

Лабораторная работа №5

**Основы работы с Midnight Commander (mc). Структура программы
на языке ассемблера NASM. Системные вызовы в ОС GNU Linux**

Прозорова Елизавета Евгеньевна

Содержание

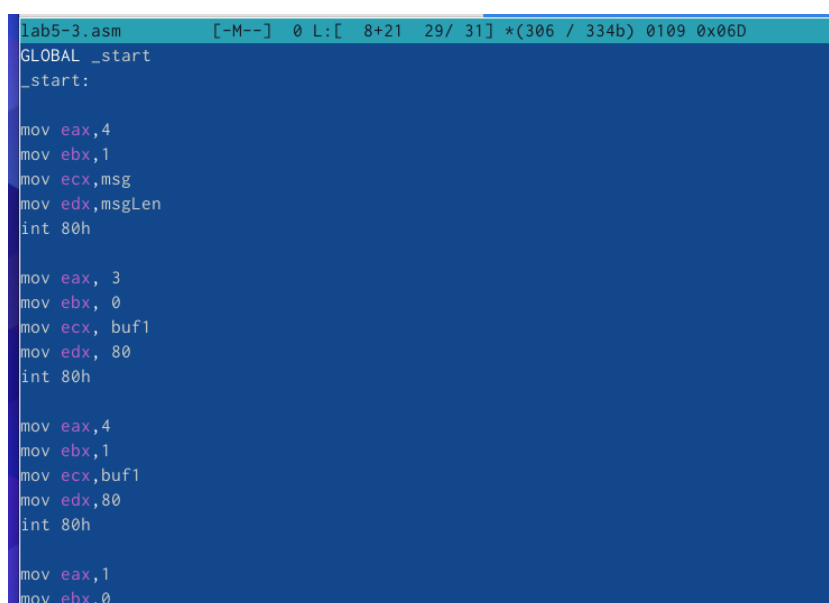
1	Цель работы	3
2	Выполнение лабораторной работы	4
3	Выполнение самостоятельной работы	9
4	Выводы	12

1 Цель работы

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

2 Выполнение лабораторной работы

1. С помощью команды ms открыла Midnight Commande



```
lab5-3.asm [-M--] 0 L:[ 8+21 29/ 31] *(306 / 334b) 0109 0x06D
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
```

Рис. 2.1: Midnight Commande

2. Пользуясь клавиатурой и функциональной клавишей F7 создала папку lab05 и перешла в созданный каталог.

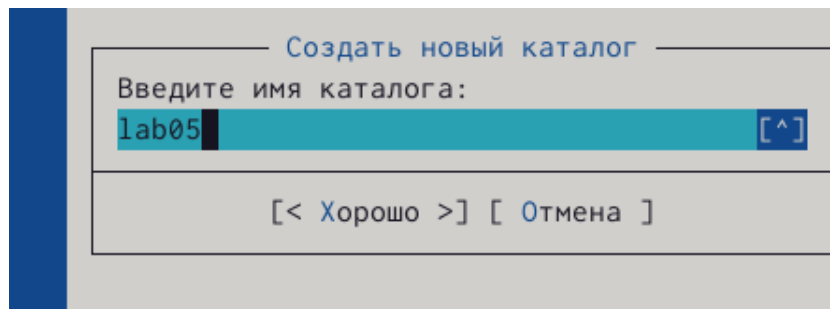


Рис. 2.2: Создание папки lab05

3. Пользуясь строкой ввода и командой touch создала файл lab5-1.asm

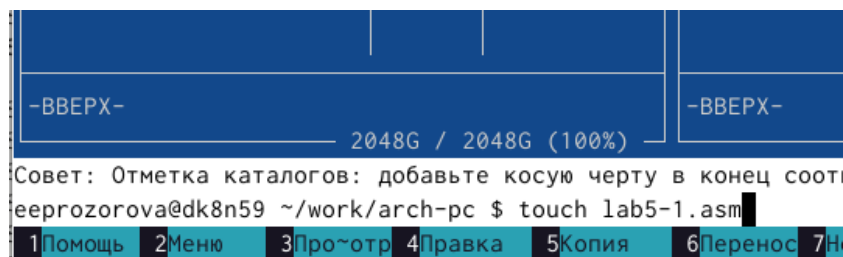


Рис. 2.3: Команда touch

4. С помощью функциональной клавиши F4 я открыла файл ab5-1.asm в редакторе mcedit и ввела текст программы из листинга 5.1.

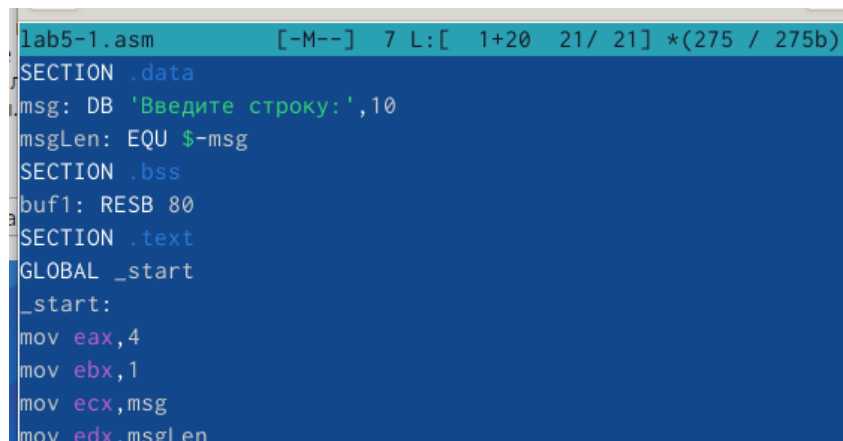


Рис. 2.4: Текст программы в файле ab5-1.asm

5. С помощью функциональной клавиши F3 я проверила содержание текста в файле.

```
mc [eeprozorova@dk8n59.dk.sci.pfu.edu.ru]:~/wo... x
/afs/.dk.sci.pfu.edu.ru/home/e~work/arch-pc/
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
```

Рис. 2.5: Проверка файла

6. Затем я оттранслировала текст программы lab5-1.asm в объектный файл, выполните компоновку объектного файла и запустила получившийся исполняемый файл. На запрос строки ввела свое ФИО.

```
eeprozorova@dk8n59 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.asm
eeprozorova@dk8n59 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o
eeprozorova@dk8n59 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку:
Прозорова Елизавета Евгеньевна
```

Рис. 2.6: Запуск программы, ввод моего ФИО

7. Я скачала файл in_out.asm со страницы курса в ТУИС и переместила его в каталог ~/work/arch-pc с файлом lab5-1.asm.

..	-ВВЕРХ-	ОКТ 24 18:45
in_out.asm	3942	ОКТ 24 18:26
*lab5-1	8744	ноя 8 16:47
lab5-1.asm	282	ноя 8 17:27

Рис. 2.7: Файл in_out.asm в одном каталоге с файлом lab5-1.asm.

- С помощью функциональной клавиши F6 создала копию файла ab5-1.asm с именем lab5-2.asm.

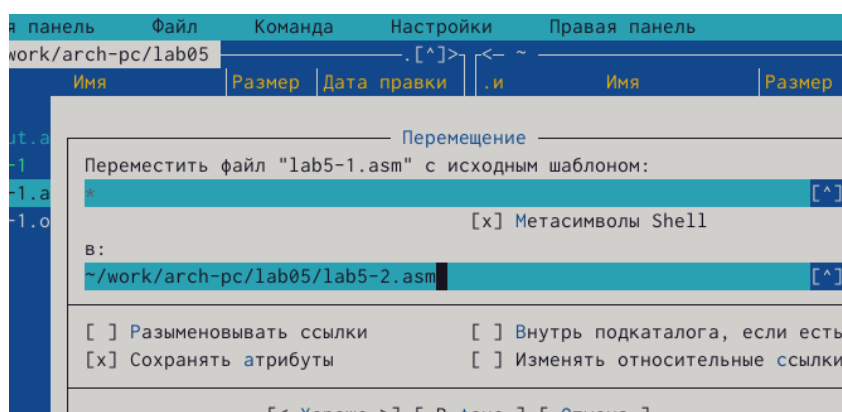


Рис. 2.8: Создание файла lab5-2.asm

- Исправила текста программы в файле с использованием подпрограмм из внешнего файла in_out.asm в соответствии с листингом 5.2.

```

lab5-2.asm.  [-M--]  9 L: [ 1+13  14/ 14] *(180 / 180b) <EOF>  [*][
%include 'in_out.asm'
SECTION .data
msg: DB
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintf
mov ecx, buf1
mov edx, 80
call sread
call quit

```

Рис. 2.9: Текст измененной программы в файле lab5-2.asm

10. Проверка программы из файла lab5-2.asm

```
eeprozorova@dk8n59 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
eeprozorova@dk8n59 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
eeprozorova@dk8n59 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку: Прозорова Елизавета
eeprozorova@dk8n59 ~/work/arch-pc/lab05 $
```

Рис. 2.10: Проверка работы программы

11. Теперь в файле lab5-2.asm заменила подпрограмму sprintLF на sprint. Проверила что изменилось.

```
eeprozorova@dk8n59 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
eeprozorova@dk8n59 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
eeprozorova@dk8n59 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку: Прозорова Елизавета
eeprozorova@dk8n59 ~/work/arch-pc/lab05 $ █
```

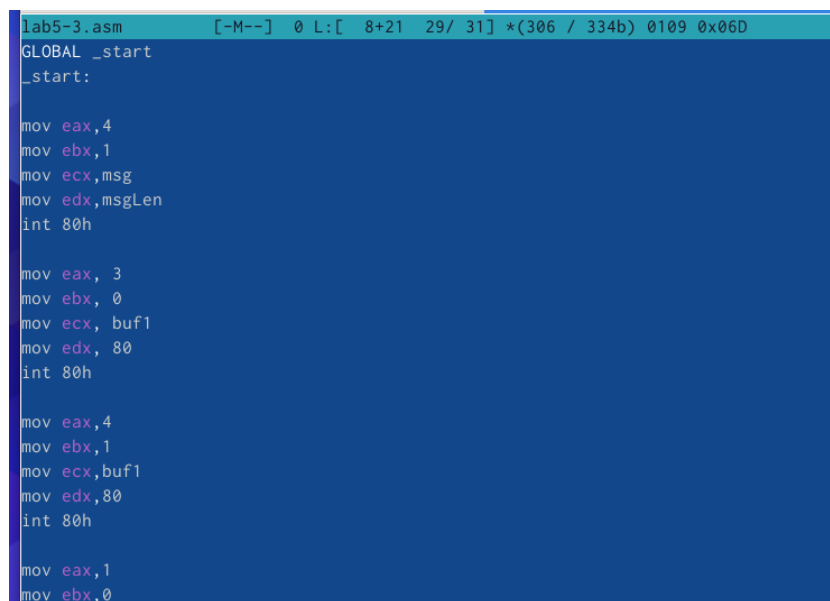
Рис. 2.11: Проверка работы измененной программы

Теперь строка запрашивает ввод на одной строке со строчкой “Введите строку:”.

3 Выполнение самостоятельной работы

1. Создала копию файла lab5-1.asm с именем lab5-3.asm. Изменила текст программы так, работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введенную строку на экран.

Для этого я добавила еще один системный вызов write после read, который выведет введенную строку на экран.



```
lab5-3.asm      [-M--]  0 L:[ 8+21 29/ 31] *(306 / 334b) 0109 0x06D
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
```

Рис. 3.1: Программа в файле lab5-3.asm

2. Проверила исполняемый файл, ввела свою фамилию.

```
eeprozorova@dk8n59 ~/work/arch-pc/lab05 $ nasm -f elf lab5-3.asm
eeprozorova@dk8n59 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-3 lab5-3.o
eeprozorova@dk8n59 ~/work/arch-pc/lab05 $ ./lab5-3
Введите строку:
Прозорова
Прозорова
```

Рис. 3.2: Проверка работы программы lab5-3.asm

3. Создала копию файла lab5-2.asm с именем lab5-4.asm. Исправила текст программы с использованием подпрограмм из внешнего файла in_out.asm, так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введенную строку на экран



```
mc [eeprozorova@dk8n59.dk... x eeprozorova@
lab5-4.asm [---0] 11 L:[ 1+18 19
%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

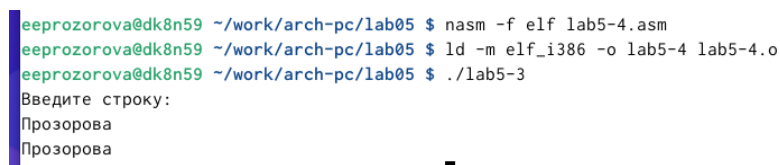
mov eax, buf1

call sread

call quit
```

Рис. 3.3: Текст изменной программы lab5-4.asm

4. Проверила исполняемый файл, ввела свою фамилию.



```
eeprozorova@dk8n59 ~/work/arch-pc/lab05 $ nasm -f elf lab5-4.asm
eeprozorova@dk8n59 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-4 lab5-4.o
eeprozorova@dk8n59 ~/work/arch-pc/lab05 $ ./lab5-3
Введите строку:
Прозорова
Прозорова
```

Рис. 3.4: Проверка работы программы lab5-4.asm

4 Выводы

В ходе выполнения лабораторной работы я приобрела практические навыки работы в Midnight Commander и изучила инструкций языка ассемблера mov и int.