строку: assembler  
drozhzhanova@1:~/work/arch-pc/lab05$
```

Рис. 4.12: Запуск программы lab05-2.asm

4.1 Выполнение заданий для самостоятельной работы.

Скопировала программу lab05-1.asm и изменила код, так чтобы она работала по следующему алгоритму:

- вывести приглашение типа “Введите строку:”;
- ввести строку с клавиатуры;
- вывести введённую строку на экран.

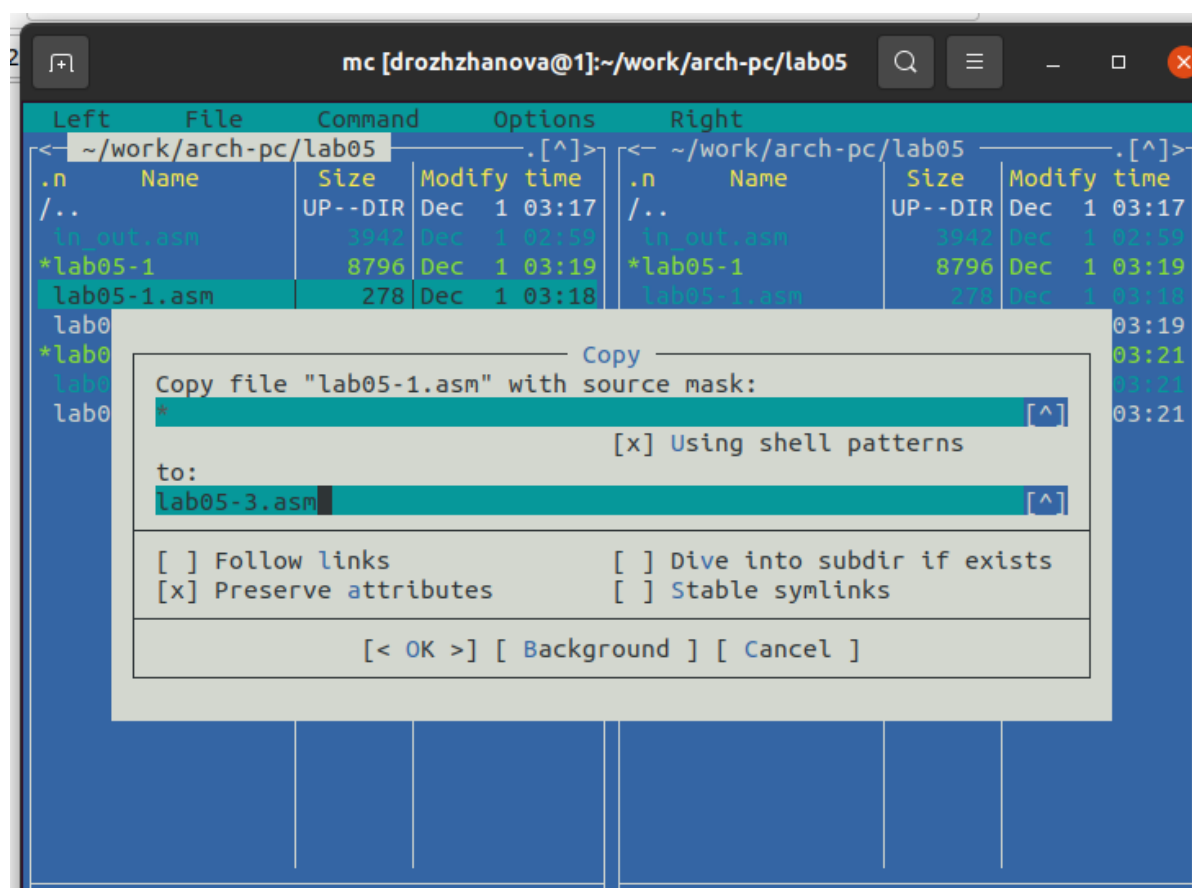


Рис. 4.13: Копирование файла lab05-1.asm

```
mc [drozhzhanova@1]:~/work/arch
/home/dr~05-3.asm [----] 12 L:[ 1+21 22/ 28]
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. 4.14: Программа в файле lab05-3.asm

```

drozhzhanova@1:~/work/arch-pc/lab05$
drozhzhanova@1:~/work/arch-pc/lab05$ nasm -f elf lab05-3.asm
drozhzhanova@1:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-3.o -o lab05-3
drozhzhanova@1:~/work/arch-pc/lab05$ ./lab05-3
Введите строку:
assembler
assembler
drozhzhanova@1:~/work/arch-pc/lab05$

```

Рис. 4.15: Запуск программы lab05-3.asm

Аналогично скопировала программу lab05-2.asm и изменила код, но теперь использовал подпрограммы из файла in_out.asm.

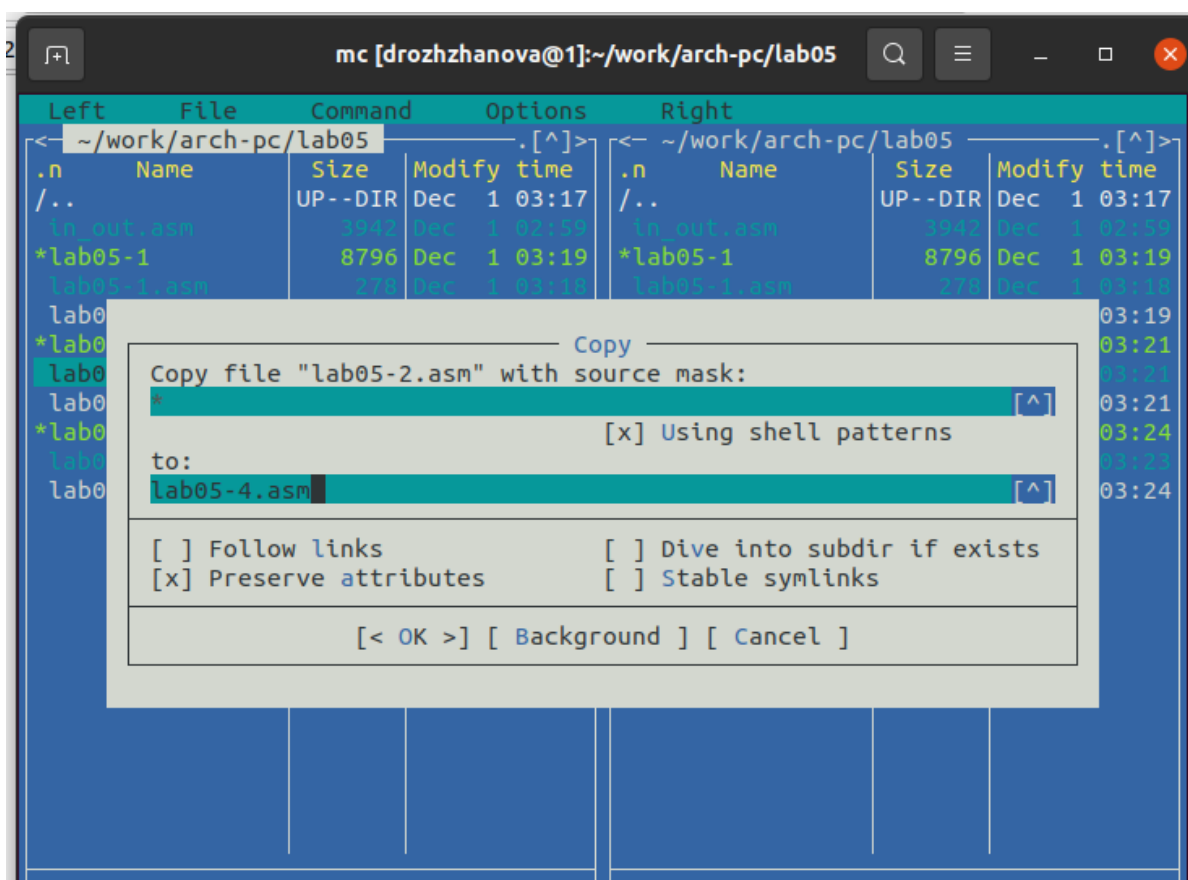
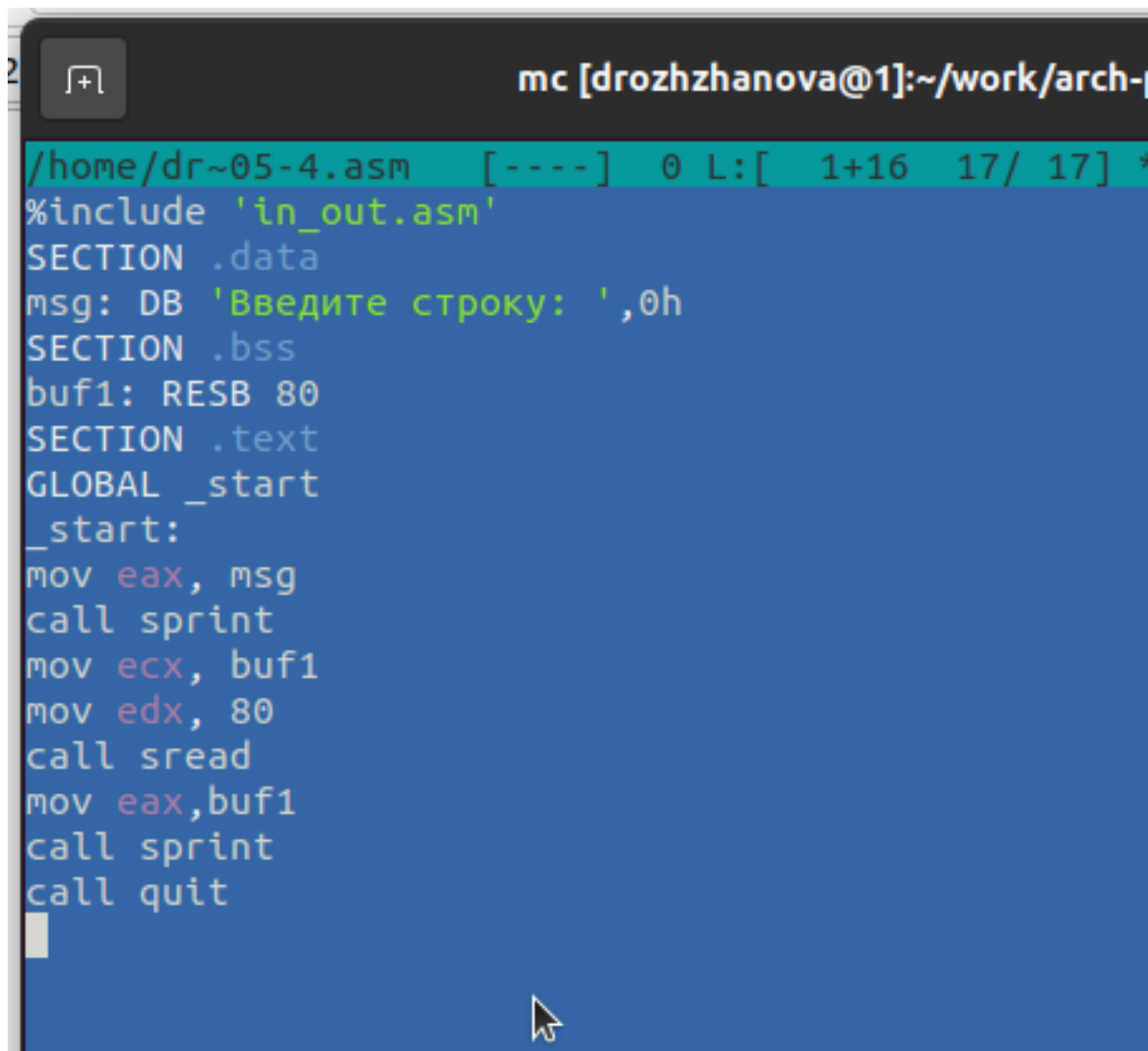
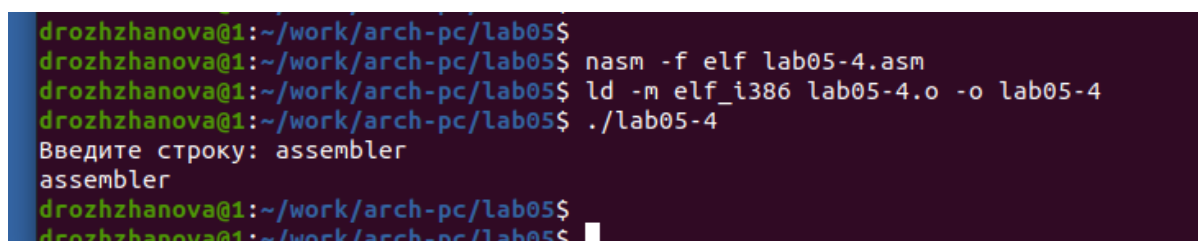


Рис. 4.16: Копирование файла lab05-2.asm



```
mc [drozhzhanova@1]:~/work/arch-  
/home/dr~05-4.asm [----] 0 L:[ 1+16 17/ 17] *  
%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. 4.17: Программа в файле lab05-4.asm



```
drozhzhanova@1:~/work/arch-pc/lab05$  
drozhzhanova@1:~/work/arch-pc/lab05$ nasm -f elf lab05-4.asm  
drozhzhanova@1:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-4.o -o lab05-4  
drozhzhanova@1:~/work/arch-pc/lab05$ ./lab05-4  
Введите строку: assembler  
assembler  
drozhzhanova@1:~/work/arch-pc/lab05$  
drozhzhanova@1:~/work/arch-pc/lab05$
```

Рис. 4.18: Запуск программы lab05-4.asm

5 Выводы

Научились писать базовые ассемблерные программы. Освоили ассемблерные инструкции `mov` и `int`.

6 Источники

1. Архитектура ЭВМ - Материалы курса