

Отчёт по лабораторной работе 5

Архитектура компьютеров

Дмитраков Михаил Алексеевич

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Знакомство с Midnight Commander	6
2.2	Подключение внешнего файла in_out.asm	13
2.3	Задание для самостоятельной работы	17
3	Выводы	21

Список иллюстраций

2.1	Запуск Midnight Commander	7
2.2	Создание каталога	8
2.3	Создание файла lab05-1.asm	9
2.4	Создание файла lab05-1.asm	10
2.5	Программа lab05-1.asm	11
2.6	Просмотр файла lab05-1.asm	12
2.7	Запуск программы lab05-1.asm	13
2.8	Копирование файла in_out.asm	13
2.9	Копирование файла lab05-1.asm	14
2.10	Программа lab05-2.asm	15
2.11	Запуск программы lab05-2.asm	15
2.12	Программа в файле lab05-2.asm	16
2.13	Запуск программы lab05-2.asm	16
2.14	Программа lab05-3.asm	18
2.15	Запуск программы lab05-3.asm	18
2.16	Программа lab05-4.asm	19
2.17	Запуск программы lab05-4.asm	20

Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

2.1 Знакомство с Midnight Commander

В начале работы я запускаю файловый менеджер Midnight Commander (рис. 2.1). С помощью клавиш навигации и Enter перехожу в каталог `~/work/arch-`рс. Далее нажимаю клавишу F7 и создаю новый каталог с именем `lab05` (рис. 2.2).

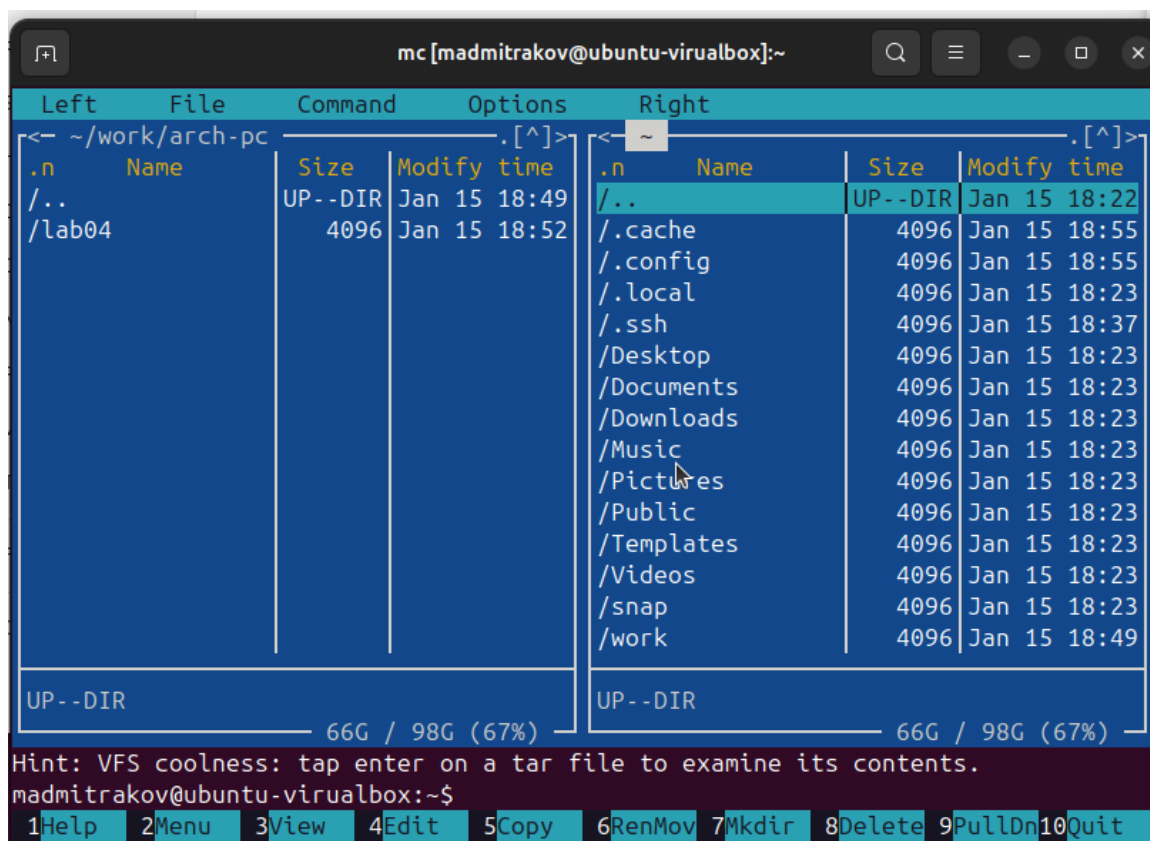


Рисунок 2.1: Запуск Midnight Commander

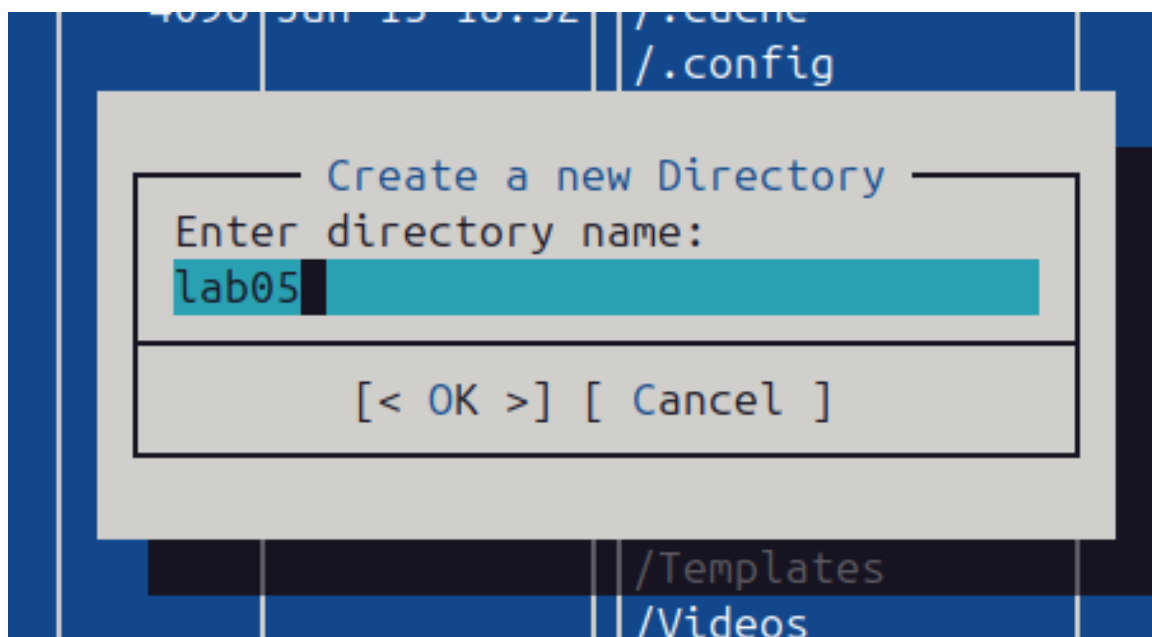


Рисунок 2.2: Создание каталога

После этого при помощи команды `touch` создаю файл `lab05-1.asm` (рис. 2.3, 2.4).

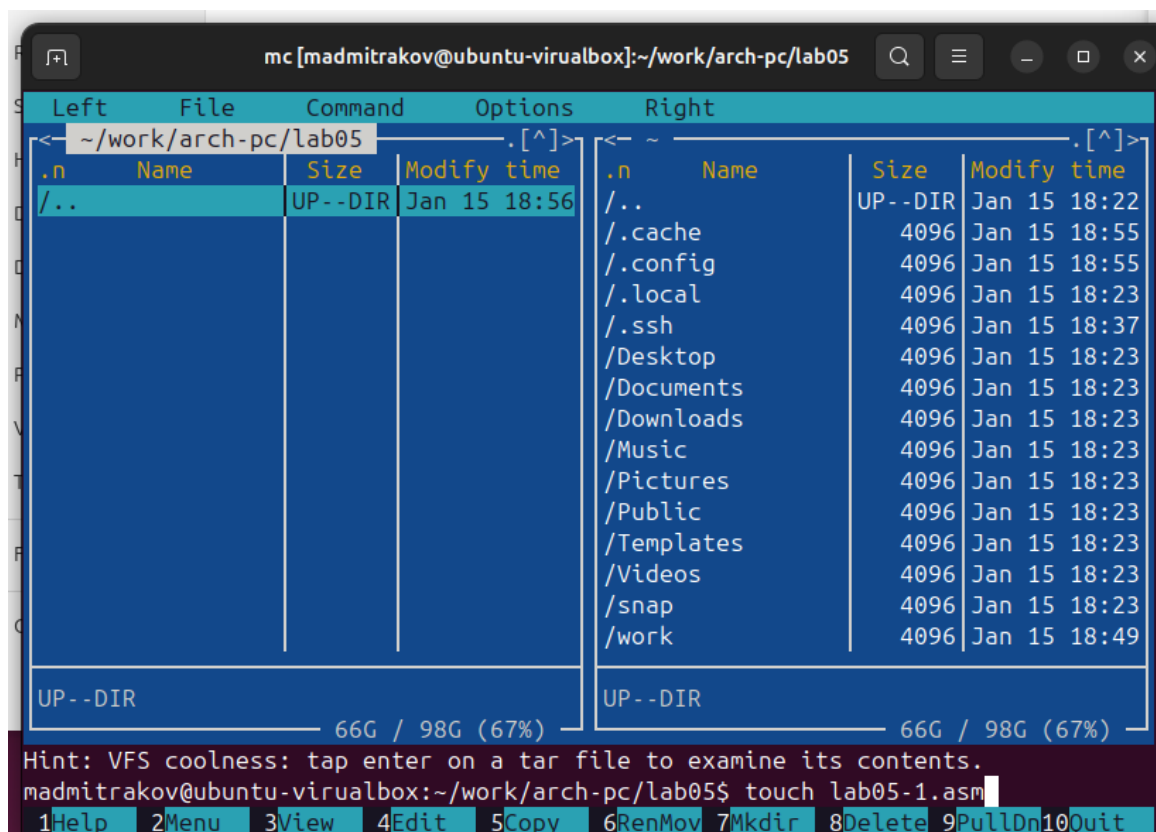


Рисунок 2.3: Создание файла lab05-1.asm

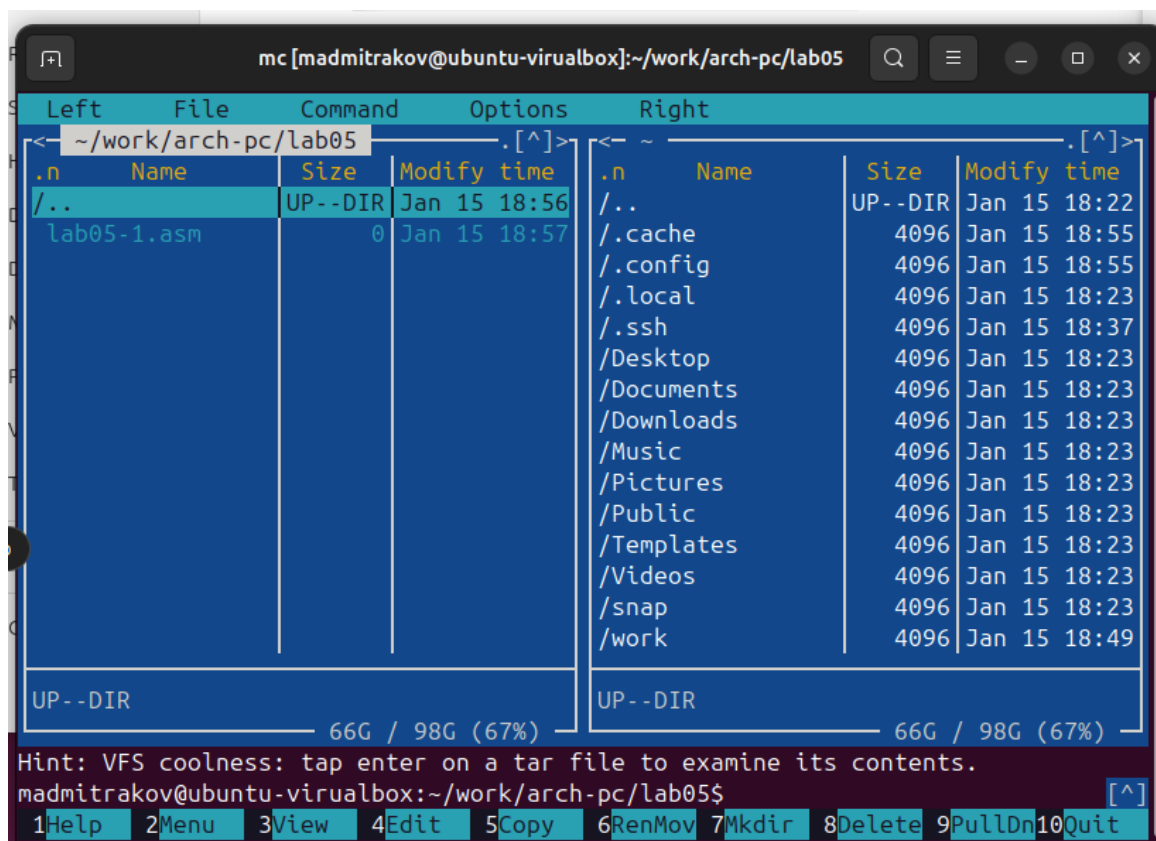
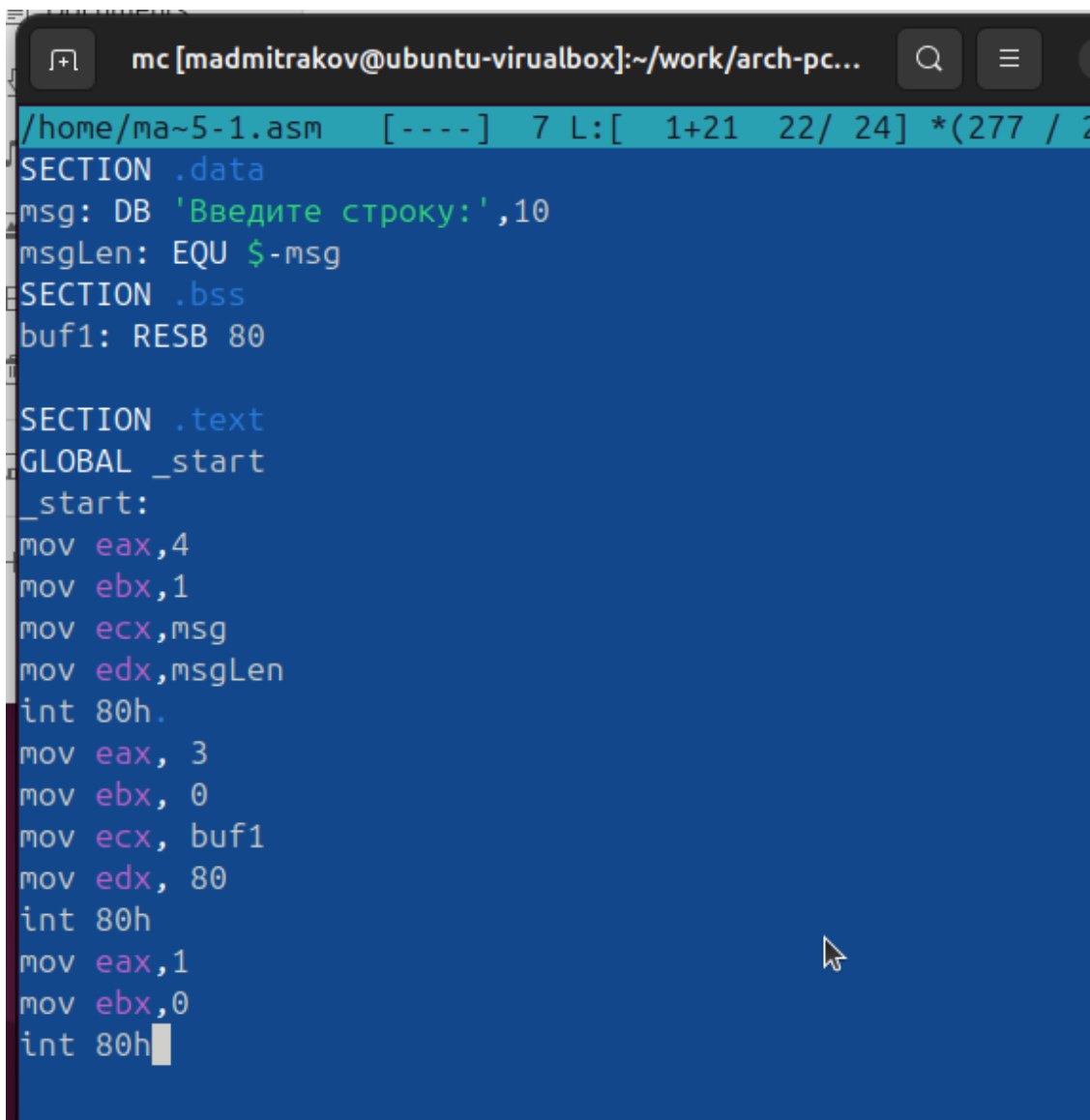


Рисунок 2.4: Создание файла lab05-1.asm

Для редактирования файла нажимаю F4, выбираю встроенный редактор mcedit и ввожу исходный код программы в соответствии с заданием лабораторной работы (рис. 2.5).

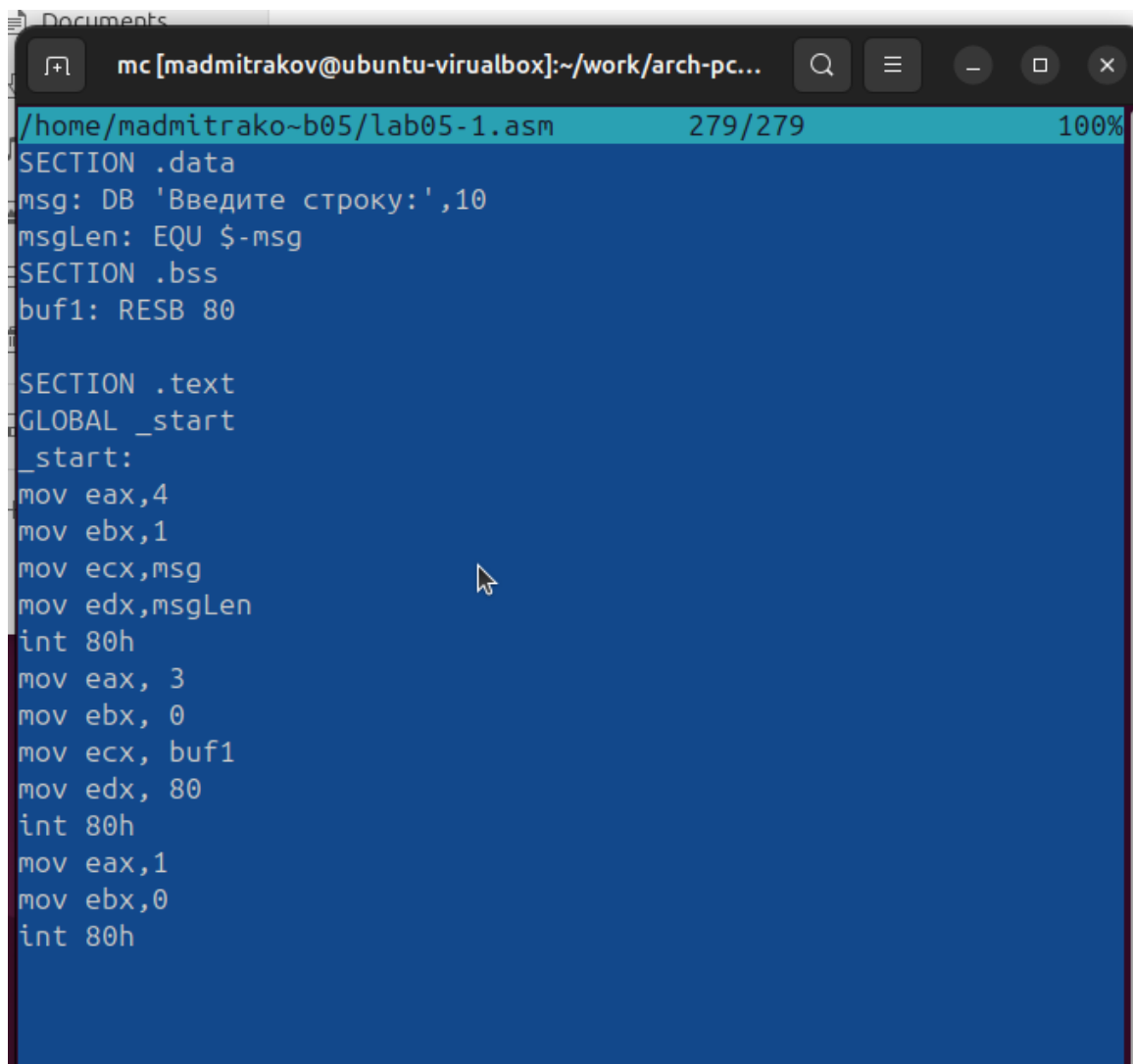


```
mc [madmitrakov@ubuntu-virtualbox]:~/work/arch-pc...
/home/ma~5-1.asm [----] 7 L:[ 1+21 22/ 24] *(277 / 2
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

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
```

Рисунок 2.5: Программа lab05-1.asm

Затем открываю файл в режиме просмотра с помощью клавиши F3 и проверяю корректность введённого кода (рис. 2.6).

A screenshot of a code editor window titled 'mc [madmitrakov@ubuntu-virtualbox]:~/work/arch-pc...'. The editor displays the assembly file 'lab05-1.asm' at 100% zoom. The code is as follows:

```
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
```

Рисунок 2.6: Просмотр файла lab05-1.asm

Далее выполняю трансляцию исходного кода в объектный файл, затем осуществляю компоновку и получаю исполняемый файл программы, после чего проверяю её работу (рис. 2.7).

```

madmitrakov@ubuntu-virtualbox:~/work/arch-pc/lab05$ nasm -f elf lab05-1.asm
madmitrakov@ubuntu-virtualbox:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-1.o -o lab05-1
madmitrakov@ubuntu-virtualbox:~/work/arch-pc/lab05$ ./lab05-1
Введите строку:
hello
madmitrakov@ubuntu-virtualbox:~/work/arch-pc/lab05$

```

Рисунок 2.7: Запуск программы lab05-1.asm

2.2 Подключение внешнего файла in_out.asm

На следующем этапе скачиваю файл `in_out.asm` и размещаю его в рабочем каталоге. Для копирования использую клавишу F5, а для перемещения — F6 (рис. 2.8).

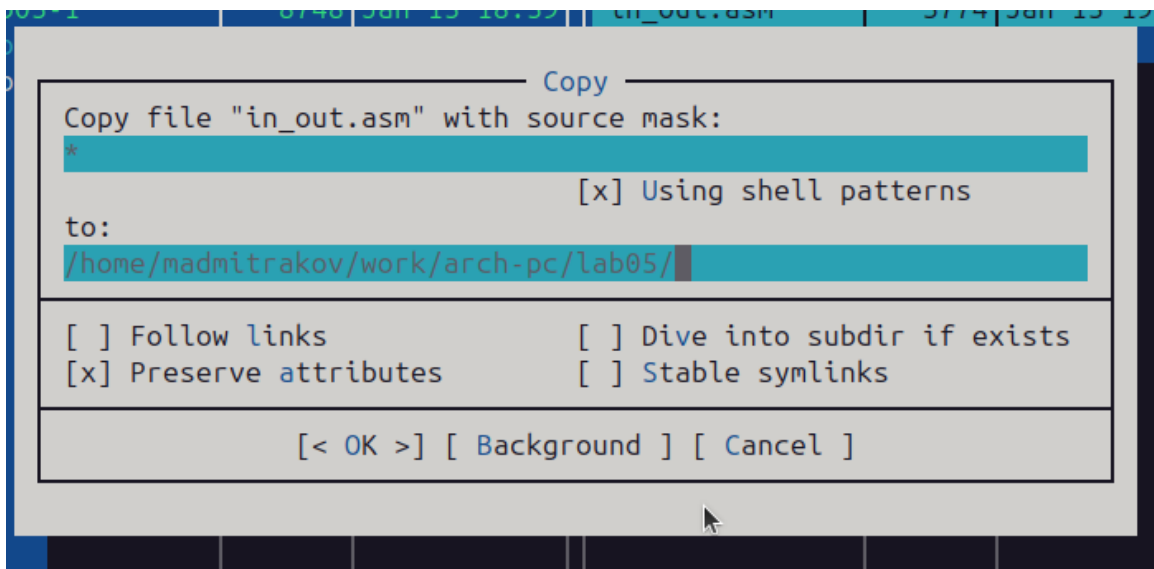


Рисунок 2.8: Копирование файла in_out.asm

После этого копирую файл `lab05-1.asm`, создавая на его основе `lab05-2.asm` (рис. 2.9).

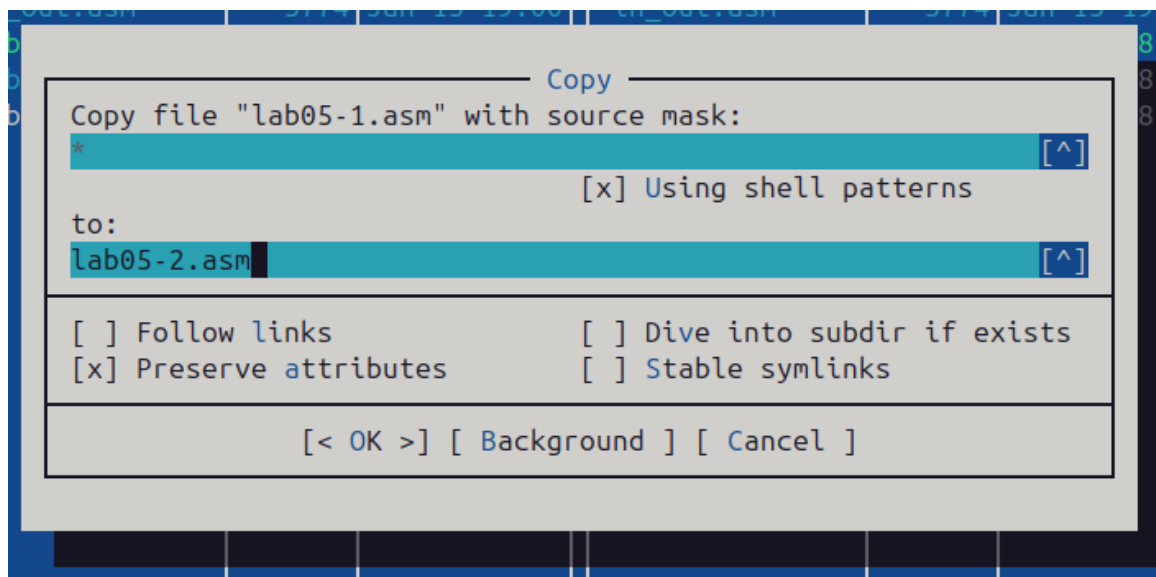
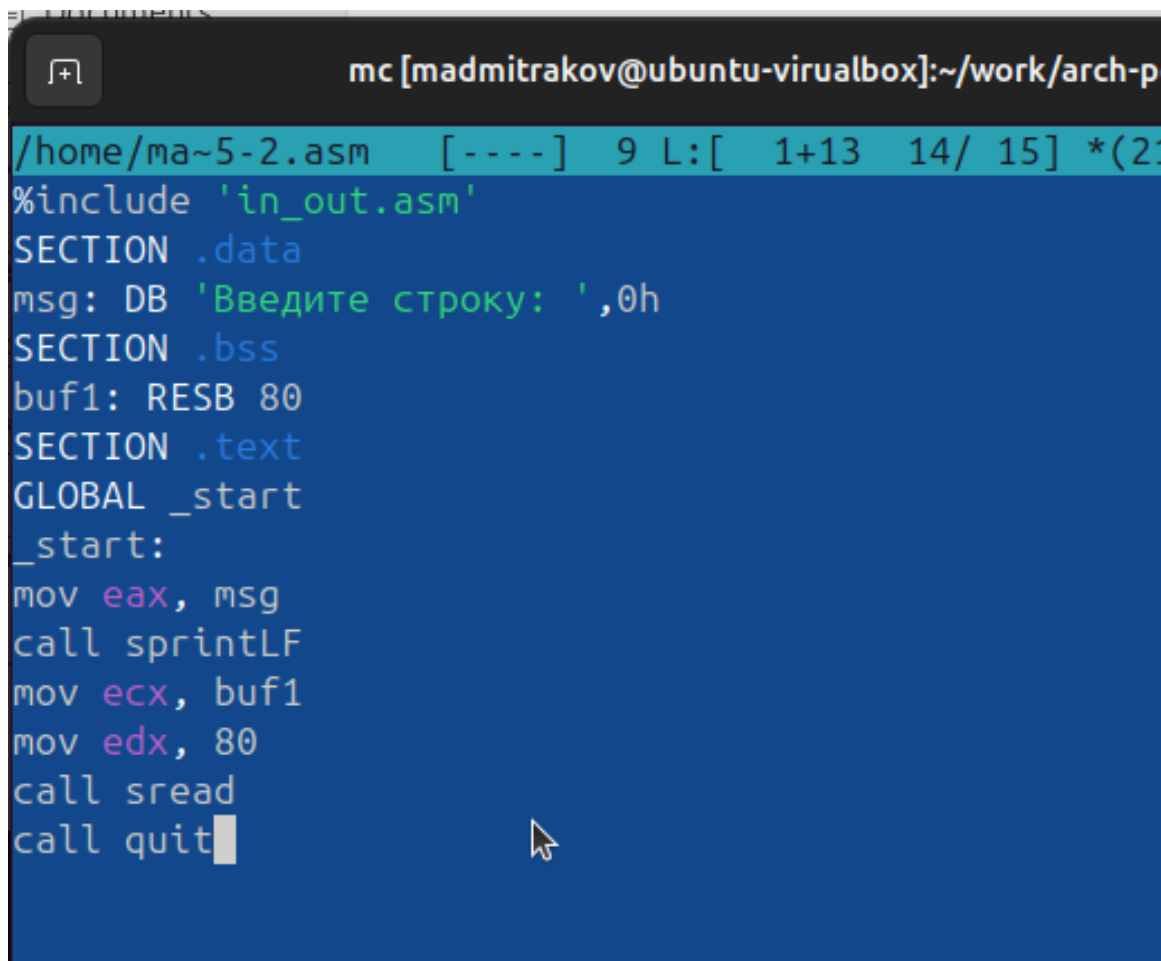


Рисунок 2.9: Копирование файла lab05-1.asm

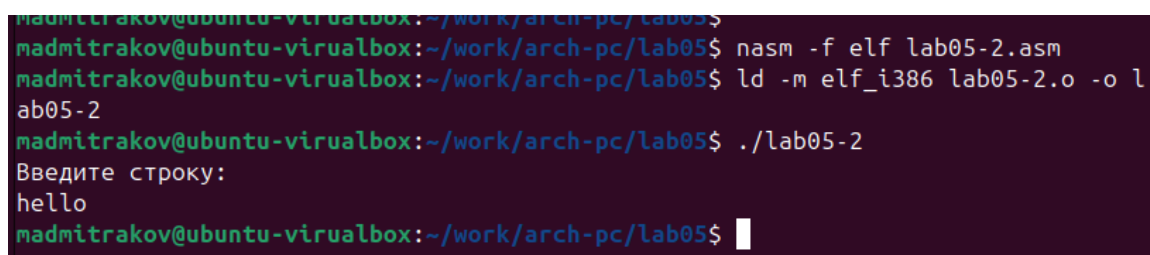
В новом файле lab05-2.asm реализую программу с использованием подпрограмм, подключаемых из внешнего файла in_out.asm (рис. 2.10).



```
mc [madmitrakov@ubuntu-virtualbox]:~/work/arch-pc
/home/ma~5-2.asm [----] 9 L:[ 1+13 14/ 15] *(21
%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
```

Рисунок 2.10: Программа lab05-2.asm

После компиляции и компоновки проверяю запуск полученной программы (рис. 2.11).

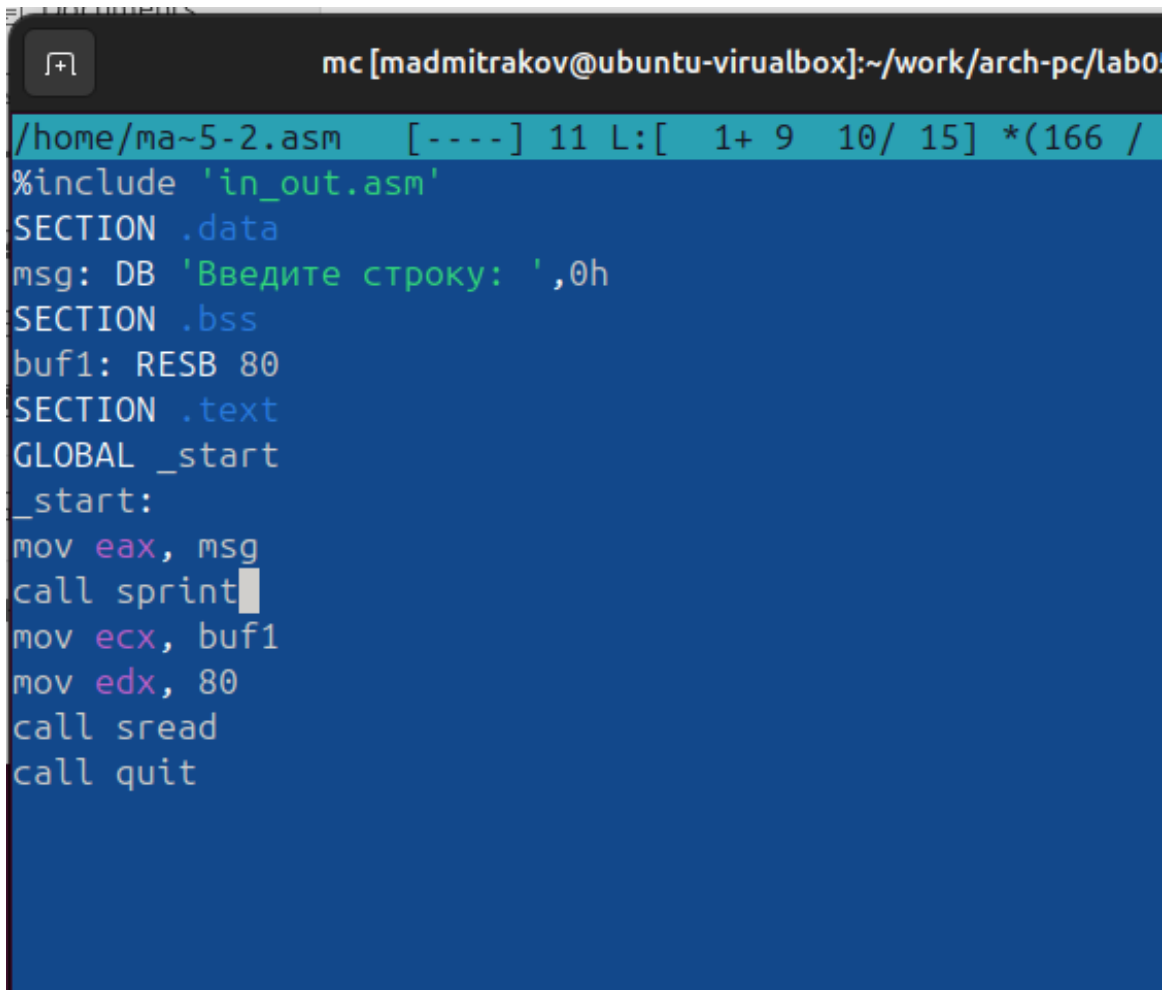


```
madmitrakov@ubuntu-virtualbox:~/work/arch-pc/lab05$
madmitrakov@ubuntu-virtualbox:~/work/arch-pc/lab05$ nasm -f elf lab05-2.asm
madmitrakov@ubuntu-virtualbox:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2
madmitrakov@ubuntu-virtualbox:~/work/arch-pc/lab05$ ./lab05-2
Введите строку:
hello
madmitrakov@ubuntu-virtualbox:~/work/arch-pc/lab05$
```

Рисунок 2.11: Запуск программы lab05-2.asm

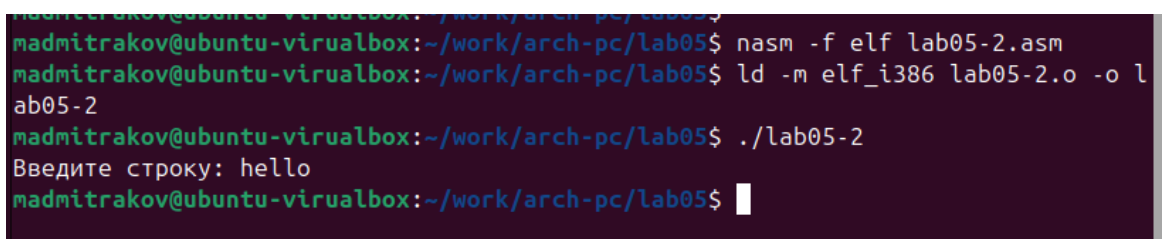
Затем в файле lab05-2.asm заменяю подпрограмму sprintLF на sprint и по-

вторно собираю исполняемый файл. Результаты редактирования и запуска показаны на рисунках (рис. 2.12, 2.13).



```
mc [madmitrakov@ubuntu-virtualbox]:~/work/arch-pc/lab05-2$ cat lab05-2.asm
/home/madmitrakov/work/arch-pc/lab05-2/lab05-2.asm [----] 11 L: [ 1+ 9 10/ 15] *(166 /
%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
```

Рисунок 2.12: Программа в файле lab05-2.asm



```
madmitrakov@ubuntu-virtualbox:~/work/arch-pc/lab05-2$ nasm -f elf lab05-2.asm
madmitrakov@ubuntu-virtualbox:~/work/arch-pc/lab05-2$ ld -m elf_i386 lab05-2.o -o lab05-2
madmitrakov@ubuntu-virtualbox:~/work/arch-pc/lab05-2$ ./lab05-2
Введите строку: hello
madmitrakov@ubuntu-virtualbox:~/work/arch-pc/lab05-2$
```

Рисунок 2.13: Запуск программы lab05-2.asm

В результате выполненных изменений вывод строки перестал завершаться

символом перевода строки.

2.3 Задание для самостоятельной работы

В рамках самостоятельного задания я скопировал файл `lab05-1.asm` и модифицировал программу так, чтобы она выполняла следующий алгоритм: (рис. 2.14, 2.15)

- вывод приглашения вида «Введите строку:»;
- ввод строки с клавиатуры;
- вывод введённой строки на экран.

```
mc [madmitrakov@ubuntu-virtualbox]:~/work/arch-pc/lab05
/home/ma~5-3.asm [----] 7 L:[ 1+26 27/ 28] *(330 / 331b
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

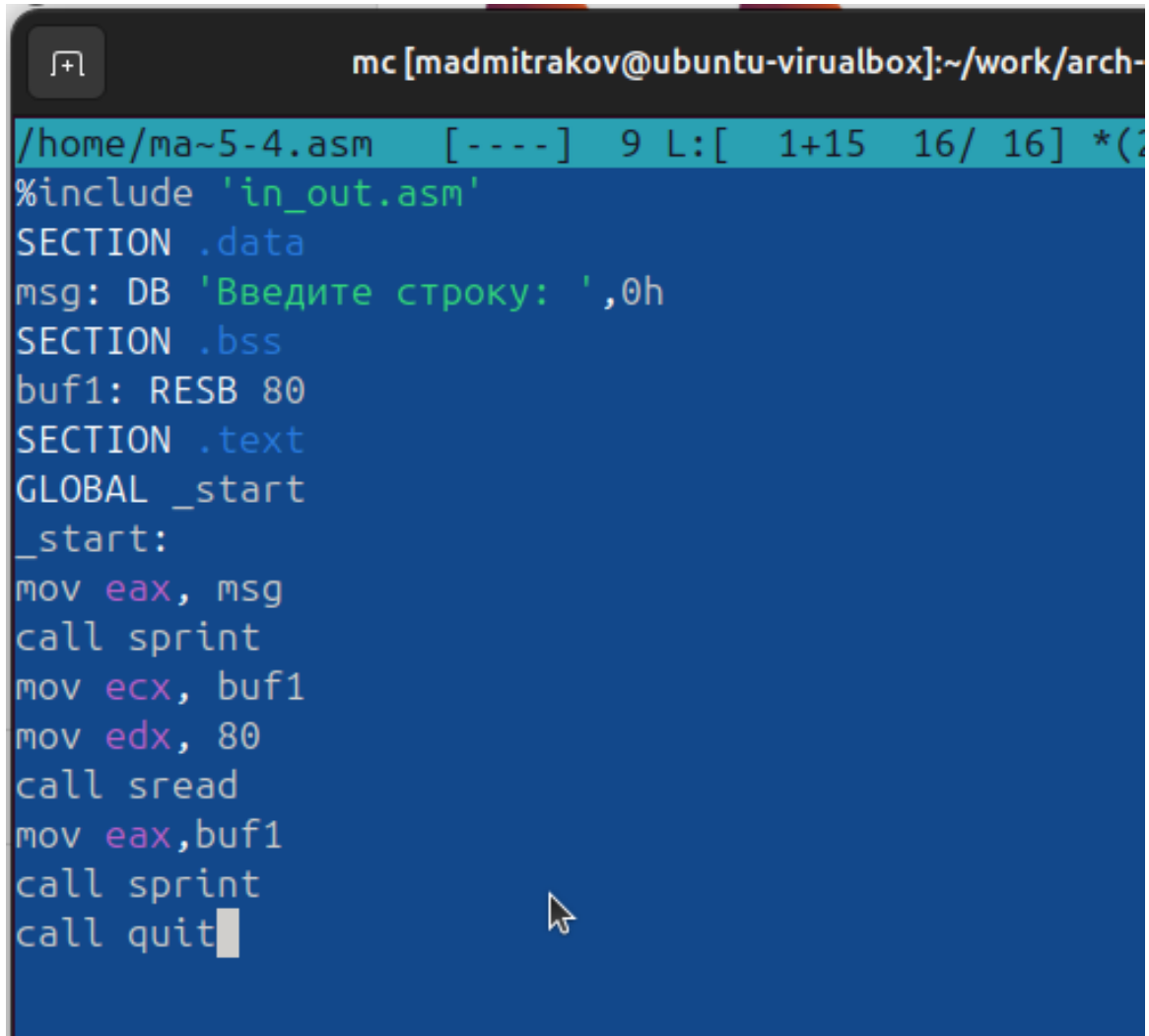
SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h.
mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h.
mov eax,4
mov ebx,1
mov ecx,buf1
mov edx,80
int 80h
mov eax,1
mov ebx,0
int 80h
```

Рисунок 2.14: Программа lab05-3.asm

```
madmitrakov@ubuntu-virtualbox:~/work/arch-pc/lab05$ nasm -f elf lab05-3.asm
madmitrakov@ubuntu-virtualbox:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-3.o -o lab05-3
madmitrakov@ubuntu-virtualbox:~/work/arch-pc/lab05$ ./lab05-3
Введите строку:
hello
hello
madmitrakov@ubuntu-virtualbox:~/work/arch-pc/lab05$
```

Рисунок 2.15: Запуск программы lab05-3.asm

Аналогичным образом я скопировал программу lab05-2.asm и переработал её код, используя подпрограммы из файла in_out.asm. Результаты работы приведены ниже (рис. 2.16, 2.17).



```
mc [madmitrakov@ubuntu-virtualbox]:~/work/arch-  
/home/ma~5-4.asm [----] 9 L:[ 1+15 16/ 16] *(?  
%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
```

Рисунок 2.16: Программа lab05-4.asm

```
madmitrakov@ubuntu-virtualbox:~/work/arch-pc/lab05$ nasm -f elf lab05-4.asm
madmitrakov@ubuntu-virtualbox:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-4.o -o lab05-4
madmitrakov@ubuntu-virtualbox:~/work/arch-pc/lab05$ ./lab05-4
Введите строку: hello
hello
madmitrakov@ubuntu-virtualbox:~/work/arch-pc/lab05$
```

Рисунок 2.17: Запуск программы lab05-4.asm

3 Выводы

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