

# Отчёт по лабораторной работе №5

Архитектура компьютера

Диденко Дмитрий Владимирович НПИбд-03-23

# 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	18

## Список иллюстраций

2.1	окно Midnight Commander . . . . .	6
2.2	Создание каталога . . . . .	7
2.3	touch lab05-1.asm . . . . .	8
2.4	Код программы lab05-1.asm . . . . .	9
2.5	Проверка кода lab05-1.asm . . . . .	10
2.6	Компиляция и запуск программы lab05-1.asm . . . . .	10
2.7	Копирование файла in_out.asm . . . . .	11
2.8	Копирование файла in_out.asm . . . . .	12
2.9	Копирование файла lab05-1.asm . . . . .	13
2.10	Код программы lab05-2.asm . . . . .	14
2.11	Компиляция и запуск программы lab05-2.asm . . . . .	14
2.12	Код программы lab05-2.asm . . . . .	15
2.13	Компиляция и запуск программы lab05-2.asm . . . . .	15
2.14	Код программы lab05-3.asm . . . . .	16
2.15	Компиляция и запуск программы lab05-3.asm . . . . .	16
2.16	Код программы lab05-4.asm . . . . .	17
2.17	Компиляция и запуск программы lab05-4.asm . . . . .	17

## 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

Открыл Midnight Commander (рис. [2.1]) и с помощью клавиш со стрелками и Enter перешел в каталог ~/work/arch-рс. Затем нажал F7 и создал каталог lab05 (рис. [2.2]).

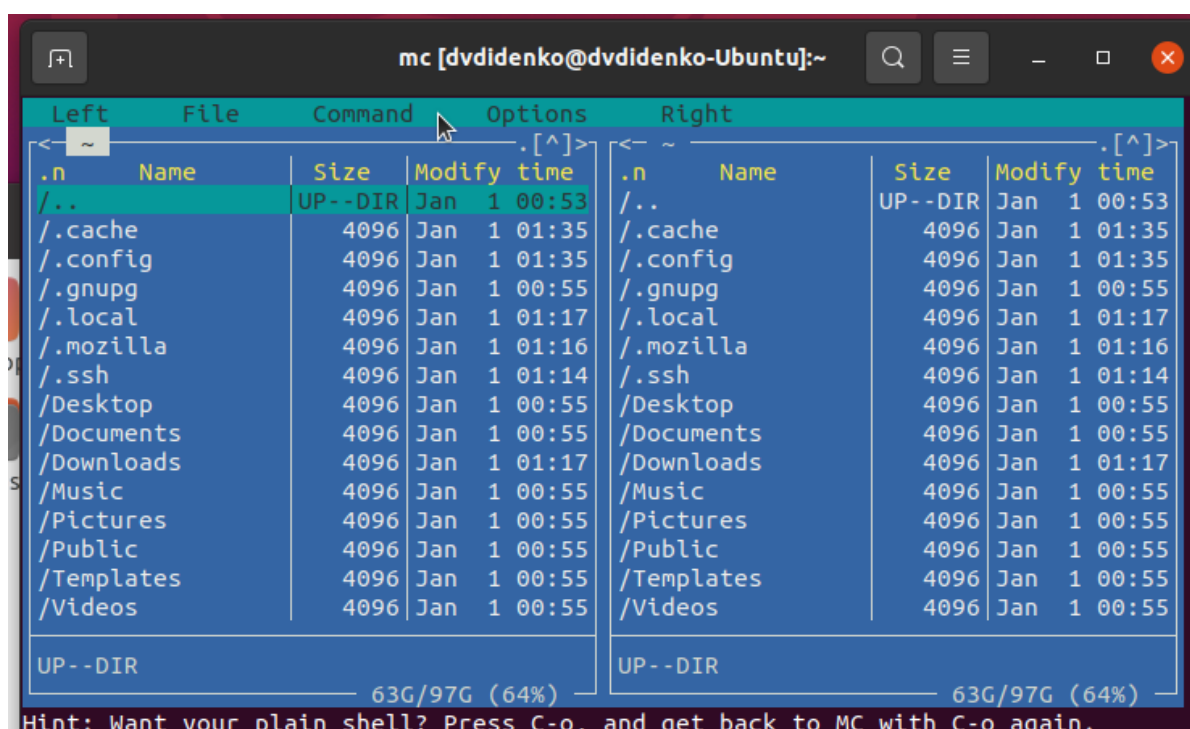


Рис. 2.1: окно Midnight Commander

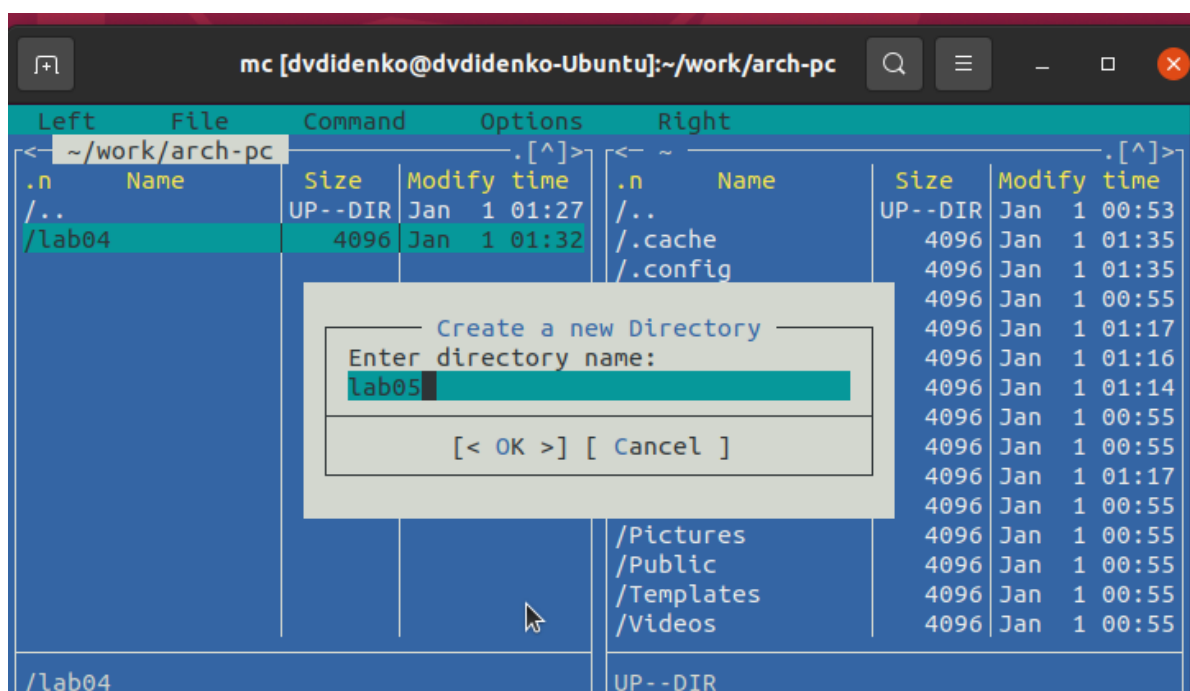


Рис. 2.2: Создание каталога

С использованием команды `touch` создал файл `lab05-1.asm` (рис. [2.3]).

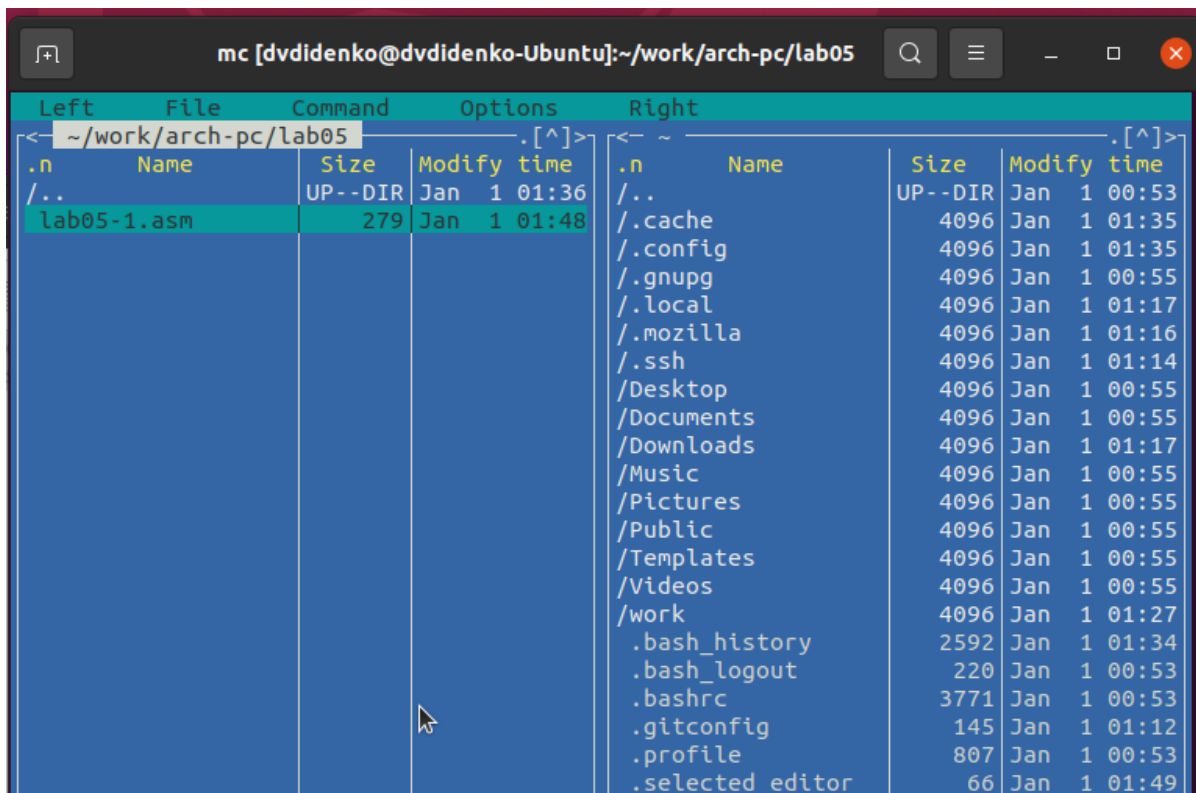
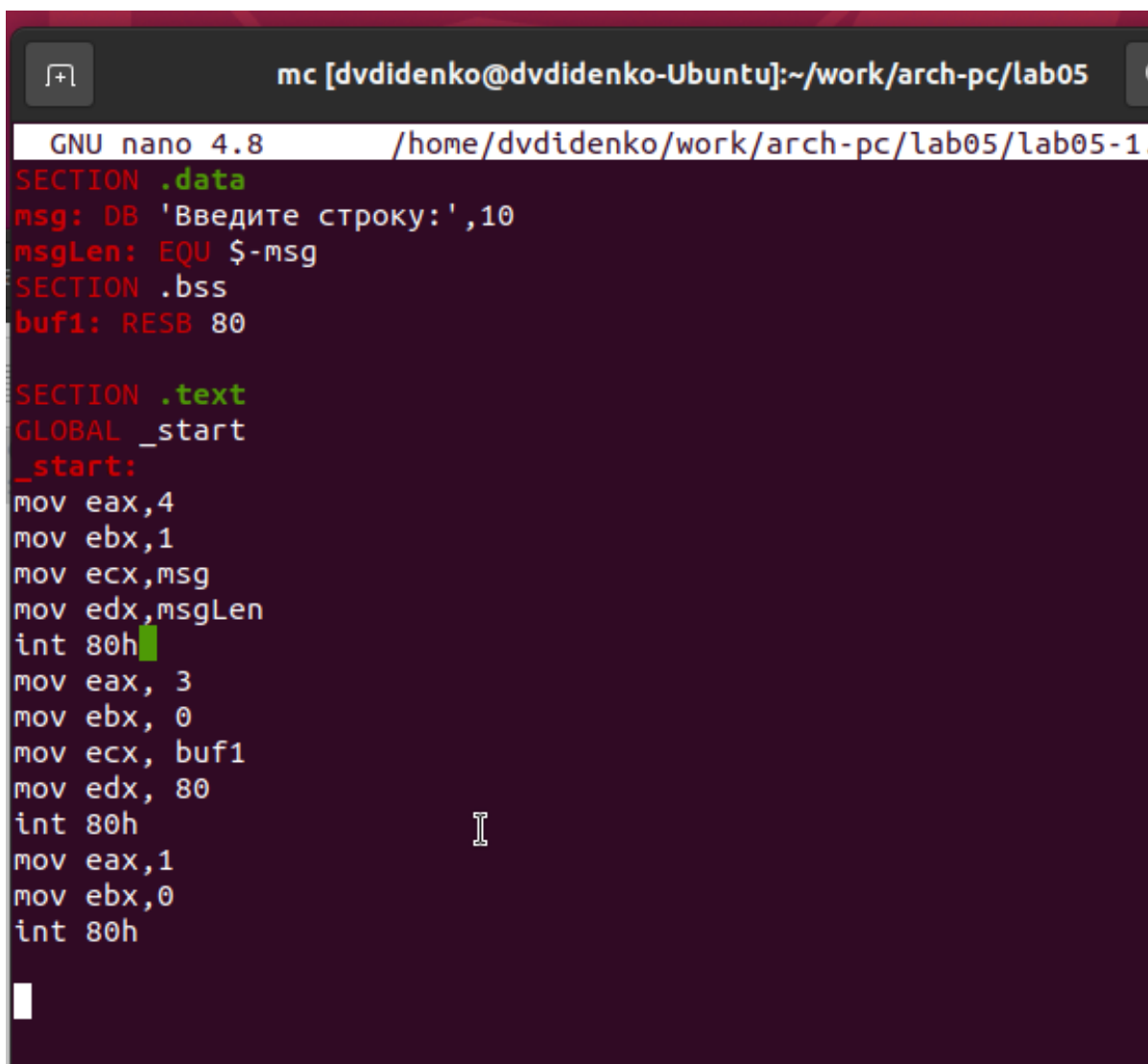


Рис. 2.3: touch lab05-1.asm

Открыл файл для редактирования, нажав клавишу F4, и выбрал редактор mceditor. Затем написал код программы, соответствующий заданию (рис. [2.4]).



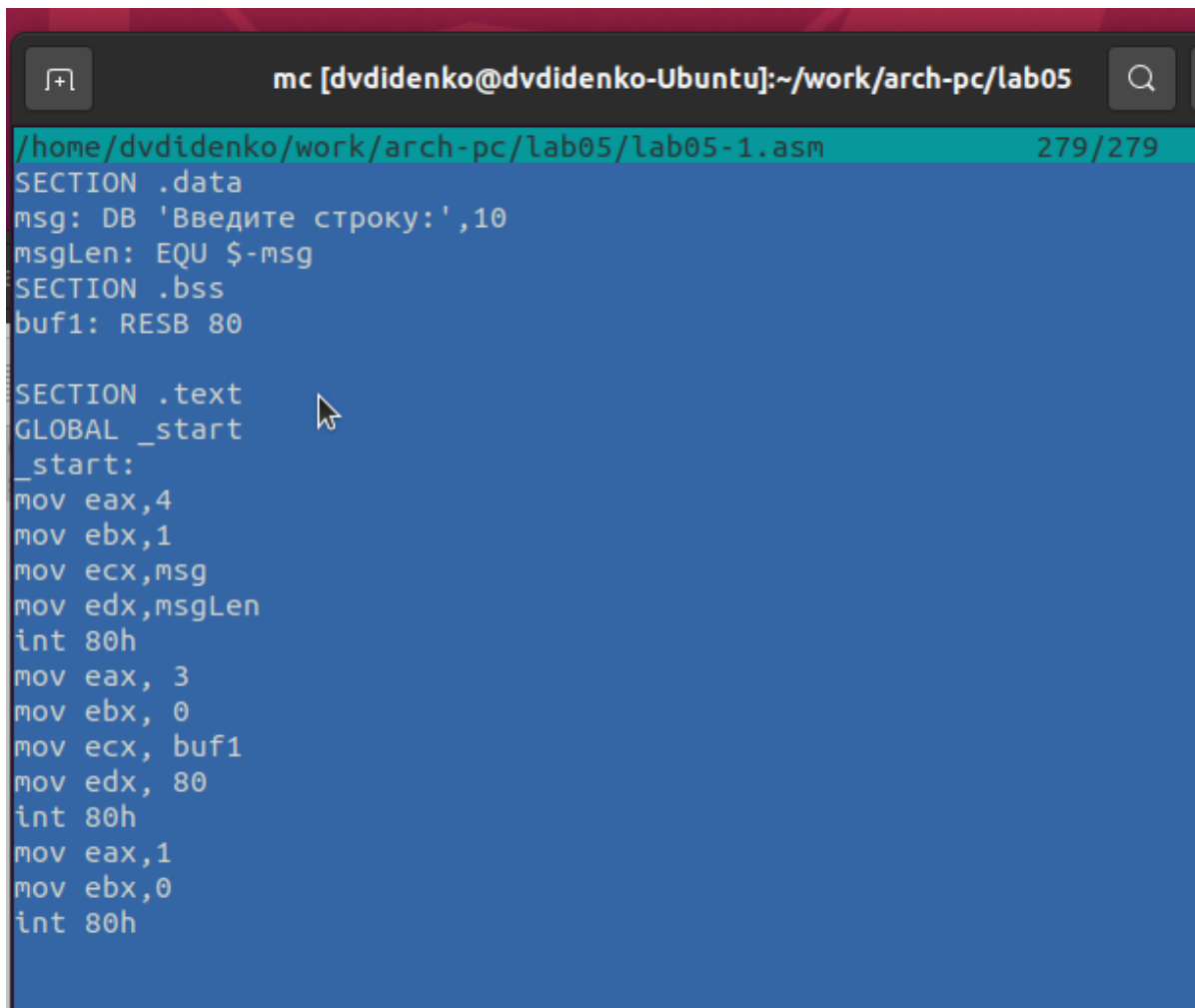


```
mc [dvdidenko@dvdidenko-Ubuntu]:~/work/arch-pc/lab05
GNU nano 4.8 /home/dvdidenko/work/arch-pc/lab05/lab05-1.asm
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

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. 2.4: Код программы lab05-1.asm

Открыл файл для просмотра, нажав клавишу F3, и убедился, что он содержит написанный код (рис. [2.5]).

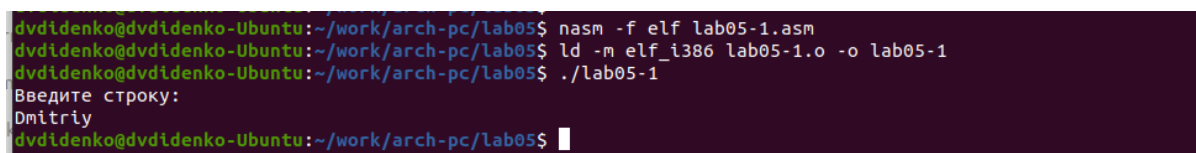


```
mc [dvdidenko@dvdidenko-Ubuntu]:~/work/arch-pc/lab05
/home/dvdidenko/work/arch-pc/lab05/lab05-1.asm 279/279
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

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. 2.5: Проверка кода lab05-1.asm

С помощью трансляции файла программы в объектный файл, выполнения компоновки объектного файла и получения исполняемого файла, проверил работу программы (рис. [2.6]).



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

Рис. 2.6: Компиляция и запуск программы lab05-1.asm

## 2.2 Подключение внешнего файла in\_out.asm

Скачал файл in\_out.asm (рис. [2.7]) и разместил его в рабочем каталоге (рис. [2.8]).

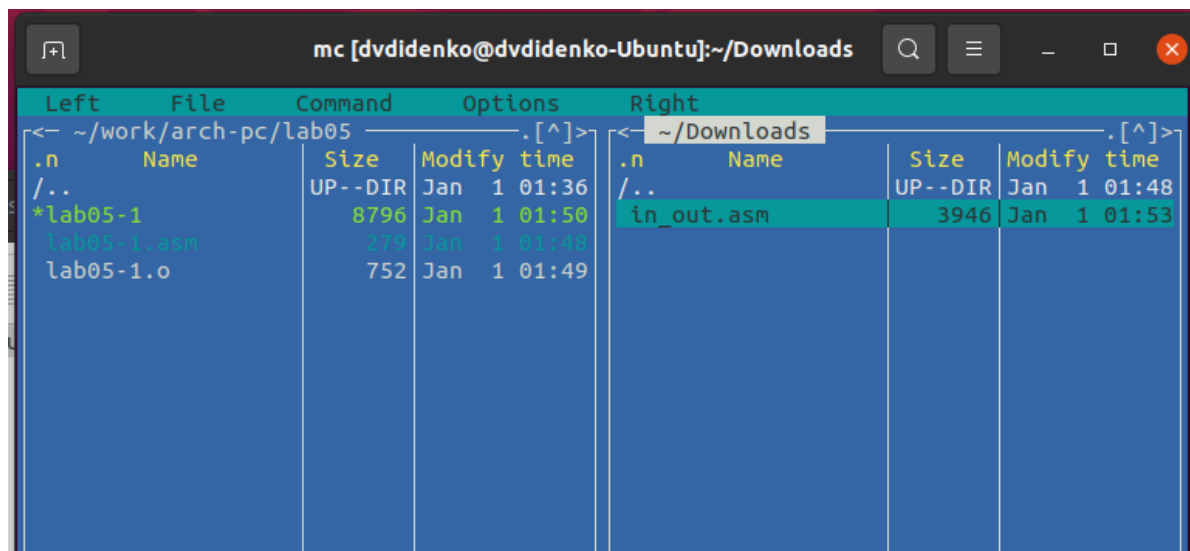


Рис. 2.7: Копирование файла in\_out.asm

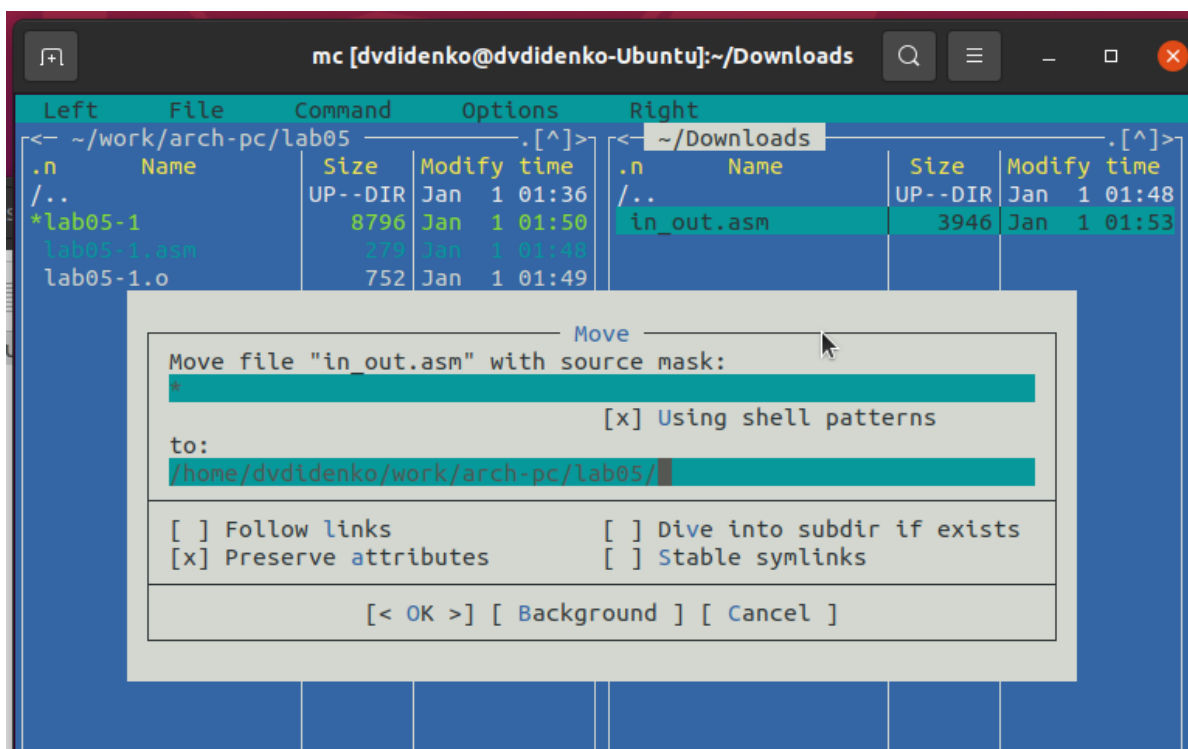


Рис. 2.8: Копирование файла in\_out.asm

С использованием клавиши F5 скопировал содержимое файла lab05-1.asm в файл lab05-2.asm (рис. [2.9]).

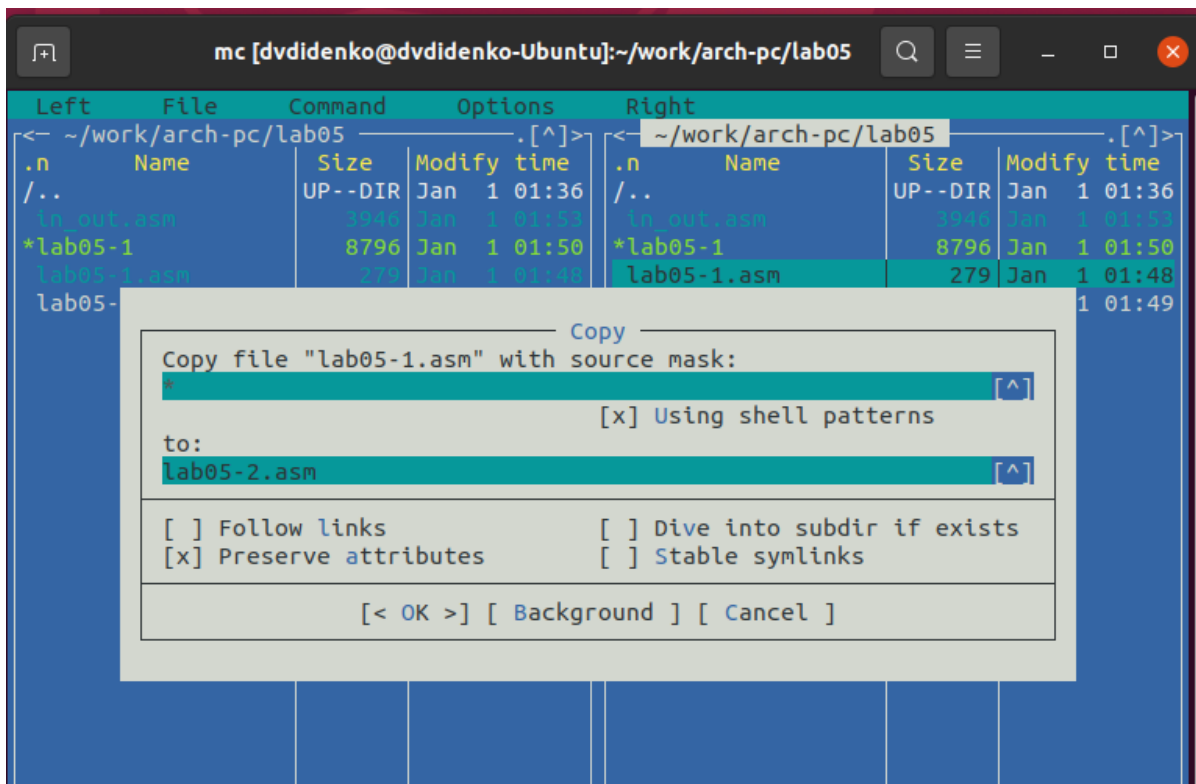
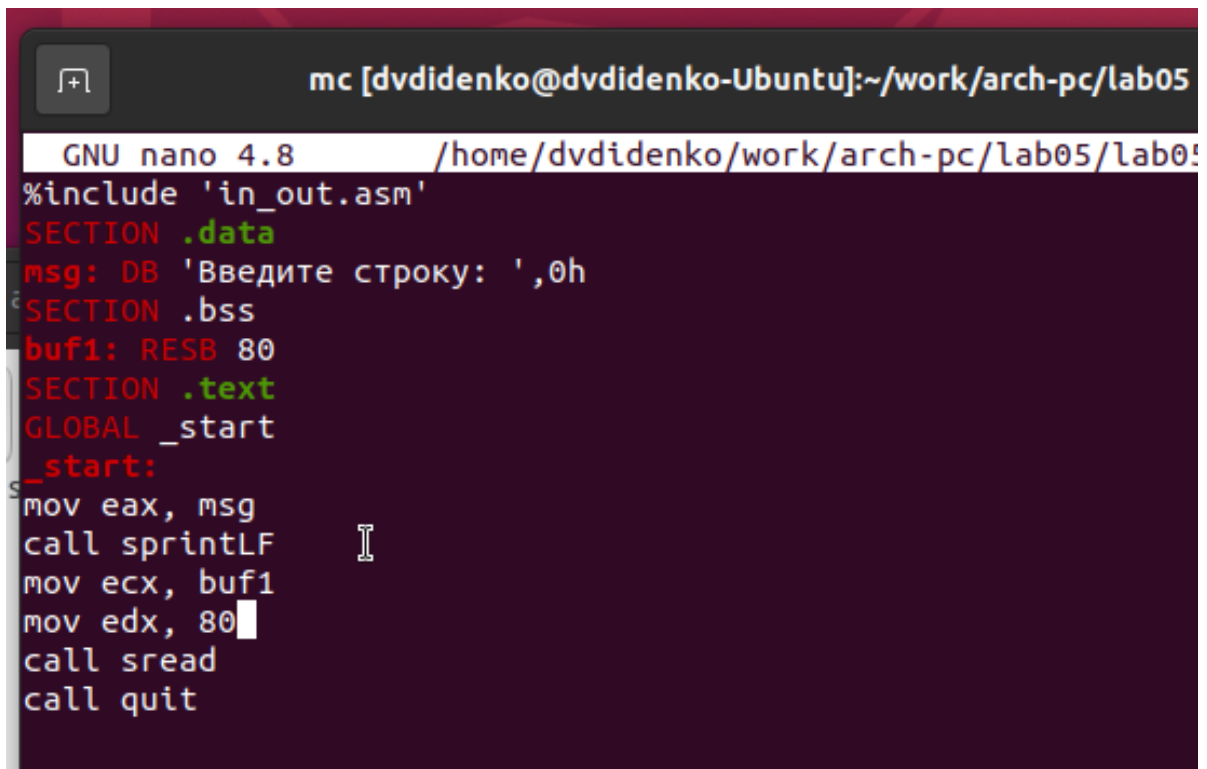


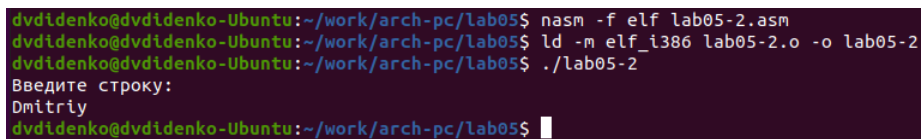
Рис. 2.9: Копирование файла lab05-1.asm

Написал код программы lab05-2.asm, используя подпрограммы из внешнего файла in\_out.asm. (рис. [2.10]) Затем скомпилировал программу и проверил ее запуск. (рис. [2.11])



```
mc [dvdidenko@dvdidenko-Ubuntu]:~/work/arch-pc/lab05
GNU nano 4.8 /home/dvdidenko/work/arch-pc/lab05/lab05
%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
```

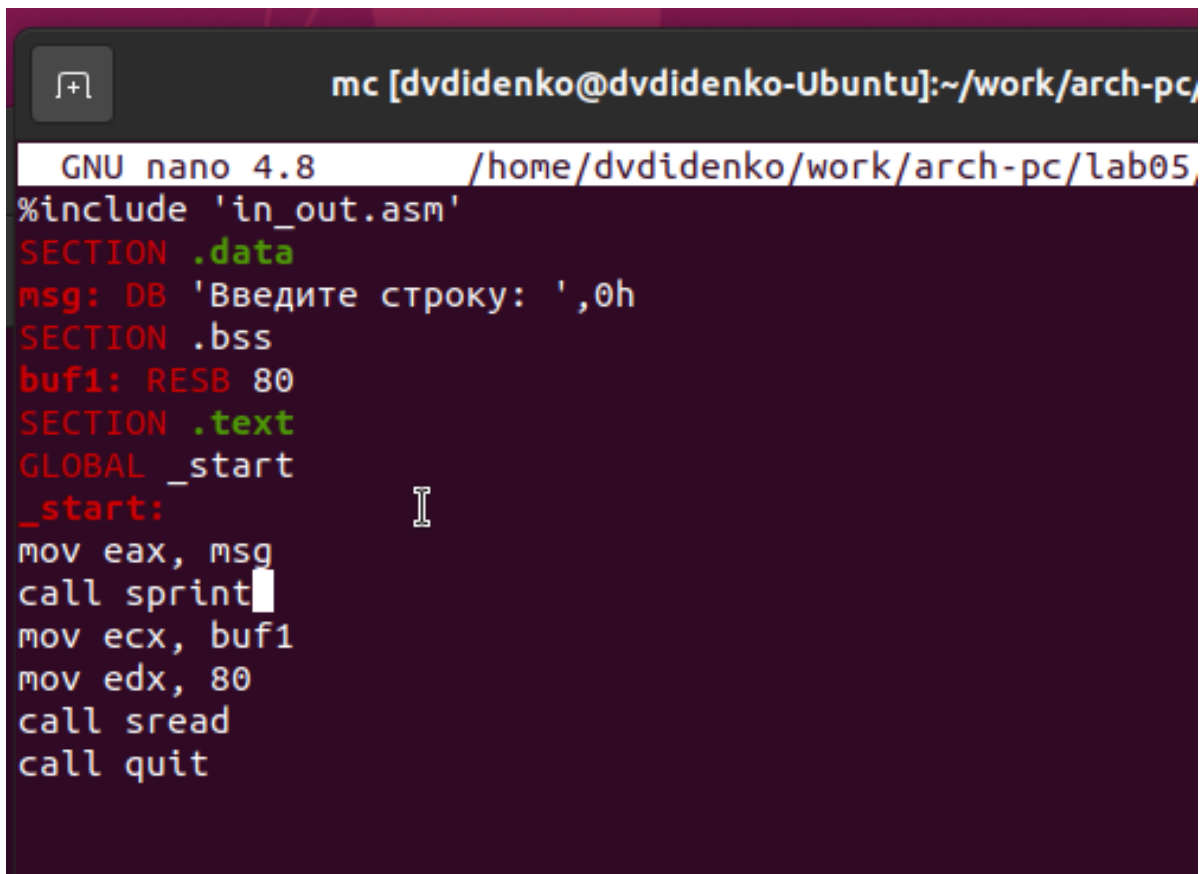
Рис. 2.10: Код программы lab05-2.asm



```
dvdidenko@dvdidenko-Ubuntu:~/work/arch-pc/lab05$ nasm -f elf lab05-2.asm
dvdidenko@dvdidenko-Ubuntu:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2
dvdidenko@dvdidenko-Ubuntu:~/work/arch-pc/lab05$ ./lab05-2
Введите строку:
Dmitriy
dvdidenko@dvdidenko-Ubuntu:~/work/arch-pc/lab05$
```

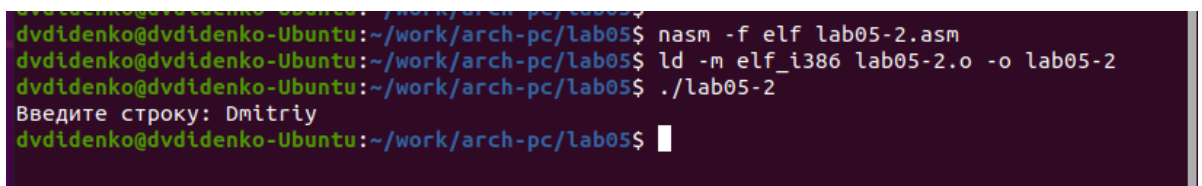
Рис. 2.11: Компиляция и запуск программы lab05-2.asm

В файле lab05-2.asm заменил вызов подпрограммы sprintLF на sprint. (рис. [2.12]) Пересобрал исполняемый файл. (рис. [2.13]) Теперь после вывода строки символ перехода на новую строку отсутствует.



```
mc [dvdidenko@dvdidenko-Ubuntu]:~/work/arch-pc/  
GNU nano 4.8 /home/dvdidenko/work/arch-pc/lab05-2.asm  
%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. 2.12: Код программы lab05-2.asm



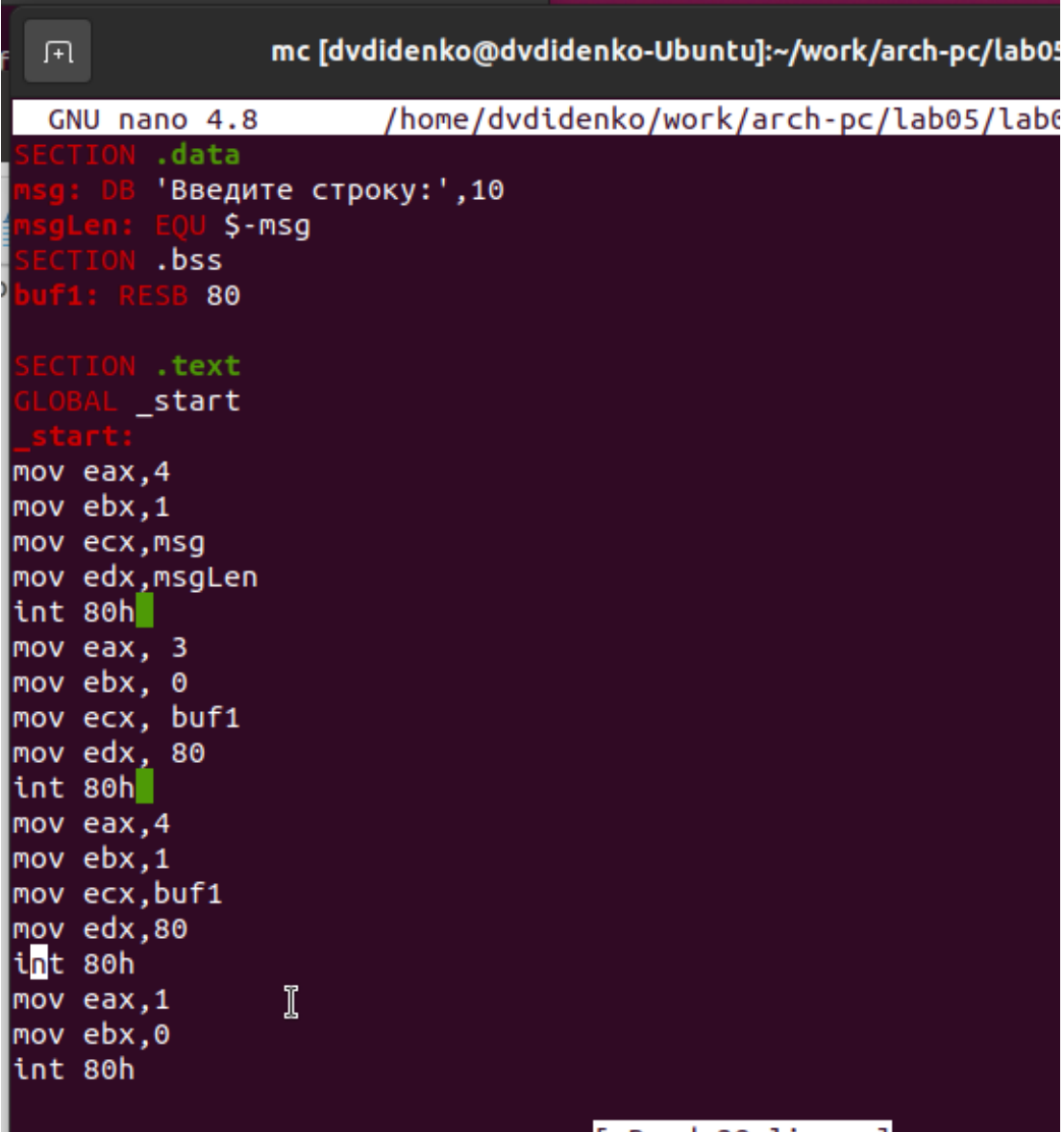
```
dvdidenko@dvdidenko-Ubuntu:~/work/arch-pc/lab05-2$  
dvdidenko@dvdidenko-Ubuntu:~/work/arch-pc/lab05-2$ nasm -f elf lab05-2.asm  
dvdidenko@dvdidenko-Ubuntu:~/work/arch-pc/lab05-2$ ld -m elf_i386 lab05-2.o -o lab05-2  
dvdidenko@dvdidenko-Ubuntu:~/work/arch-pc/lab05-2$ ./lab05-2  
Введите строку: Dmitriy  
dvdidenko@dvdidenko-Ubuntu:~/work/arch-pc/lab05-2$
```

Рис. 2.13: Компиляция и запуск программы lab05-2.asm

## 2.3 Задание для самостоятельной работы

Скопировал программу lab05-1.asm и внес изменения в код, чтобы программа работала по следующему алгоритму: выводила приглашение вида “Введите строку:”; считывала строку с клавиатуры; выводила введенную строку на экран.

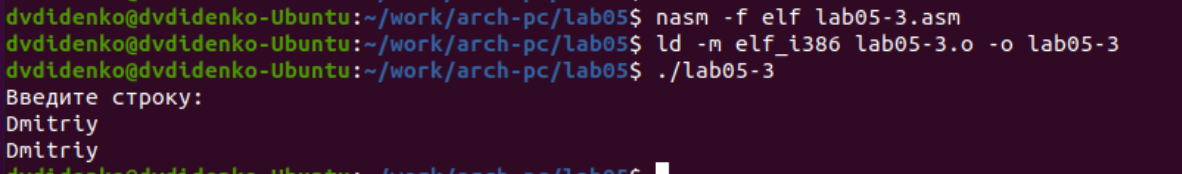
(рис. [2.14]) (рис. [2.15])



```
mc [dvdidenko@dvdidenko-Ubuntu]:~/work/arch-pc/lab05$
GNU nano 4.8 /home/dvdidenko/work/arch-pc/lab05/lab05-3.asm
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. 2.14: Код программы lab05-3.asm

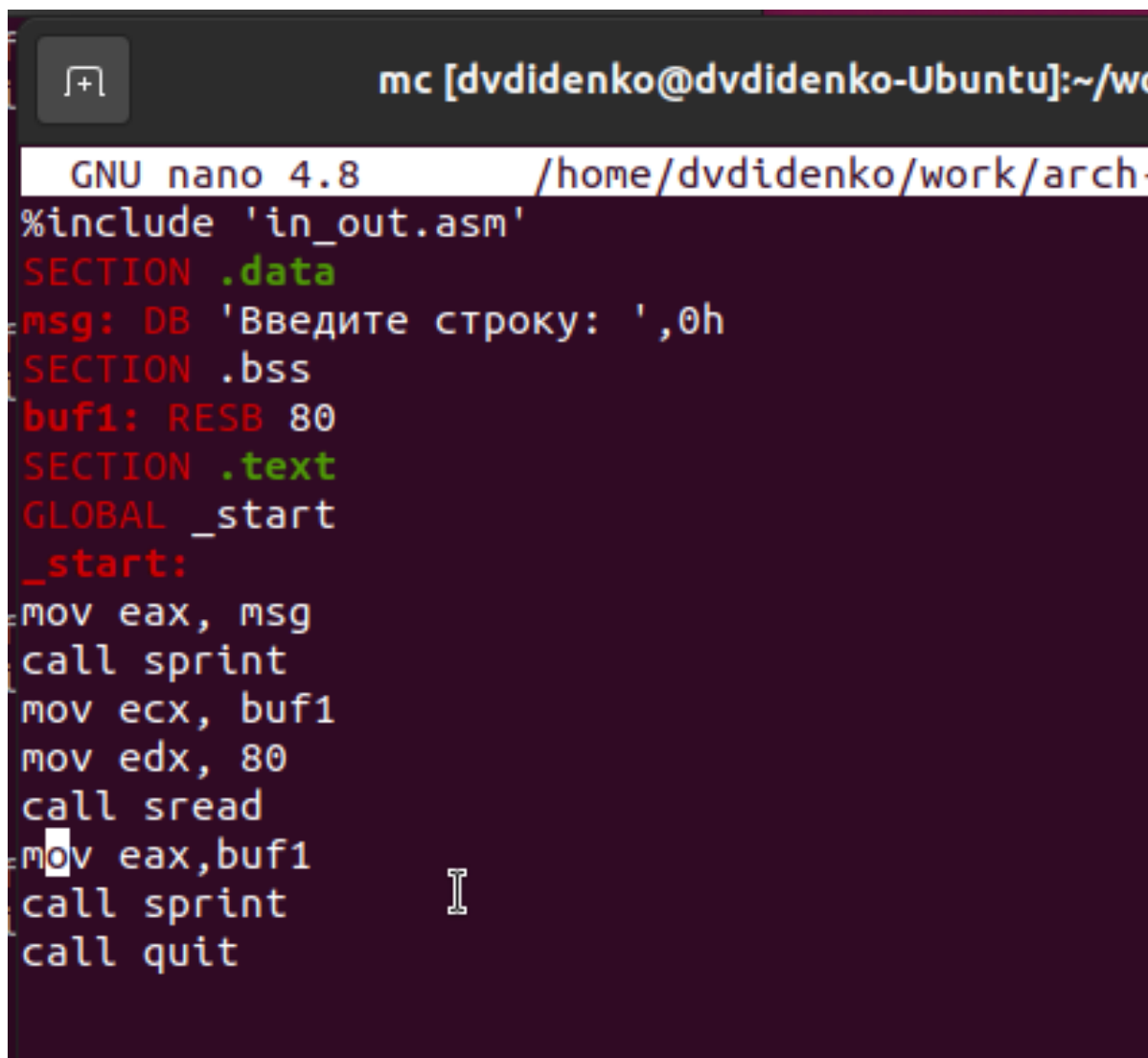


```
dvdidenko@dvdidenko-Ubuntu:~/work/arch-pc/lab05$ nasm -f elf lab05-3.asm
dvdidenko@dvdidenko-Ubuntu:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-3.o -o lab05-3
dvdidenko@dvdidenko-Ubuntu:~/work/arch-pc/lab05$ ./lab05-3
Введите строку:
Dmitriy
Dmitriy
```

Рис. 2.15: Компиляция и запуск программы lab05-3.asm

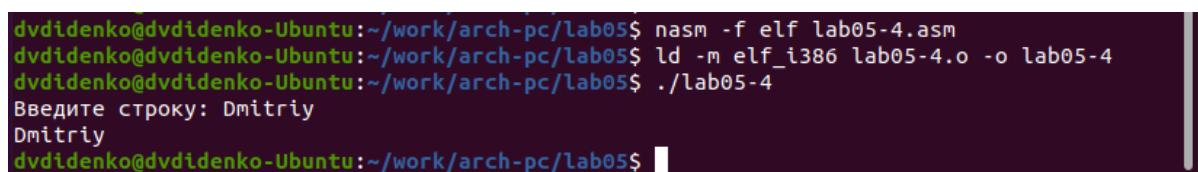


Аналогично скопировал программу lab05-2.asm и изменил код, но теперь использовал подпрограммы из файла in\_out.asm. (рис. [2.16]) (рис. [2.17])



```
mc [dvdidenko@dvdidenko-Ubuntu]:~/wo
GNU nano 4.8 /home/dvdidenko/work/arch-
%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. 2.16: Код программы lab05-4.asm



```
dvdidenko@dvdidenko-Ubuntu:~/work/arch-pc/lab05$ nasm -f elf lab05-4.asm
dvdidenko@dvdidenko-Ubuntu:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-4.o -o lab05-4
dvdidenko@dvdidenko-Ubuntu:~/work/arch-pc/lab05$ ./lab05-4
Введите строку: Dmitriy
Dmitriy
dvdidenko@dvdidenko-Ubuntu:~/work/arch-pc/lab05$
```

Рис. 2.17: Компиляция и запуск программы lab05-4.asm

## 3 Выводы

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