

Отчёт по лабораторной работе 5

Архитектура компьютера

Матиев Даниэль Саматович

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	10
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	Программа в файле lab05-1.asm	8
2.5	Просмотр файла lab05-1.asm	9
2.6	Запуск программы lab05-1.asm	10
2.7	Копирование файла in_out.asm	11
2.8	Копирование файла lab05-1.asm	12
2.9	Программа в файле lab05-2.asm	13
2.10	Запуск программы lab05-2.asm	13
2.11	Программа в файле lab05-2.asm	14
2.12	Запуск программы lab05-2.asm	15
2.13	Копирование файла lab05-1.asm	15
2.14	Программа в файле lab05-3.asm	16
2.15	Запуск программы lab05-3.asm	16
2.16	Копирование файла lab05-2.asm	17
2.17	Программа в файле lab05-4.asm	18
2.18	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

Открыл Midnight Commander, с помощью клавишь со стрелками и Enter перешел в каталог ~/work/arch-рс. Далее нажал F7 и создал каталог lab05

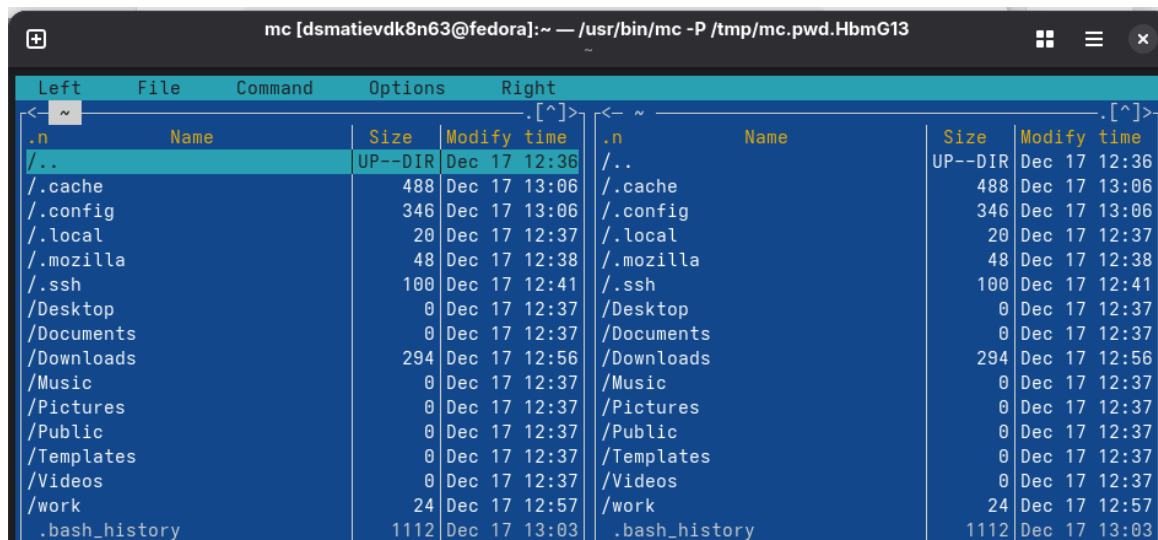


Рисунок 2.1: Запуск Midnight Commander

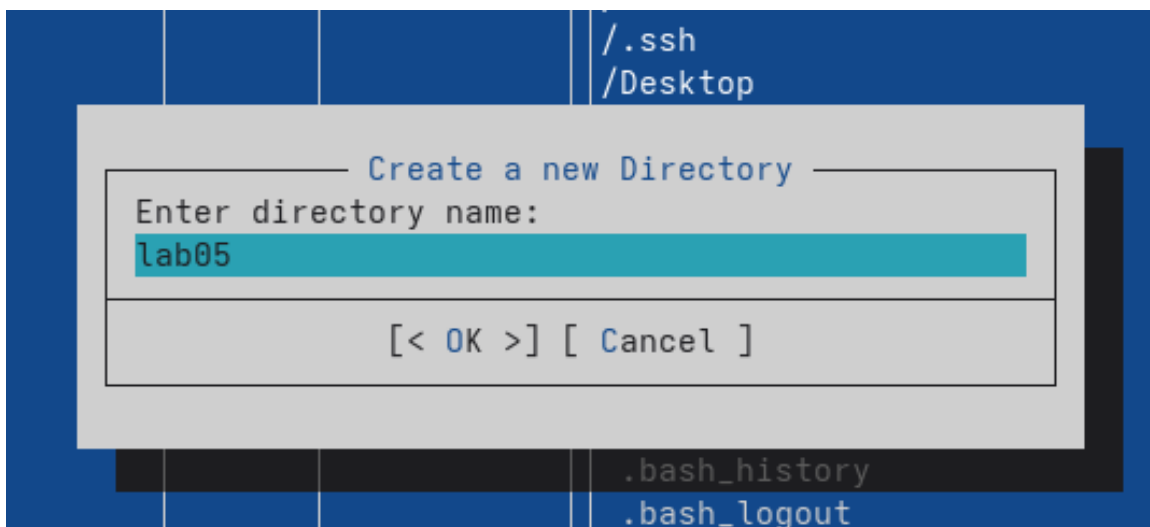


Рисунок 2.2: Создание каталога

При помощи touch создал файл lab05-1.asm

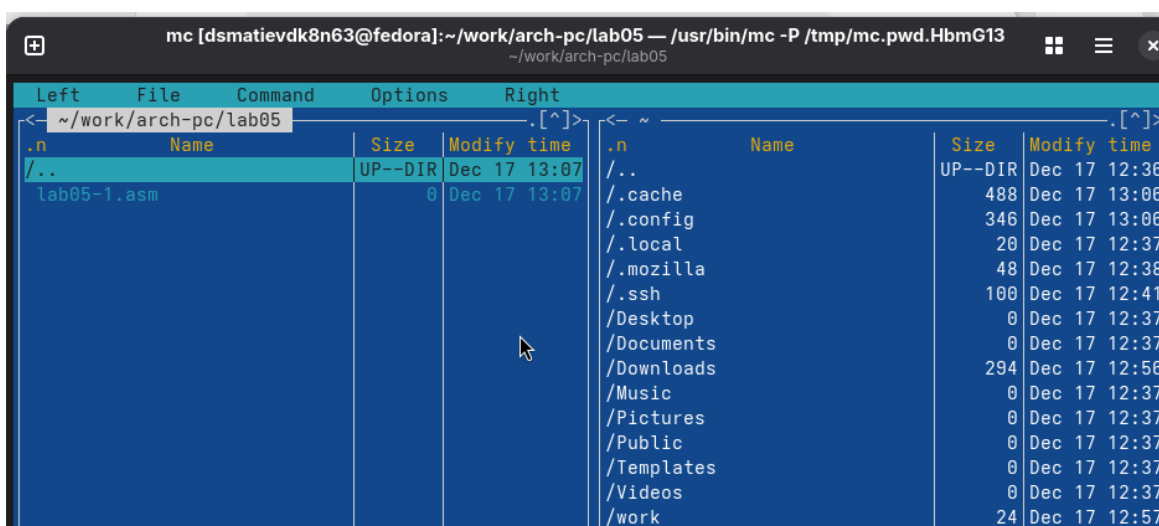


Рисунок 2.3: Создание файла lab05-1.asm

Открыл файл на редактирование клавишей F4, выбрал редактор mceditor, написал код программы из задания.

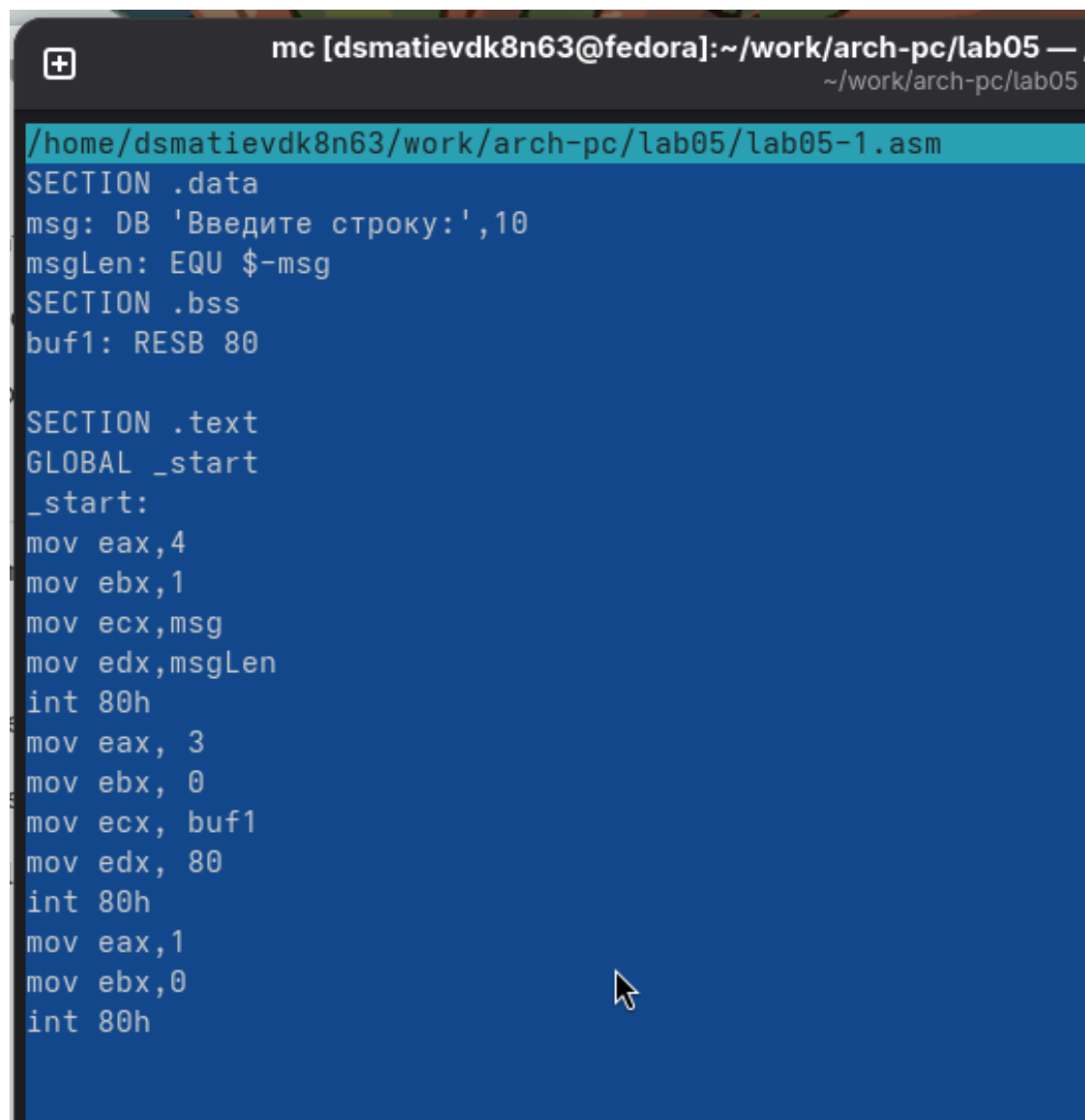


```
mc [dsmatievdk8n63@fedora]:~/work/arch
lab05-1.asm [----] 9 L:[ 1+10 11/ 23]
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80

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 2.4: Программа в файле lab05-1.asm

Открыл файл на просмотр клавишей F3 и убедился, что он содержит набран-
ный код.



```
mc [dsmatievdk8n63@fedora]:~/work/arch-pc/lab05 —  
~/work/arch-pc/lab05  
/home/dsmatievdk8n63/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
```

Рисунок 2.5: Просмотр файла lab05-1.asm

Транслировал файл программы в объектный файл, выполнил компоновку объектного файла, получил исполняемый файл программы и проверил ее работу.

```
dsmatievdk8n63@fedora:~/work/arch-pc/lab05$ nasm -f elf lab05-1.asm
dsmatievdk8n63@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-1.o -o lab05-1
dsmatievdk8n63@fedora:~/work/arch-pc/lab05$ ./lab05-1
Введите строку:
Daniel
dsmatievdk8n63@fedora:~/work/arch-pc/lab05$
```

Рисунок 2.6: Запуск программы lab05-1.asm

2.2 Подключение внешнего файла in_out.asm

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.

Для выполнения лабораторных работ используется файл in_out.asm, который содержит следующие подпрограммы:

- `slen` – вычисление длины строки (используется в подпрограммах печати сообщения для определения количества выводимых байтов);
- `sprint` – вывод сообщения на экран, перед вызовом `sprint` в регистр `eax` необходимо записать выводимое сообщение (`mov eax,;`);
- `sprintLF` – работает аналогично `sprint`, но при выводе на экран добавляет к сообщению символ перевода строки;
- `sread` – ввод сообщения с клавиатуры, перед вызовом `sread` в регистр `eax` необходимо записать адрес переменной в которую введенное сообщение будет записано (`mov eax,;`), в регистр `ebx` – длину вводимой строки (`mov ebx,;`);

- `iprint` – вывод на экран чисел в формате ASCII, перед вызовом `iprint` в регистр `eax` необходимо записать выводимое число (`mov eax,;`);
- `iprintLF` – работает аналогично `iprint`, но при выводе на экран после числа добавляет к символ перевода строки;
- `atoi` – функция преобразует `ascii`-код символа в целое число и записывает результат в регистр `eax`, перед вызовом `atoi` в регистр `eax` необходимо записать число (`mov eax,;`);
- `quit` – завершение программы.

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

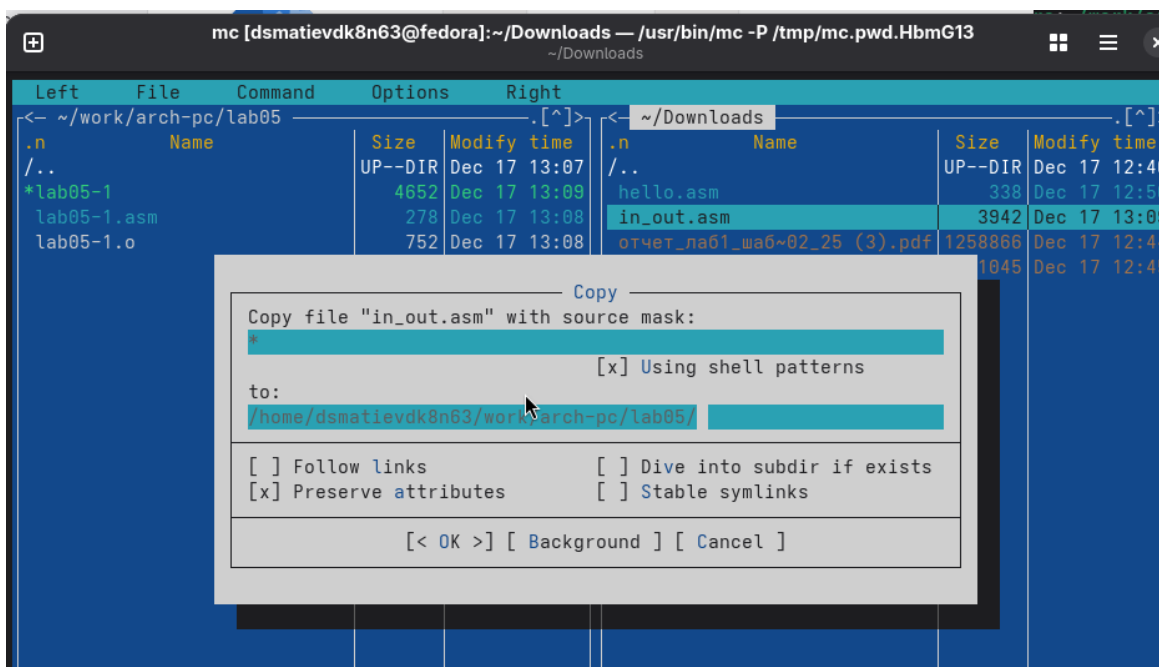


Рисунок 2.7: Копирование файла `in_out.asm`

Скопировал `lab05-1.asm` в `lab05-2.asm`.

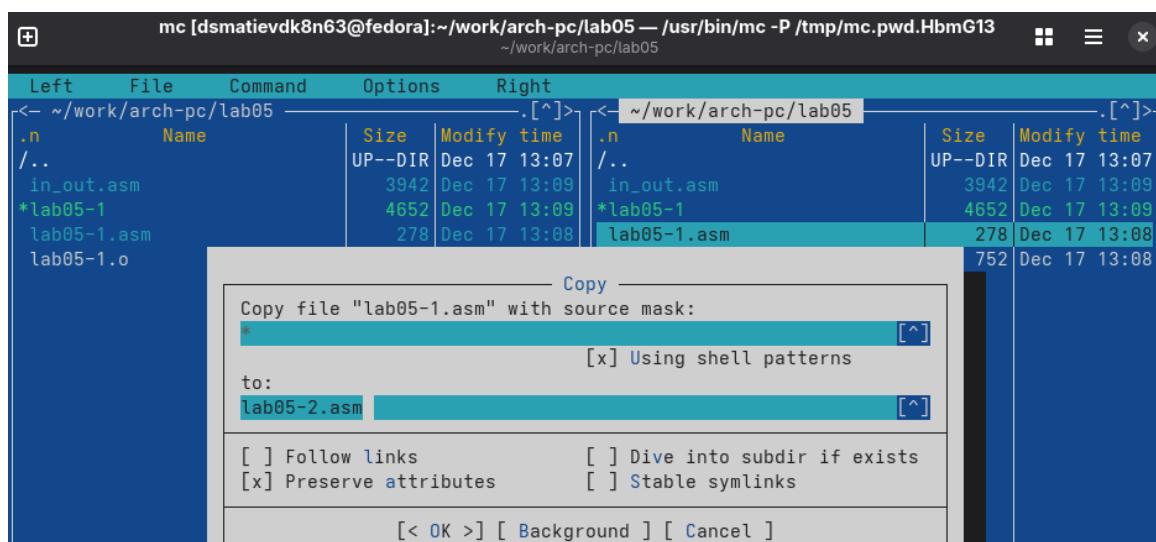
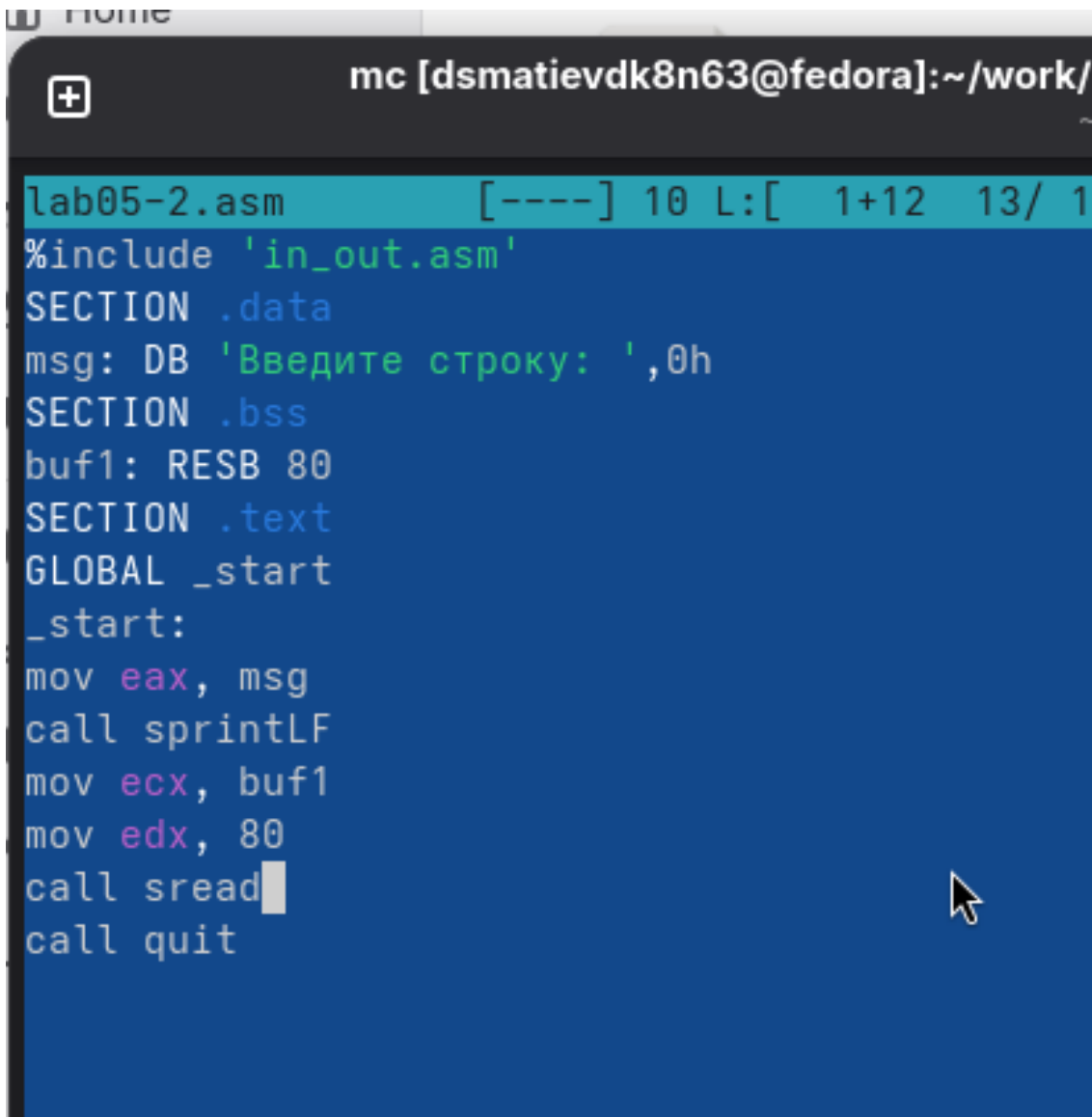


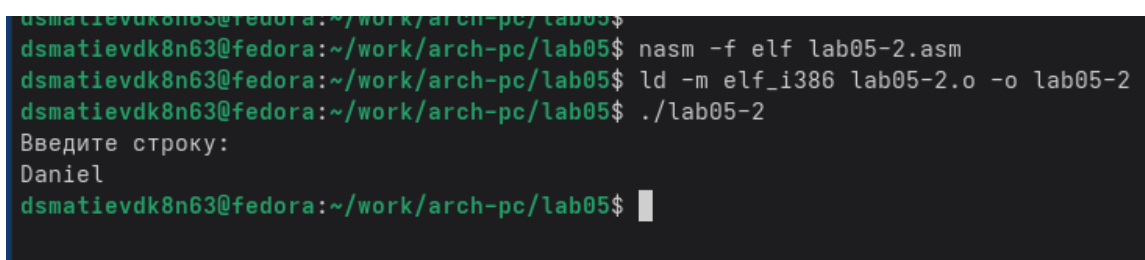
Рисунок 2.8: Копирование файла lab05-1.asm

Написал код программы lab05-2.asm с использованием подпрограмм из внешнего файла in_out.asm . Скомпилировал программу и проверил запуск.



```
lab05-2.asm [----] 10 L:[ 1+12 13/ 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 sprintLF
mov ecx, buf1
mov edx, 80
call sread
call quit
```

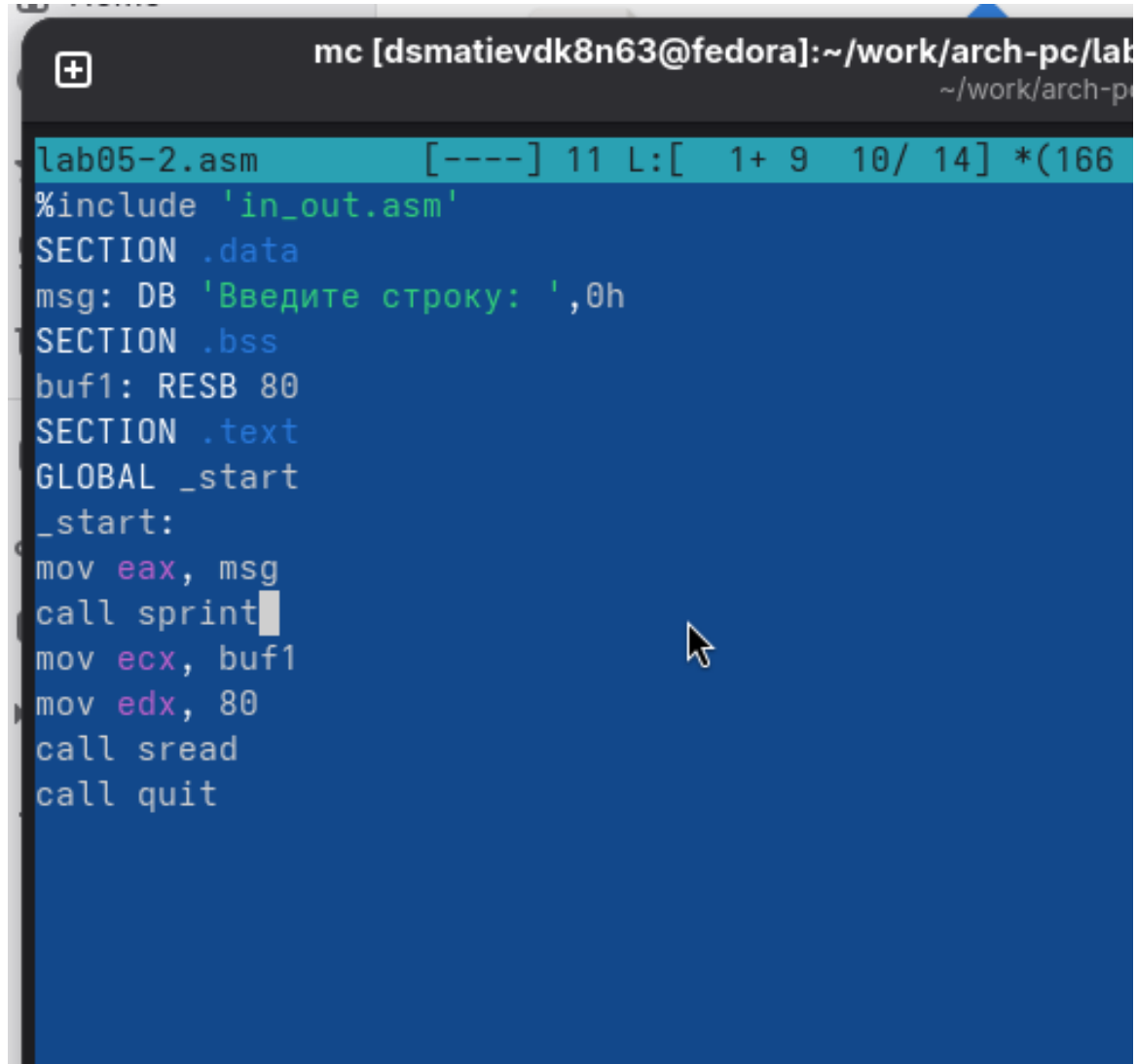
Рисунок 2.9: Программа в файле lab05-2.asm



```
dsmatievdk8n63@fedora:~/work/arch-pc/lab05$
dsmatievdk8n63@fedora:~/work/arch-pc/lab05$ nasm -f elf lab05-2.asm
dsmatievdk8n63@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2
dsmatievdk8n63@fedora:~/work/arch-pc/lab05$ ./lab05-2
Введите строку:
Daniel
dsmatievdk8n63@fedora:~/work/arch-pc/lab05$
```

Рисунок 2.10: Запуск программы lab05-2.asm

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



```
mc [dsmatievdk8n63@fedora]:~/work/arch-pc/lab
lab05-2.asm [----] 11 L:[ 1+ 9 10/ 14] *(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 2.11: Программа в файле lab05-2.asm

```
dsmatievdk8n63@fedora:~/work/arch-pc/lab05$ nasm -f elf lab05-2.asm
dsmatievdk8n63@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-2.o -o lab05-2
dsmatievdk8n63@fedora:~/work/arch-pc/lab05$ ./lab05-2
Введите строку: Daniel
dsmatievdk8n63@fedora:~/work/arch-pc/lab05$
```

Рисунок 2.12: Запуск программы lab05-2.asm

2.3 Задание для самостоятельной работы

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

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

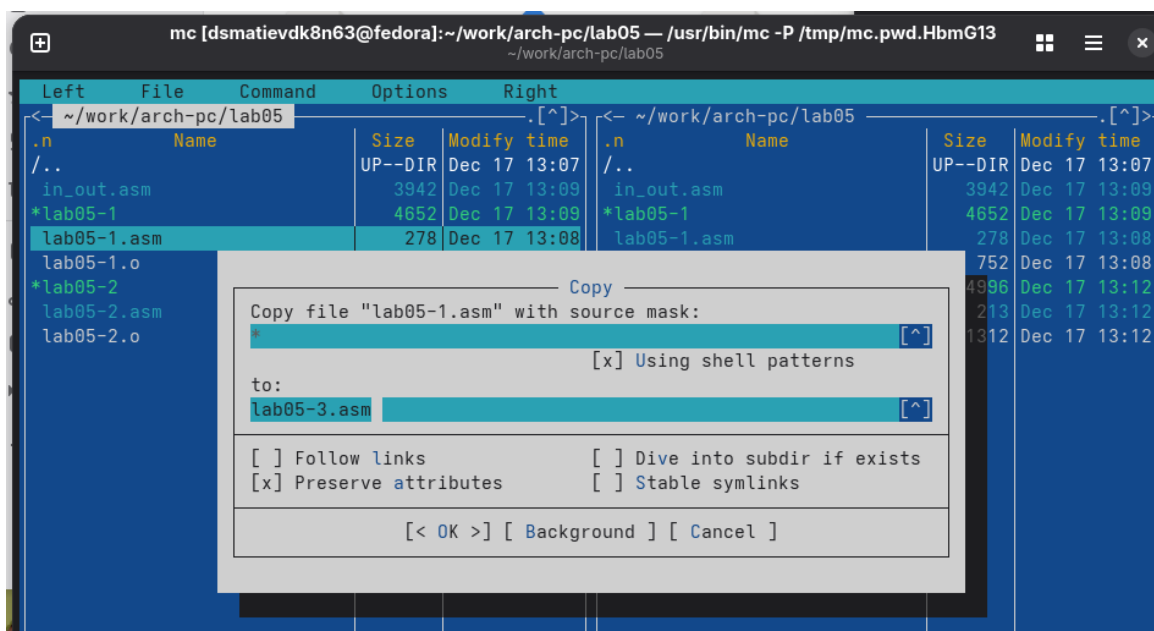


Рисунок 2.13: Копирование файла lab05-1.asm

```
mc [dsmatievdk8n63@fedora]:~/work/arch-  
~/work/  
lab05-3.asm [-----] 10 L: [ 1+15 16/ 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
```

Рисунок 2.14: Программа в файле lab05-3.asm

```
dsmatievdk8n63@fedora:~/work/arch-pc/lab05$  
dsmatievdk8n63@fedora:~/work/arch-pc/lab05$ nasm -f elf lab05-3.asm  
dsmatievdk8n63@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-3.o -o lab05-3  
dsmatievdk8n63@fedora:~/work/arch-pc/lab05$ ./lab05-3  
Введите строку:  
Daniel  
Daniel  
dsmatievdk8n63@fedora:~/work/arch-pc/lab05$
```

Рисунок 2.15: Запуск программы lab05-3.asm

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

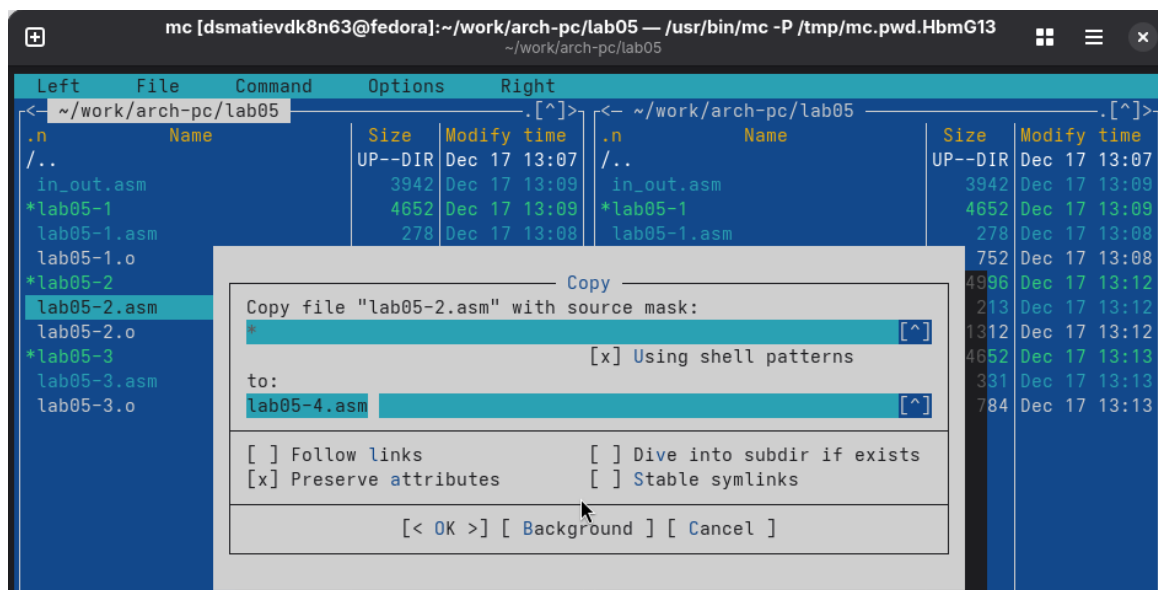
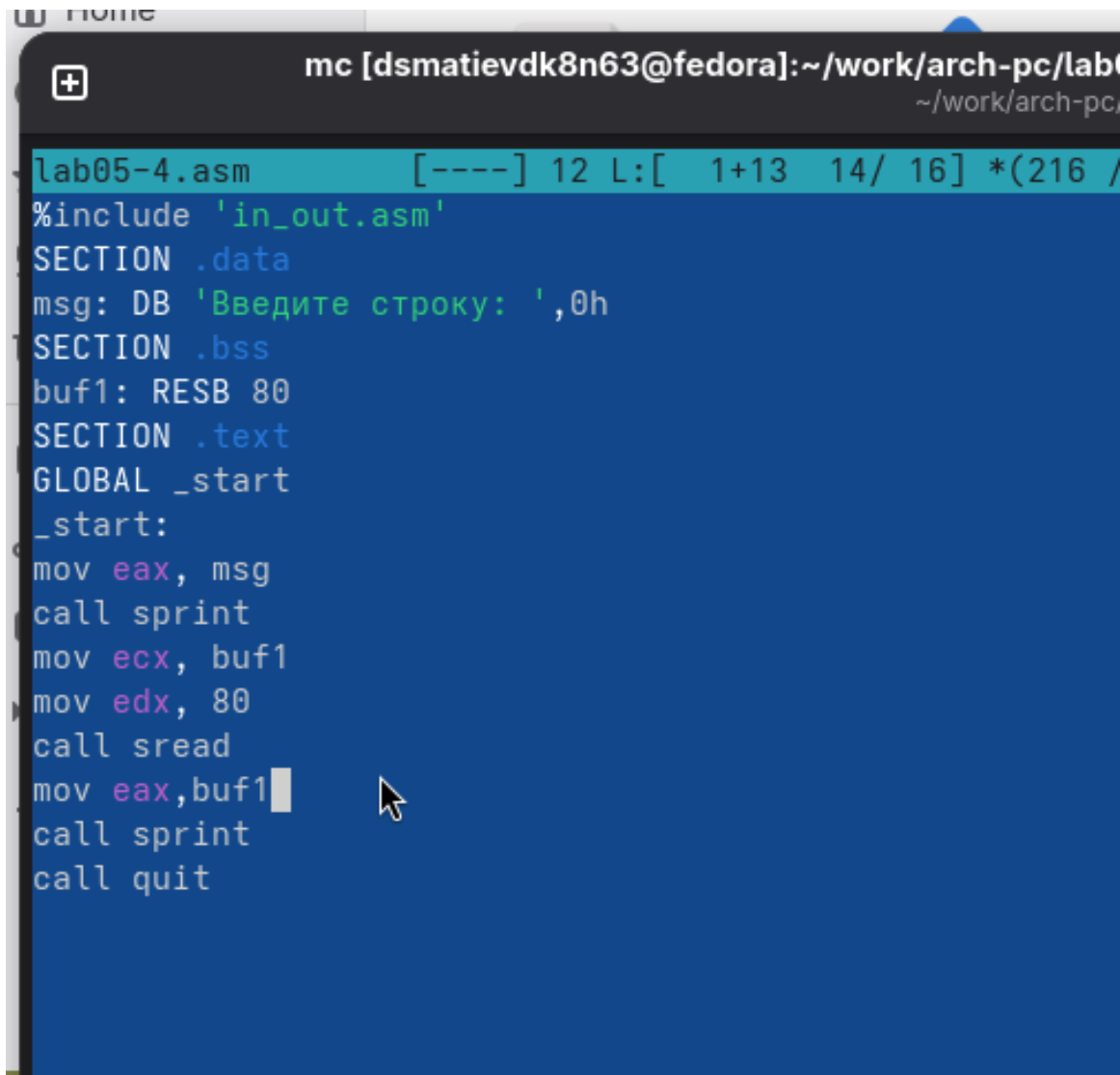
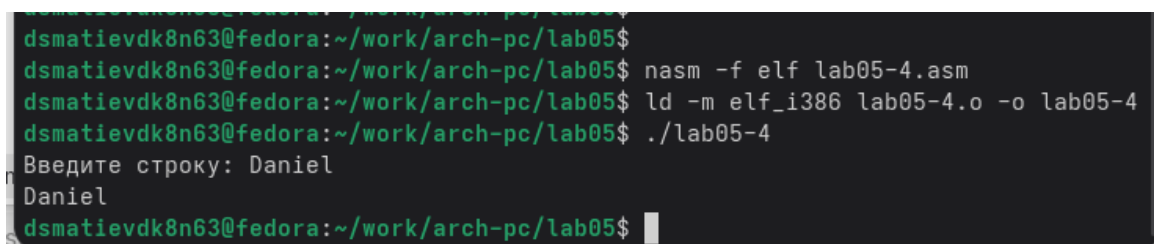


Рисунок 2.16: Копирование файла lab05-2.asm



```
mc [dsmatievdk8n63@fedora]:~/work/arch-pc/lab05$  
lab05-4.asm [----] 12 L:[ 1+13 14/ 16] *(216 /  
%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 2.17: Программа в файле lab05-4.asm



```
dsmatievdk8n63@fedora:~/work/arch-pc/lab05$  
dsmatievdk8n63@fedora:~/work/arch-pc/lab05$ nasm -f elf lab05-4.asm  
dsmatievdk8n63@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 lab05-4.o -o lab05-4  
dsmatievdk8n63@fedora:~/work/arch-pc/lab05$ ./lab05-4  
Введите строку: Daniel  
Daniel  
dsmatievdk8n63@fedora:~/work/arch-pc/lab05$
```

Рисунок 2.18: Запуск программы lab05-4.asm

3 Выводы

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