

Отчёт по лабораторной работе 5

Архитектура компьютеров

Алхоев Абдулмалик-Салим Гапурович

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	11
2.3 Задание для самостоятельной работы	15
3 Выводы	19

Список иллюстраций

2.1 Запуск Midnight Commander	6
2.2 Создание каталога	7
2.3 Создание файла lab05-1.asm	7
2.4 Создание файла lab05-1.asm	8
2.5 Программа lab05-1.asm	9
2.6 Просмотр файла lab05-1.asm	10
2.7 Запуск программы lab05-1.asm	11
2.8 Копирование файла in_out.asm	11
2.9 Копирование файла lab05-1.asm	12
2.10 Программа lab05-2.asm	13
2.11 Запуск программы lab05-2.asm	13
2.12 Программа в файле lab05-2.asm	14
2.13 Запуск программы lab05-2.asm	15
2.14 Программа lab05-3.asm	16
2.15 Запуск программы lab05-3.asm	16
2.16 Программа lab05-4.asm	17
2.17 Запуск программы lab05-4.asm	18

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

Я открываю Midnight Commander (рис. 2.1) и с помощью клавиш со стрелками и Enter перехожу в каталог `~/work/arch-pc`. Затем нажимаю F7, чтобы создать новый каталог `lab05` (рис. 2.2).

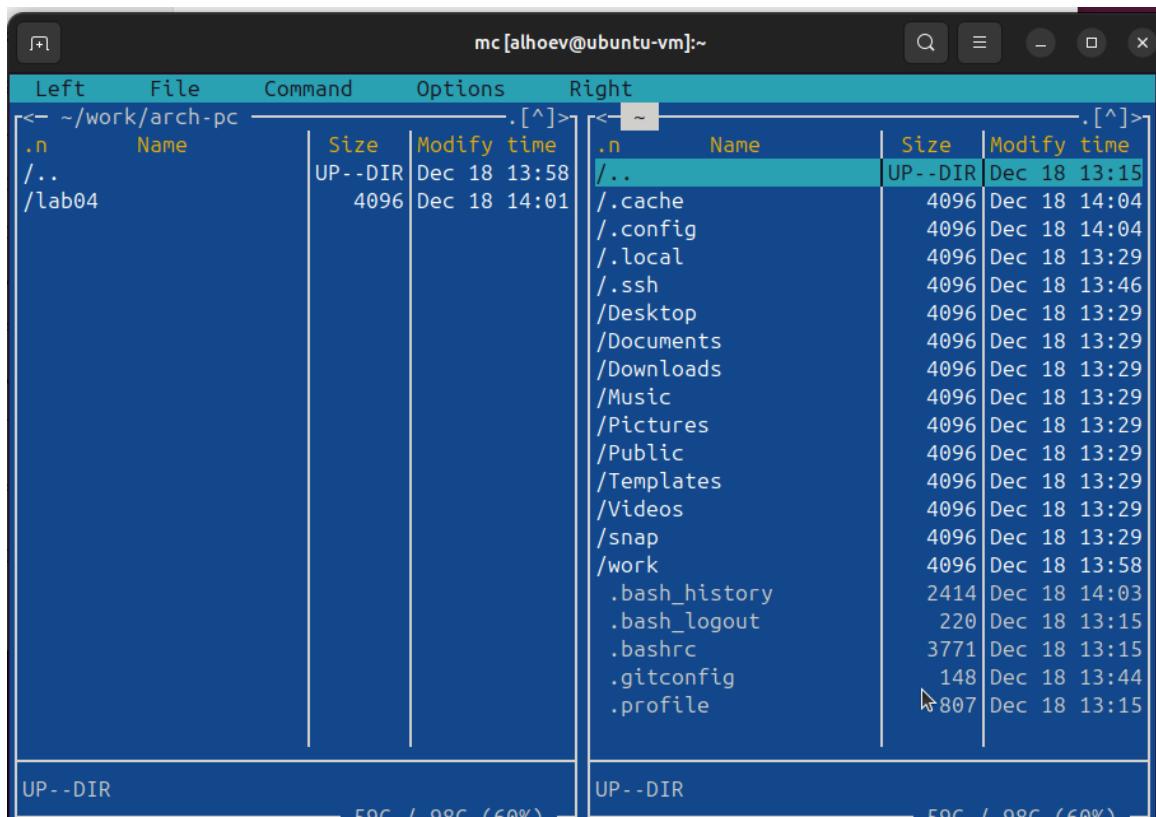


Рисунок 2.1: Запуск Midnight Commander

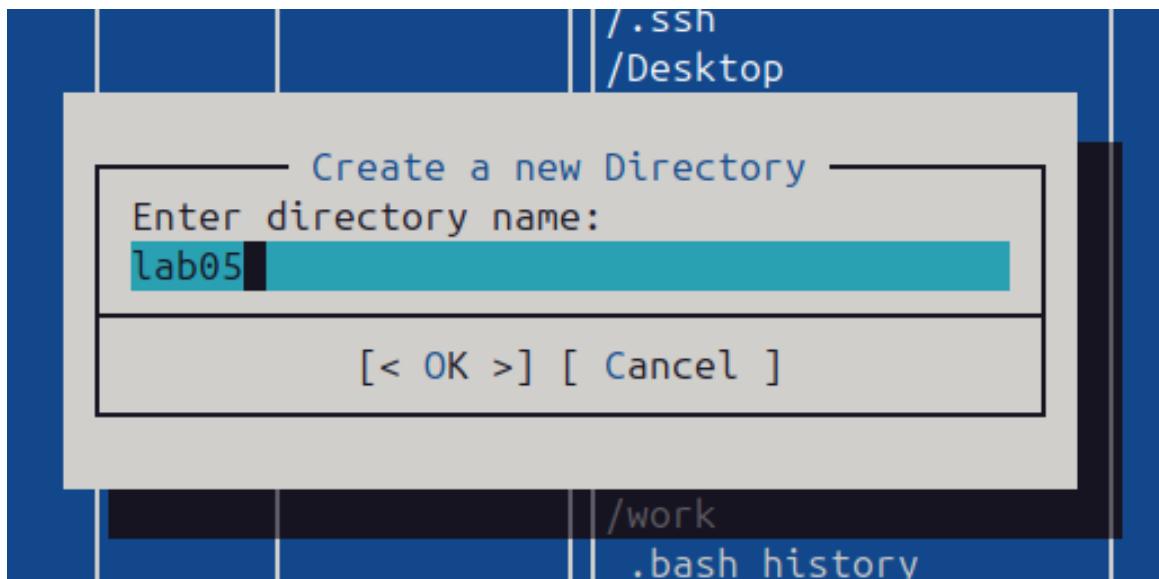


Рисунок 2.2: Создание каталога

Используя команду touch, создаю файл lab05-1.asm (рис. 2.3).

```
mc [alhoev@ubuntu-vm]:~/work/arch-pc/lab05
```

Left	File	Command	Options	Right			
<- ~	~/work/arch-pc/lab05		.[^\>]	<- ~ .[^\>]			
.n	Name	Size	Modify time	.n	Name	Size	Modify time
/..		UP--DIR	Dec 18 14:04	/..		UP--DIR	Dec 18 13:15
lab05-1.asm		0	Dec 18 14:06	/.cache		4096	Dec 18 14:04
				/.config		4096	Dec 18 14:04
				/.local		4096	Dec 18 13:29
				/.ssh		4096	Dec 18 13:46
				/Desktop		4096	Dec 18 13:29
				/Documents		4096	Dec 18 13:29
				/Downloads		4096	Dec 18 13:29
				/Music		4096	Dec 18 13:29
				/Pictures		4096	Dec 18 13:29
				/Public		4096	Dec 18 13:29
				/Templates		4096	Dec 18 13:29
				/Videos		4096	Dec 18 13:29
				/snap		4096	Dec 18 13:29
				/work		4096	Dec 18 13:58
				.bash_history		2414	Dec 18 14:03
				.bash_logout		220	Dec 18 13:15
				.bashrc		3771	Dec 18 13:15
				gitignore		140	Dec 18 12:44

Рисунок 2.3: Создание файла lab05-1.asm

The screenshot shows a terminal window with the following text:

```
alhoev@ubuntu-vm:~/work/arch-pc$ mc
Select an editor. To change later, run 'select-editor'.
1. /bin/nano      <---- easiest
2. /usr/bin/mcedit
3. /usr/bin/vim.tiny
4. /bin/ed

Choose 1-4 [1]: 2
```

Рисунок 2.4: Создание файла lab05-1.asm

Открываю файл на редактирование, нажав клавишу F4. Выбираю редактор mcedit и пишу код программы согласно заданию (рис. 2.5).

The screenshot shows a terminal window with a dark blue background and white text. The title bar reads "alhoev@ubuntu-vm: ~/work". The command entered is "/home/alhoev/~/5/lab05-1.asm". The assembly 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 2.5: Программа lab05-1.asm

Для проверки кода открываю файл на просмотр, нажав клавишу F3, и убеждаюсь, что он содержит необходимый текст (рис. 2.6).

The screenshot shows a terminal window titled 'mc [alhoev@ubuntu-vm]:~/work'. The file path is '/home/alhoev/work/arch-pc/lab05/lab05-1.asm'. The assembly 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

Затем я компилирую файл программы в объектный файл, выполняю компоновку объектного файла и получаю исполняемый файл программы (рис. 2.7).

```
alhoev@ubuntu-vm:~/work/arch-pc/lab05$ nasm -f elf lab05-1.asm
alhoev@ubuntu-vm:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-1.o -o lab05-1
alhoev@ubuntu-vm:~/work/arch-pc/lab05$ ./lab05-1
Введите строку:
test
alhoev@ubuntu-vm:~/work/arch-pc/lab05$
```

Рисунок 2.7: Запуск программы lab05-1.asm

2.2 Подключение внешнего файла in_out.asm

Скачиваю файл in_out.asm и помещаю его в рабочий каталог (рис. 2.8). Для копирования файла использую клавишу F5, а для перемещения — клавишу F6.

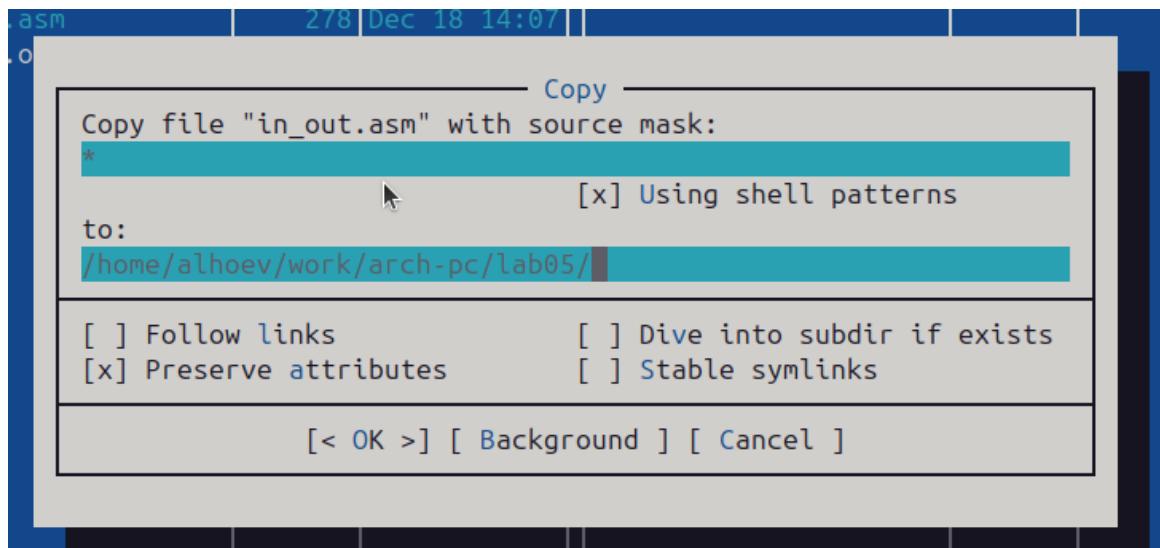


Рисунок 2.8: Копирование файла in_out.asm

Затем я копирую lab05-1.asm в lab05-2.asm (рис. 2.9).

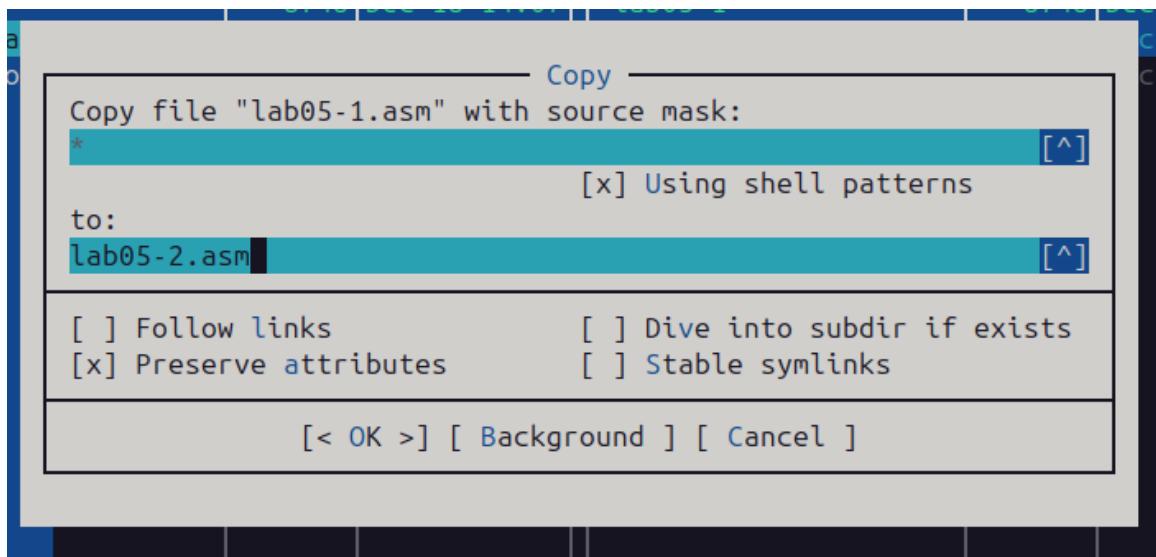
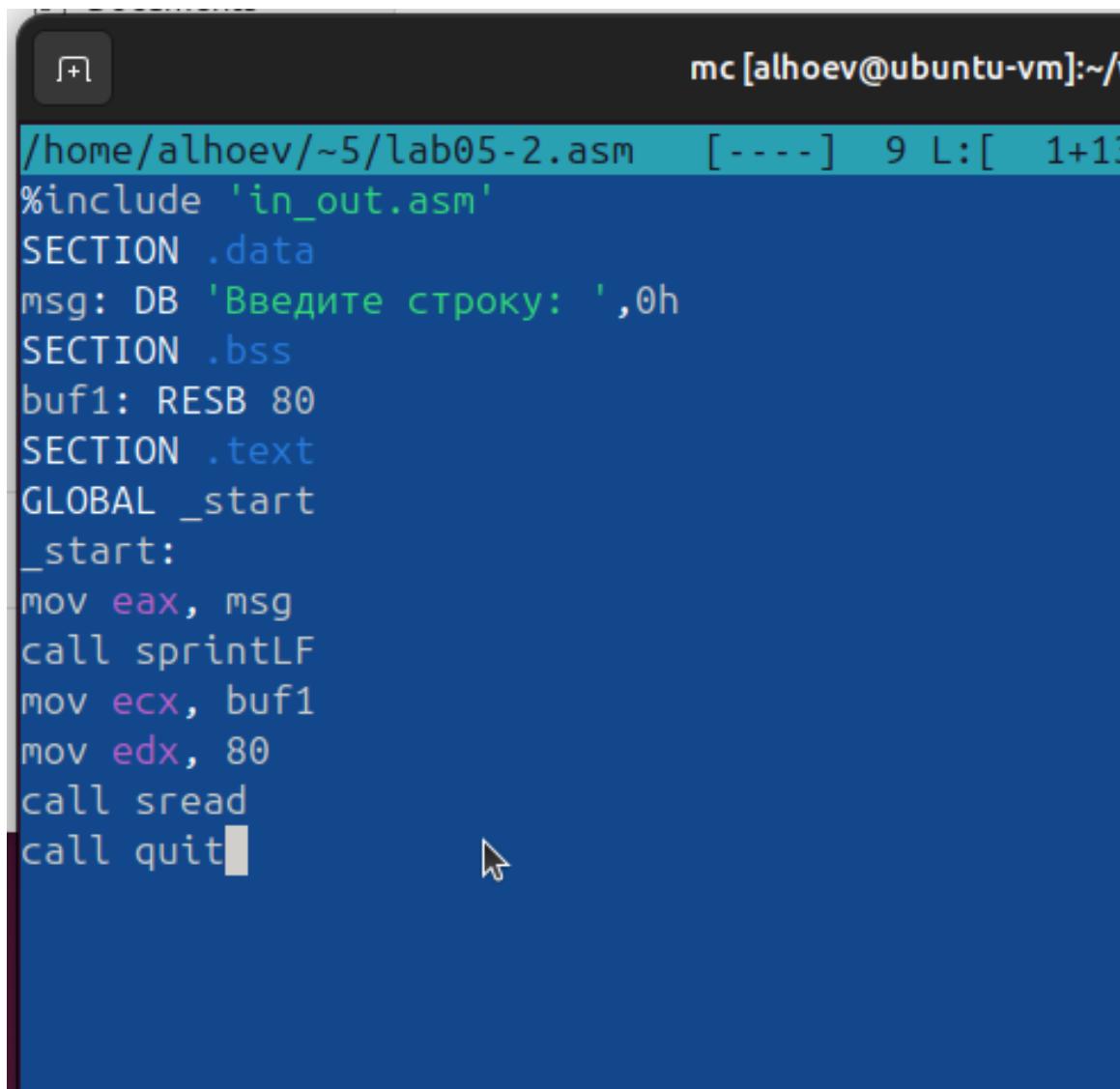


Рисунок 2.9: Копирование файла lab05-1.asm

В новом файле lab05-2.asm пишу код программы с использованием подпрограмм из внешнего файла in_out.asm (рис. 2.10).



The screenshot shows a terminal window with the following content:

```
mc [alhoev@ubuntu-vm]:~/
```

```
/home/alhoev/~/5/lab05-2.asm      [ ----- ]  9 L:[ 1+13
```

```
%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
```

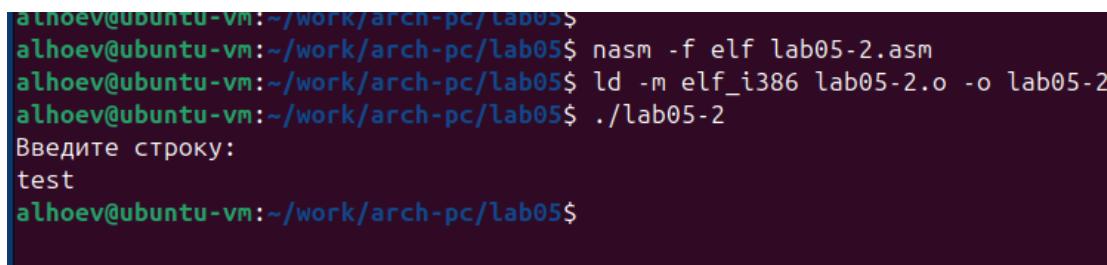
```
    mov edx, 80
```

```
    call sread
```

```
    call quit
```

Рисунок 2.10: Программа lab05-2.asm

После компиляции программы я проверяю её запуск (рис. 2.11).



```
alhoev@ubuntu-vm:~/work/arch-pc/lab05$
```

```
alhoev@ubuntu-vm:~/work/arch-pc/lab05$ nasm -f elf lab05-2.asm
```

```
alhoev@ubuntu-vm:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2
```

```
alhoev@ubuntu-vm:~/work/arch-pc/lab05$ ./lab05-2
```

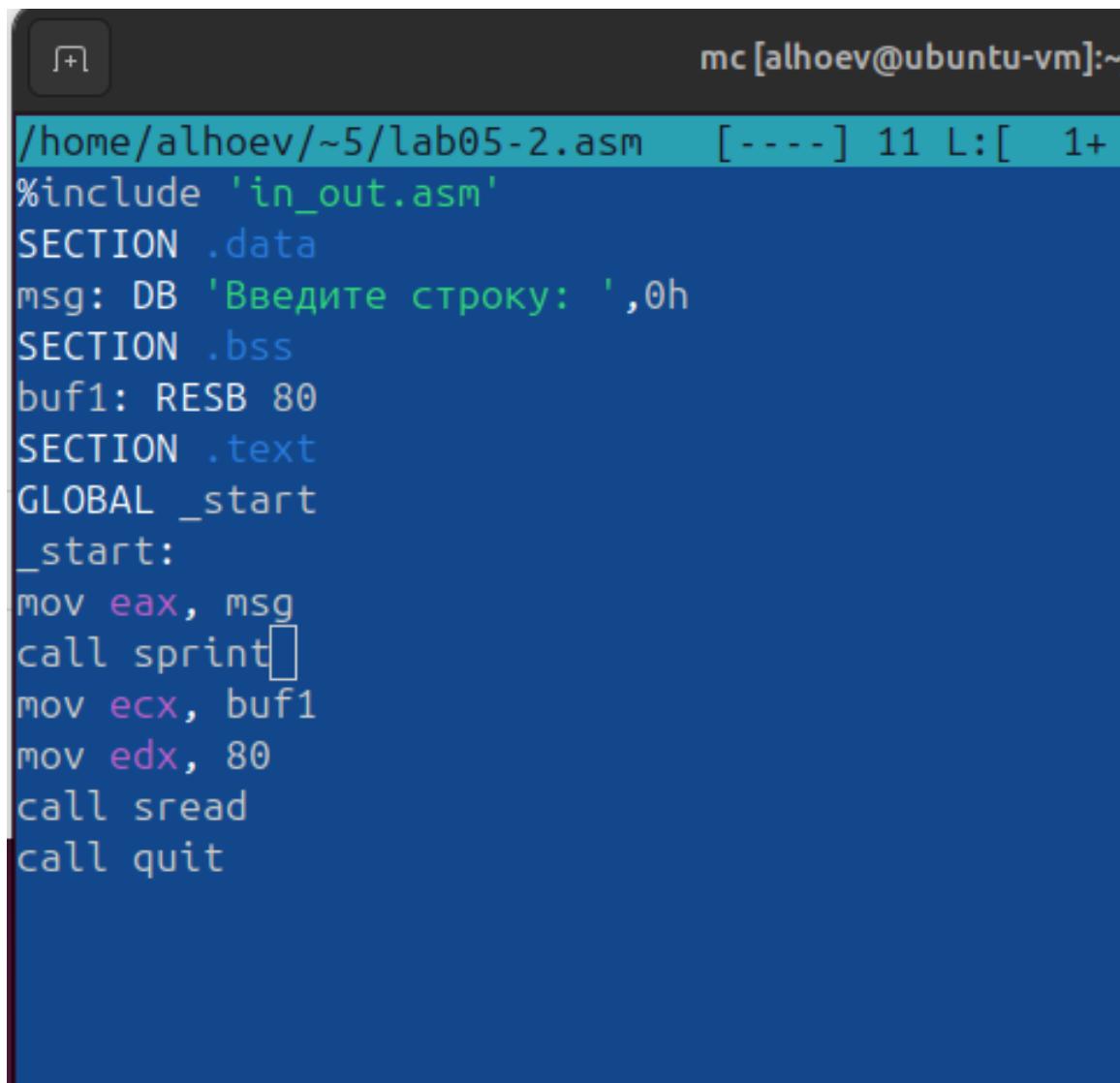
```
Введите строку:
```

```
test
```

```
alhoev@ubuntu-vm:~/work/arch-pc/lab05$
```

Рисунок 2.11: Запуск программы lab05-2.asm

В файле lab05-2.asm я заменяю подпрограмму sprintLF на sprint, после чего заново собираю исполняемый файл (рис. 2.12) (рис. 2.13).



The screenshot shows a terminal window titled 'mc [alhoev@ubuntu-vm]:' containing assembly code. The code includes directives like .data, .bss, and .text, and defines symbols such as msg, buf1, and _start. It also contains instructions for moving data between registers (eax, ecx, edx) and calling functions (sprint, sread, quit). The code is color-coded for syntax highlighting.

```
/home/alhoev/~/5/lab05-2.asm [----] 11 L:[ 1+
%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

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

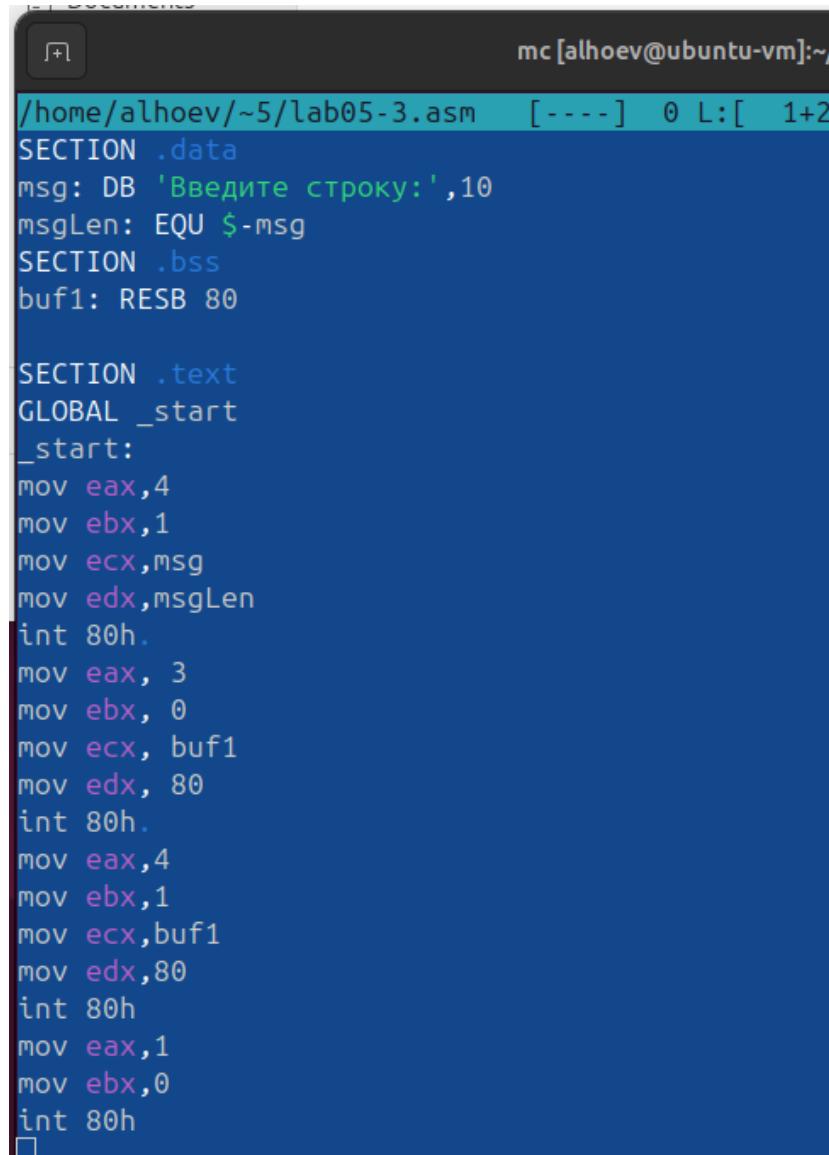
Рисунок 2.13: Запуск программы lab05-2.asm

Теперь после вывода строки программа не завершается символом перехода на новую строку.

2.3 Задание для самостоятельной работы

Я скопировал программу lab05-1.asm и изменил код так, чтобы она работала по следующему алгоритму: (рис. 2.14) (рис. 2.15)

- вывести приглашение типа «Введите строку:»;
- ввести строку с клавиатуры;
- вывести введённую строку на экран.

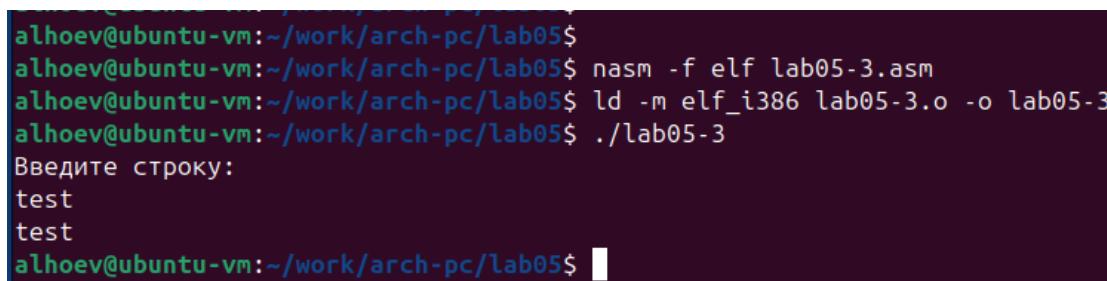


```
mc [alhoev@ubuntu-vm]:~/Documents
```

```
/home/alhoev/~/5/lab05-3.asm  [----]  0 L:[ 1+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,4
    mov ebx,1
    mov ecx,buf1
    mov edx,80
    int 80h
    mov eax,1
    mov ebx,0
    int 80h
□
```

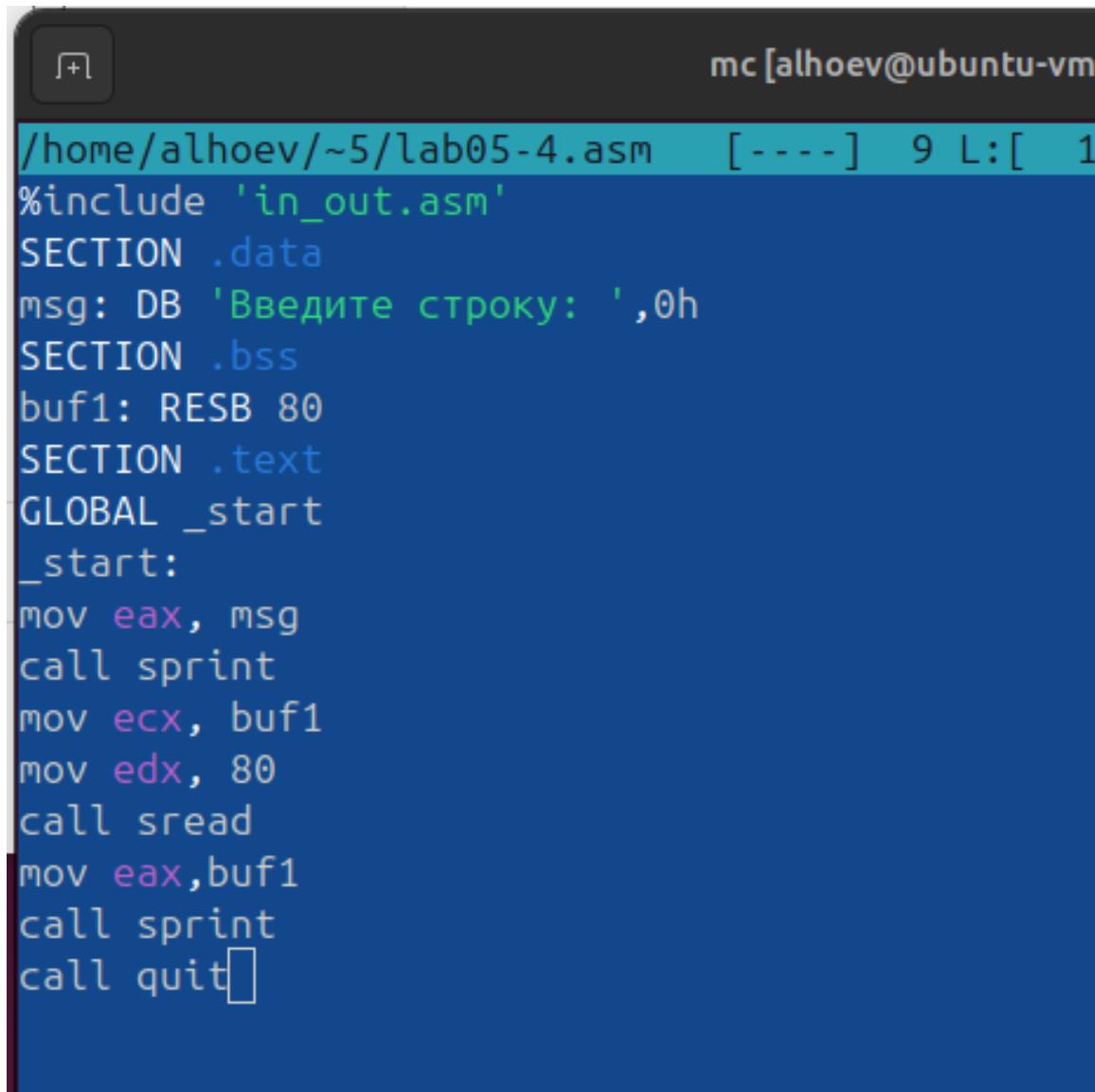
Рисунок 2.14: Программа lab05-3.asm



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

Рисунок 2.15: Запуск программы lab05-3.asm

Аналогичным образом я скопировал программу lab05-2.asm и изменил код, но теперь использовал подпрограммы из файла in_out.asm (рис. 2.16) (рис. 2.17).



The screenshot shows a terminal window titled 'mc [alhoev@ubuntu-vm]' containing assembly code. The code includes an include directive for 'in_out.asm', defines sections for data and bss, and contains assembly instructions for reading and printing strings. The assembly code is as follows:

```
/home/alhoev/~/5/lab05-4.asm      [----]  9 L:[ 1
%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

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