

Лабораторная работа №5

Архитектура компьютера и операционные системы

Бабенко Константин, НКАбд-01-23

Содержание

1	Цель работы	1
2	Выполнение лабораторной работы.....	1
2.1	Домашняя работа.....	Ошибка! Закладка не определена.
3	Выводы	9

1 Цель работы

Приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера mov и int.

2 Выполнение лабораторной работы

1) Открываю Midnight Commander:

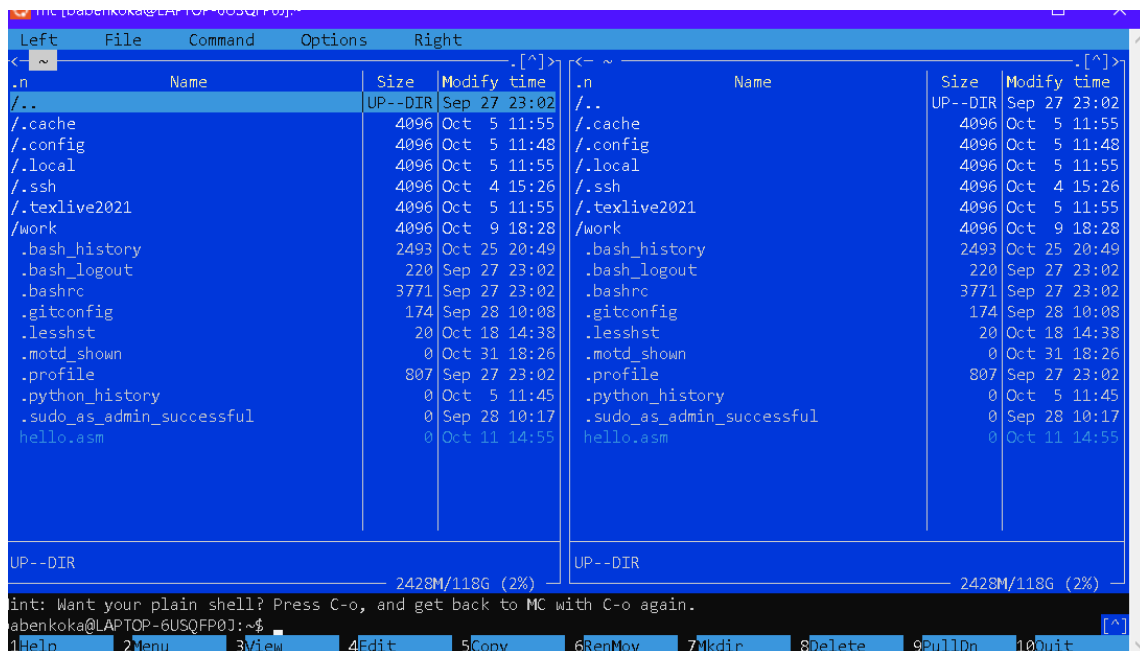


Figure 1: Midnight Commander .

2) Перехожу в каталог ~/work/arch-pc созданный при выполнении лабораторной работы №4:

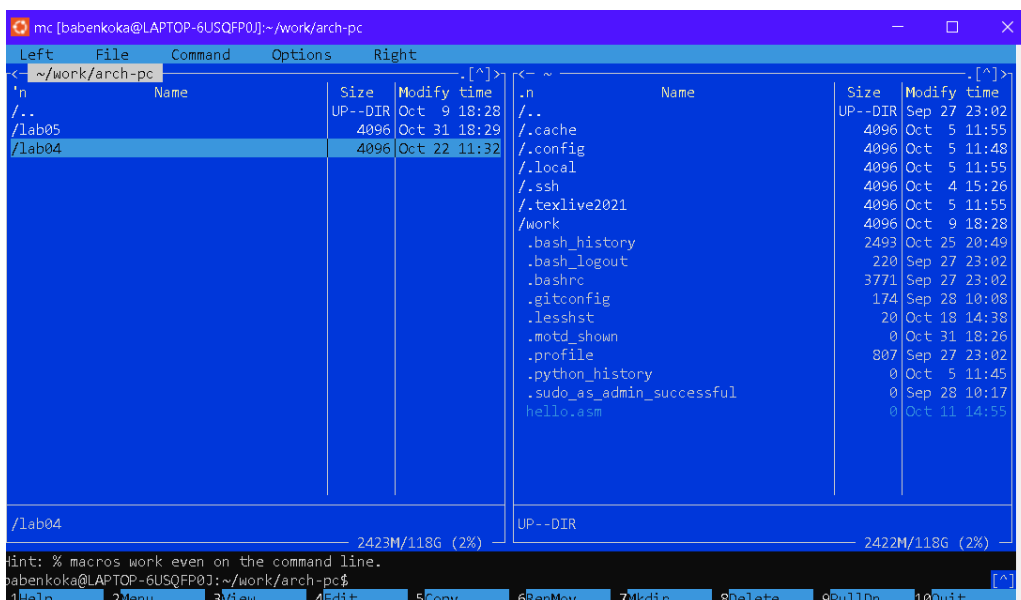


Figure 2: Перехожу в каталог ~/work/arch-pc.

3) Создаю папку lab05 и перехожу в созданный каталог:

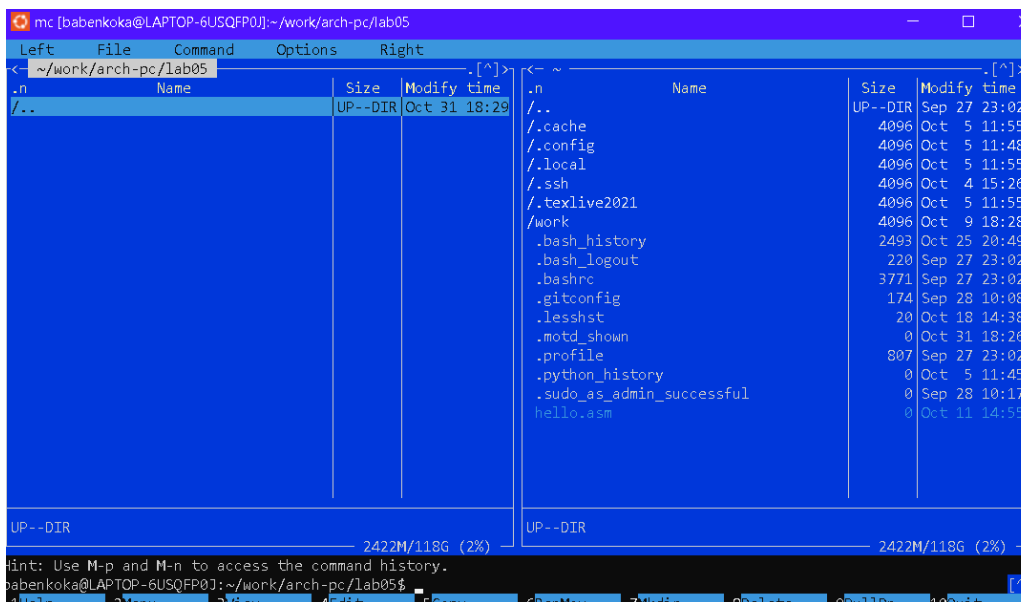


Figure 3: Перехожу в папку lab05.

4) Создаю файл lab5-1.asm:

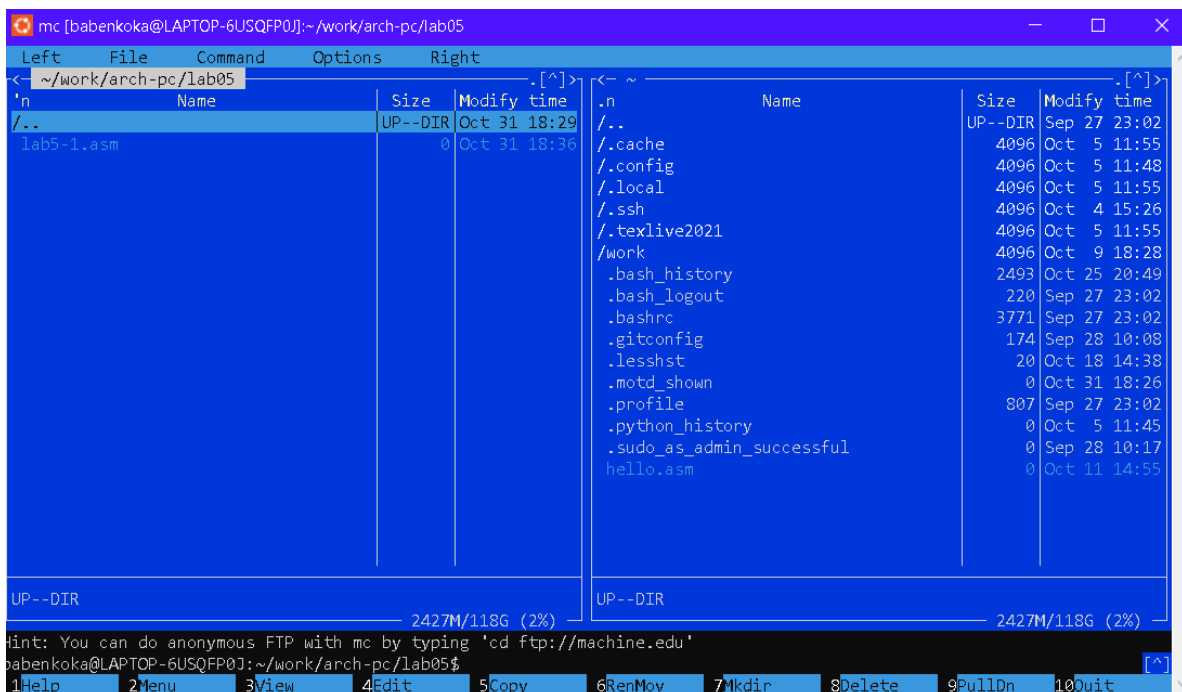


Figure 4: Создаю файл lab5-1.asm.

5) Открываю файл lab5-1.asm для редактирования во встроенном редакторе:

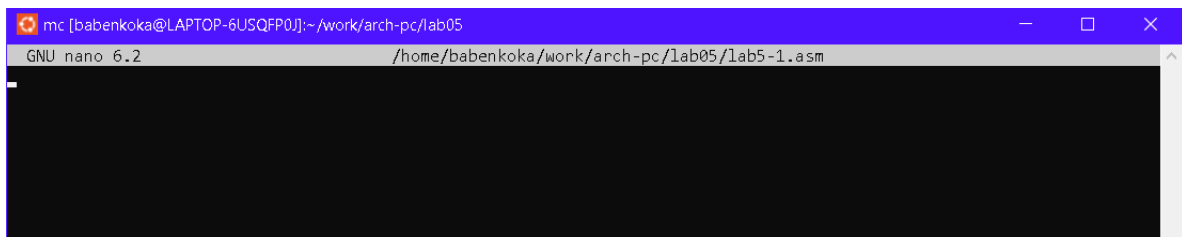
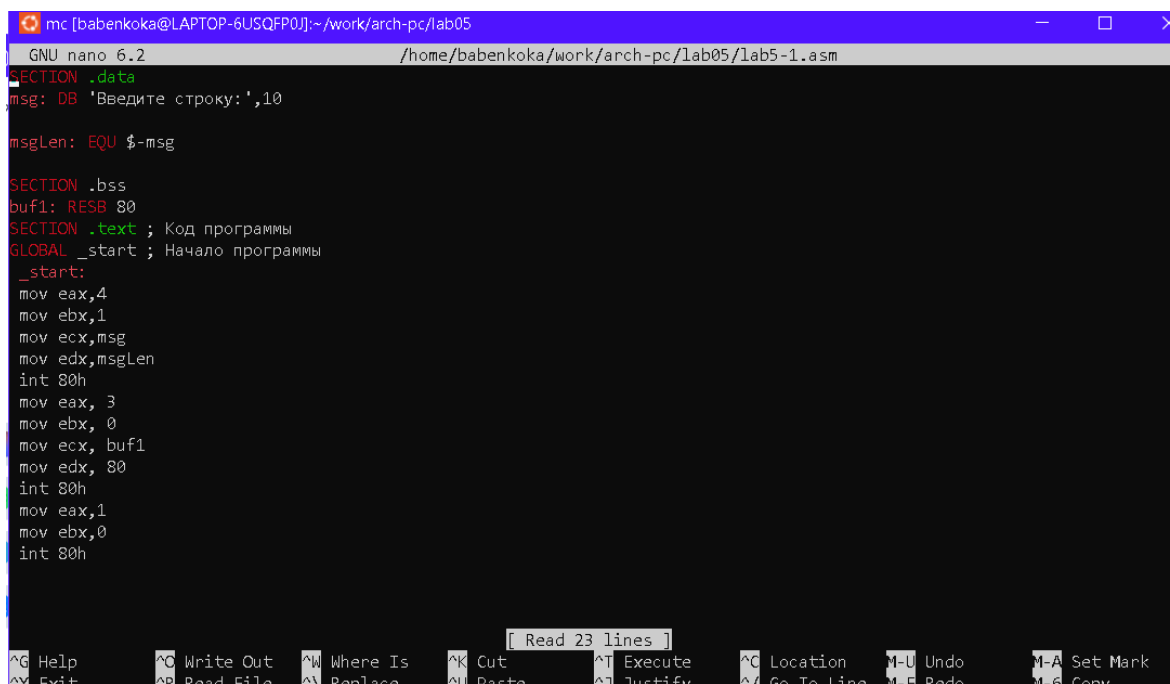


Figure 5: Открываю файл lab5-1.asm.

6) Ввожу текст программы из листинга 5.1:



```
mc [babenkoka@LAPTOP-6USQFP0J]:~/work/arch-pc/lab05
GNU nano 6.2 /home/babenkoka/work/arch-pc/lab05/lab5-1.asm
SECTION .data
msg: DB 'Введите строку:',10

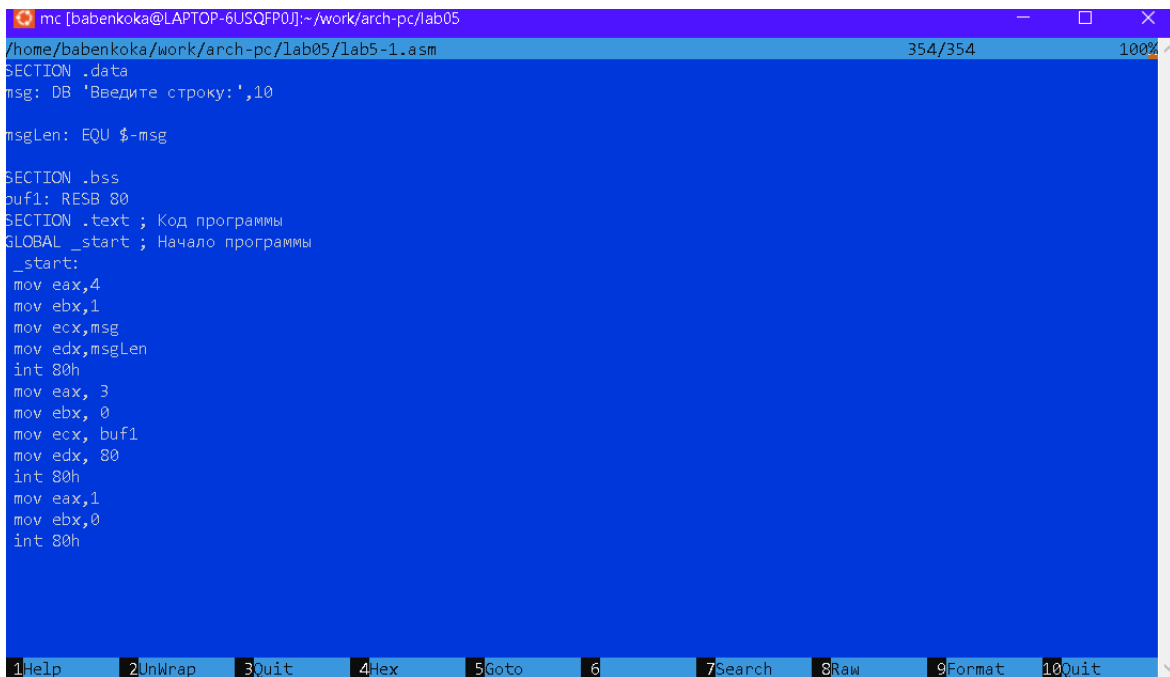
msgLen: EQU $-msg

SECTION .bss
buf1: RESB 80
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_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

[ Read 23 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   ^U Undo       ^M Set Mark
^X Exit      ^R Read File  ^N Replace    ^L Paste      ^J Justify    ^G Go To Line ^E Redo       ^M Copy
```

Figure 6: Программа.

7) Открываю файл lab5-1.asm для просмотра. Убеждаюсь, что файл содержит текст программы:



```
mc [babenkoka@LAPTOP-6USQFP0J]:~/work/arch-pc/lab05
/home/babenkoka/work/arch-pc/lab05/lab5-1.asm 354/354 100%
SECTION .data
msg: DB 'Введите строку:',10

msgLen: EQU $-msg

SECTION .bss
buf1: RESB 80
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_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

1Help 2UnWrap 3Quit 4Hex 5Goto 6 7Search 8Raw 9Format 10Quit
```

Figure 7: Открываю файл для просмотра.

8) Транслирую текст программы в объектный файл. Выполняю компоновку объектного файла:

```

babenkoka@LAPTOP-6USQFP0J:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
babenkoka@LAPTOP-6USQFP0J:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o

```

Figure 8: Трансляция и компоновка файла.

Запускаю получившийся исполняемый файл:

```

root@LAPTOP-6USQFP0J:/home# ./lab5-1
Введите строку:
Бабенко Константин Алексеевич

```

Figure 9: Работа файла.

Подключение внешнего файла in_out.asm

9) Скачиваю файл in_out.asm со страницы курса в ТУИС:

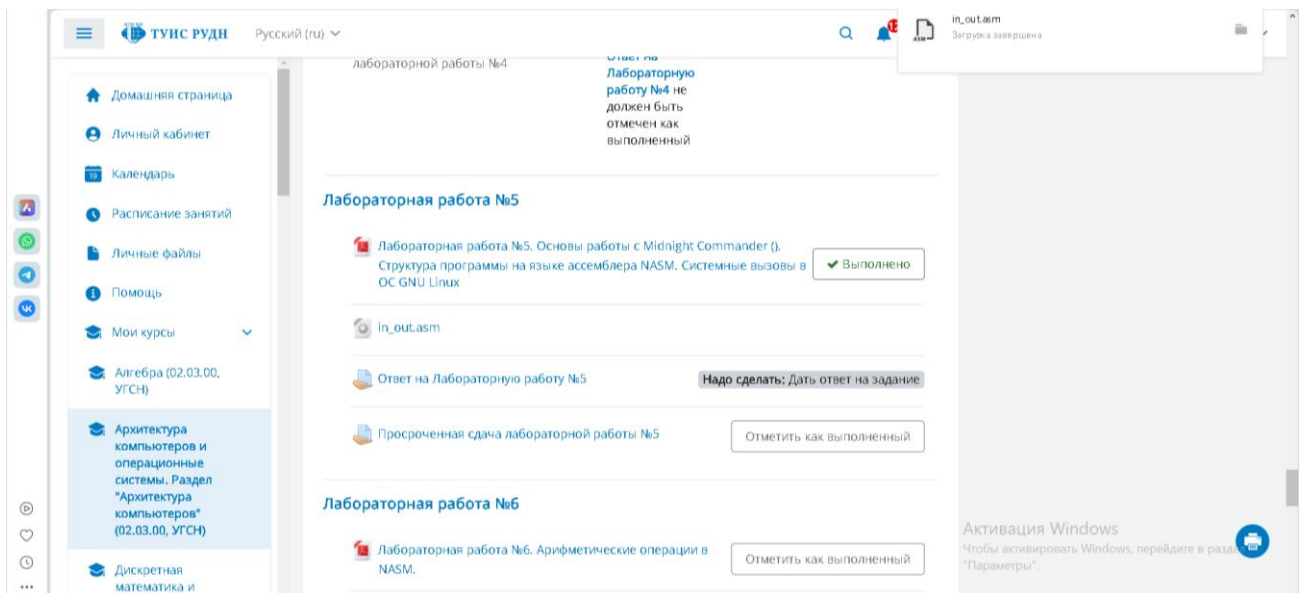


Figure 10: Скачиваю файл.

10) Подключаемый файл должен лежать в том же каталоге, что и файл с программой, в которой он используется, переносу его сюда:



Figure 11: Перенос файла.

11) С помощью функциональной клавиши F6 создаю копию файла lab5-1.asm с именем lab5-2.asm:

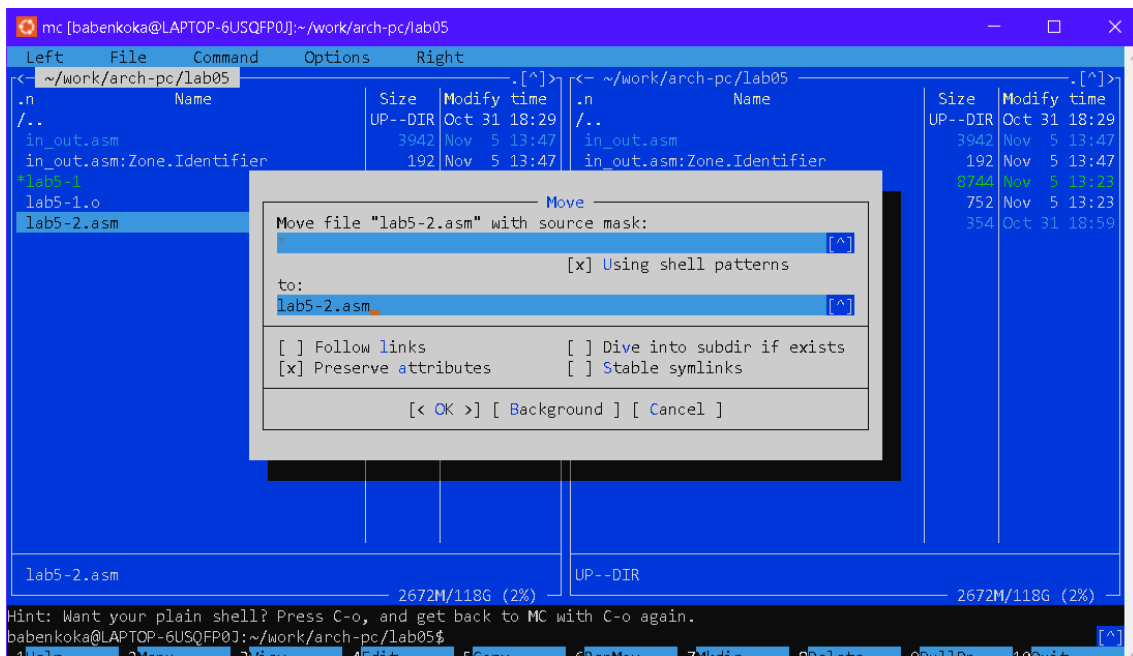


Figure 12: Копирование файла.

12) Исправляю текст программы в файле lab5-2.asm с использованием файла in_out.asm в соответствии с листингом 5.2:

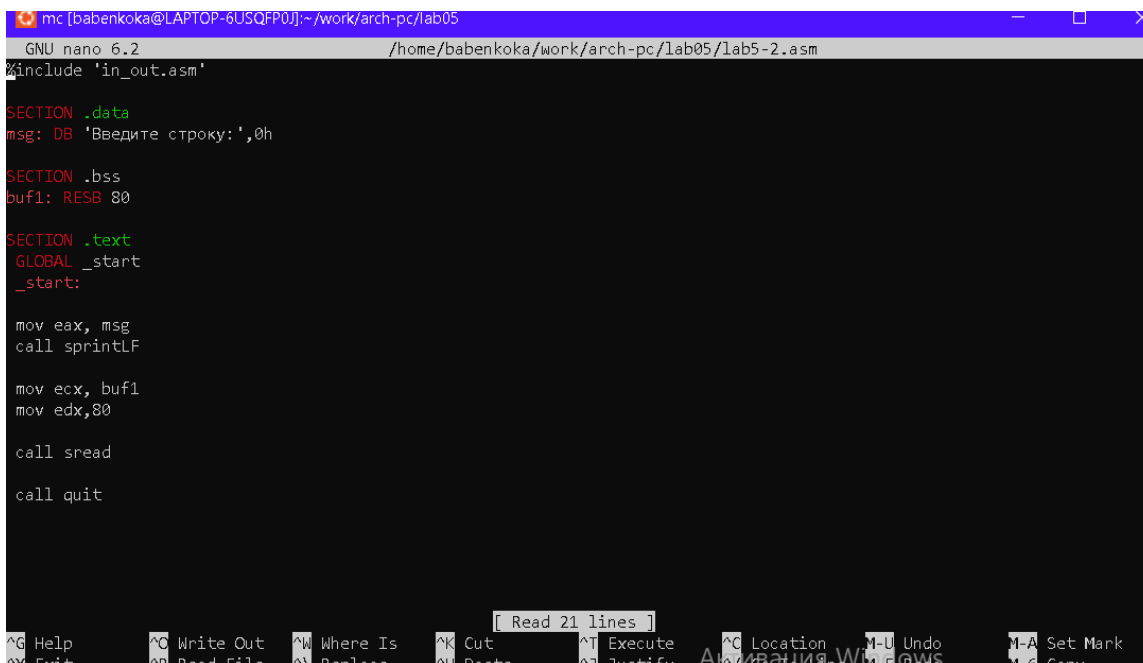


Figure 13: Изменение файла.

Создаю исполняемый файл и проверяю его работу:

```

babenkoka@LAPTOP-6USQFP0J: ~/work/arch-pc/lab05
babenkoka@LAPTOP-6USQFP0J:~$ cd ~/work/arch-pc/lab05
babenkoka@LAPTOP-6USQFP0J:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
babenkoka@LAPTOP-6USQFP0J:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
babenkoka@LAPTOP-6USQFP0J:~/work/arch-pc/lab05$ ls
in_out.asm  in_out.asm:Zone.Identifier  lab5-1  lab5-1.o  lab5-2  lab5-2.asm  lab5-2.o
babenkoka@LAPTOP-6USQFP0J:~/work/arch-pc/lab05$

```

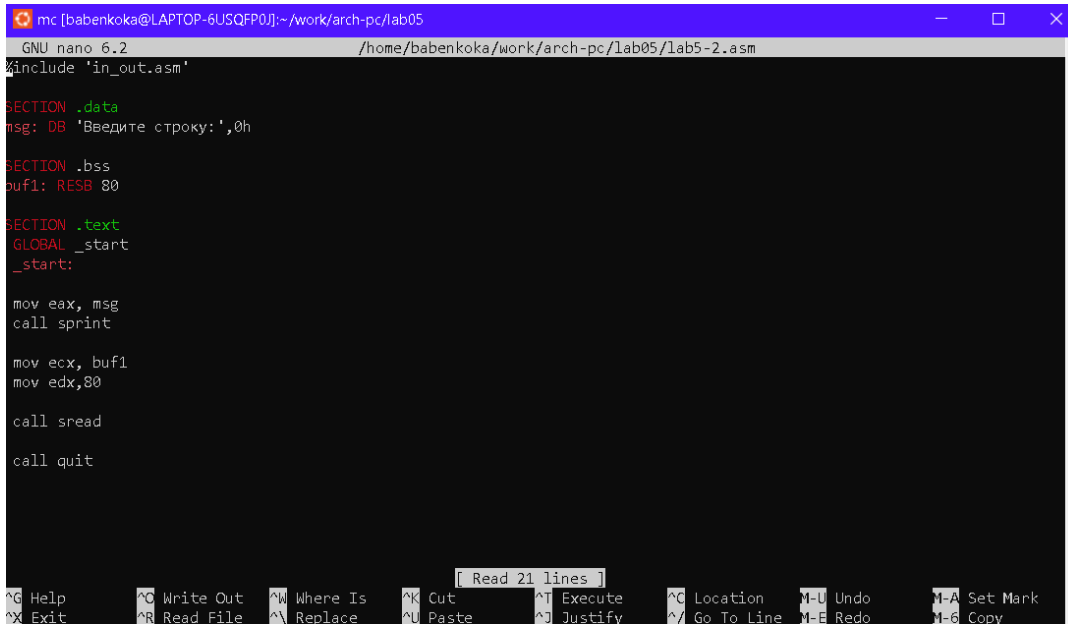
```

root@LAPTOP-6USQFP0J:/home# ./lab5-2
Введите строку:
бабенко Константин Алексеевич

```

Figure 14: Работа нового файла.

13) В файле lab5-2.asm заменяю подпрограмму `sprintLF` на `sprint`:



```

GNU nano 6.2 /home/babenkoka/work/arch-pc/lab05/lab5-2.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

```

Figure 15: Замена подпрограммы.

Создаю исполняемый файл и проверяю его работу:

```

babenkoka@LAPTOP-6USQFP0J:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
babenkoka@LAPTOP-6USQFP0J:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o

root@LAPTOP-6USQFP0J:/home# ./lab5-2
Введите строку:Бабенко Константин Алексеевич

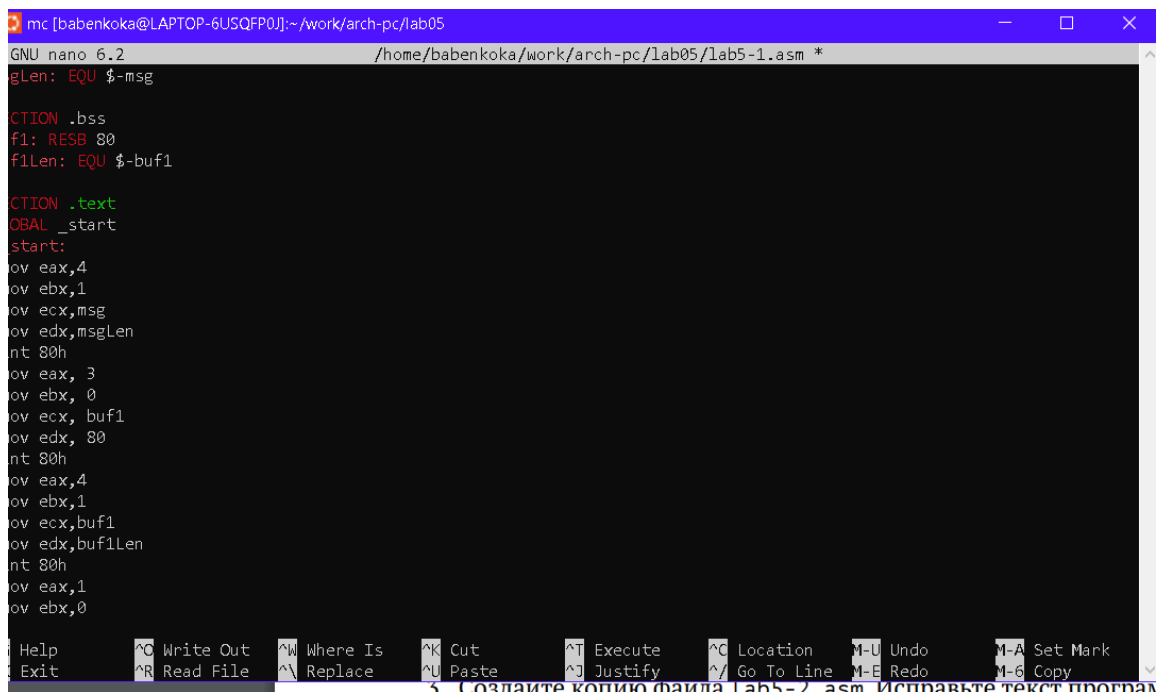
```

Figure 16: Работа измененной программы.

Измененная программа не переводит регистр на следующую строку в отличие от первой.

2.1 Домашняя работа

1. Вношу изменения в первую программу, так чтобы она выводила введенную строку:



```
mc [babenkoka@LAPTOP-6USQFP0J]:~/work/arch-pc/lab05
GNU nano 6.2 /home/babenkoka/work/arch-pc/lab05/lab5-1.asm *
gLen: EQU $-msg

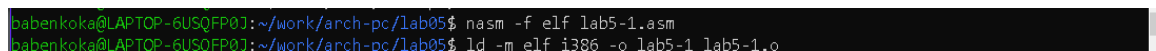
CTION .bss
f1: RESB 80
f1Len: EQU $-buf1

CTION .text
OBAL _start
start:
ov eax,4
ov ebx,1
ov ecx,msg
ov edx,msgLen
nt 80h
ov eax, 3
ov ebx, 0
ov ecx, buf1
ov edx, 80
nt 80h
ov eax,4
ov ebx,1
ov ecx,buf1
ov edx,buf1Len
nt 80h
ov eax,1
ov ebx,0

Help Write Out Where Is Cut Execute Location M-U Undo M-A Set Mark
Exit Read File Replace Paste Justify Go To Line M-E Redo M-G Copy
```

Figure 17: Изменяю первый файл.

2. Получаю исполняемый файл:



```
babenkoka@LAPTOP-6USQFP0J:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
babenkoka@LAPTOP-6USQFP0J:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
```

Figure 18: Получение файла.

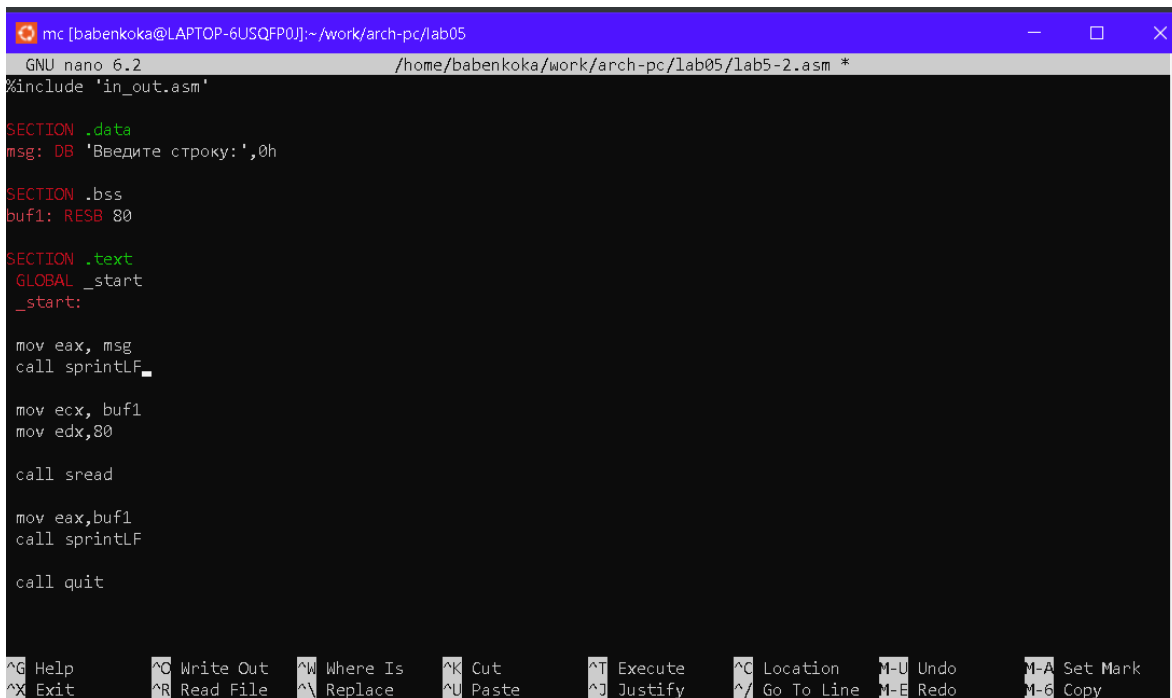
Проверяю его работу:



```
babenkoka@LAPTOP-6USQFP0J:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Бабенко
```

Figure 19: Работа файла.

3. Вношу изменения в вторую программу с использование подпрограмм из внешнего файла in_out.asm, так чтобы она выводила введенную строку:



```
mc [babenkoka@LAPTOP-6USQFP0J]:~/work/arch-pc/lab05
GNU nano 6.2 /home/babenkoka/work/arch-pc/lab05/lab5-2.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 sprintf

mov ecx, buf1
mov edx, 80

call sread

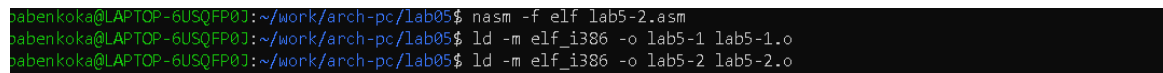
mov eax, buf1
call sprintf

call quit

^G Help      ^C Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark
^X Exit      ^R Read File  ^N Replace    ^U Paste      ^J Justify    ^_/ Go To Line  M-E Redo      M-G Copy
```

Figure 20: Изменяю второй файл.

4. Создаю исполняемый файл:



```
babenkoka@LAPTOP-6USQFP0J:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
babenkoka@LAPTOP-6USQFP0J:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
babenkoka@LAPTOP-6USQFP0J:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
```

Figure 21: Создание файла.

Запускаю файл:



```
root@LAPTOP-6USQFP0J:/home# ./lab5-2
Введите строку:Бабенко
Бабенко
```

Figure 22: Работа файла.

3 Выводы

Я приобрёл практические навыки работы в Midnight Commander, освоил инструкции языка ассемблера mov и int.