

Лабораторная работа №6

**Основы работы с Midnight Commander. Структура программ на языке
ассемблера NASM. Системные вызовы в ОС GNU Linux**

Приспешкин Андрей Андреевич

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	19
	Список литературы	20

Список иллюстраций

3.1	Открытие Midnight Commander через терминал	7
3.2	Переход в каталог arch-pc	8
3.3	Создание каталога в Midnight Commander	8
3.4	Переход в созданный нами каталог	9
3.5	Создание файла lab5-1.asm	9
3.6	Проверка правильности создания файла	10
3.7	Открытие файла в редакторе mcedit	11
3.8	Код, написанный по образцу	11
3.9	Файл, открытый для просмотра	12
3.10	Трансляция, компоновка и запуск lab5-1.asm	12
3.11	Копирование файла в Midnight Commander	13
3.12	Файл lab5-2.asm в рабочем каталоге	14
3.13	Код, написанный по образцу с использованием подпрограмм из внешнего файла in_out.asm	15
3.14	Работа файла lab5-2.asm	15
3.15	Замена подпрограммы sprintLF на sprint	16
3.16	Результат работы lab5-2.asm после замены sprintLF на sprint . . .	16
3.17	Код в файле lab5-1.asm	17
3.18	Проверка работоспособности кода	17
3.19	Написанный в lab5-2.asm код	18
3.20	Проверка работоспособности lab5-2.asm	18

Список таблиц

1 Цель работы

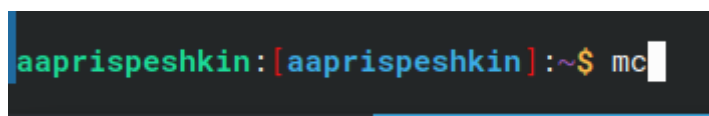
Целью данной лабораторной работы является приобретение навыков работы с программой Midnight Commander и освоение инструкций `mov` и `int` в языке ассемблера NASM.

2 Задание

1. Основы работы в Midnight Commander
2. Структура программ на языке ассемблера NASM
3. Использование внешних файлов в языке ассемблера NASM
- 4.

3 Выполнение лабораторной работы

Откроем Midnight Commander через терминал командой mc(рис.1).

A screenshot of a terminal window with a dark background. The prompt 'aaprispeshkin:[aaprispeshkin]:~\$' is displayed in green and blue text. The command 'mc' is entered in white text, followed by a white cursor. A blue horizontal line is visible at the bottom of the terminal window.

```
aaprispeshkin:[aaprispeshkin]:~$ mc
```

Рис. 3.1: Открытие Midnight Commander через терминал

Перейдём в созданный нами в ходе лабораторной работы №4 каталог arch-рс(рис.2).

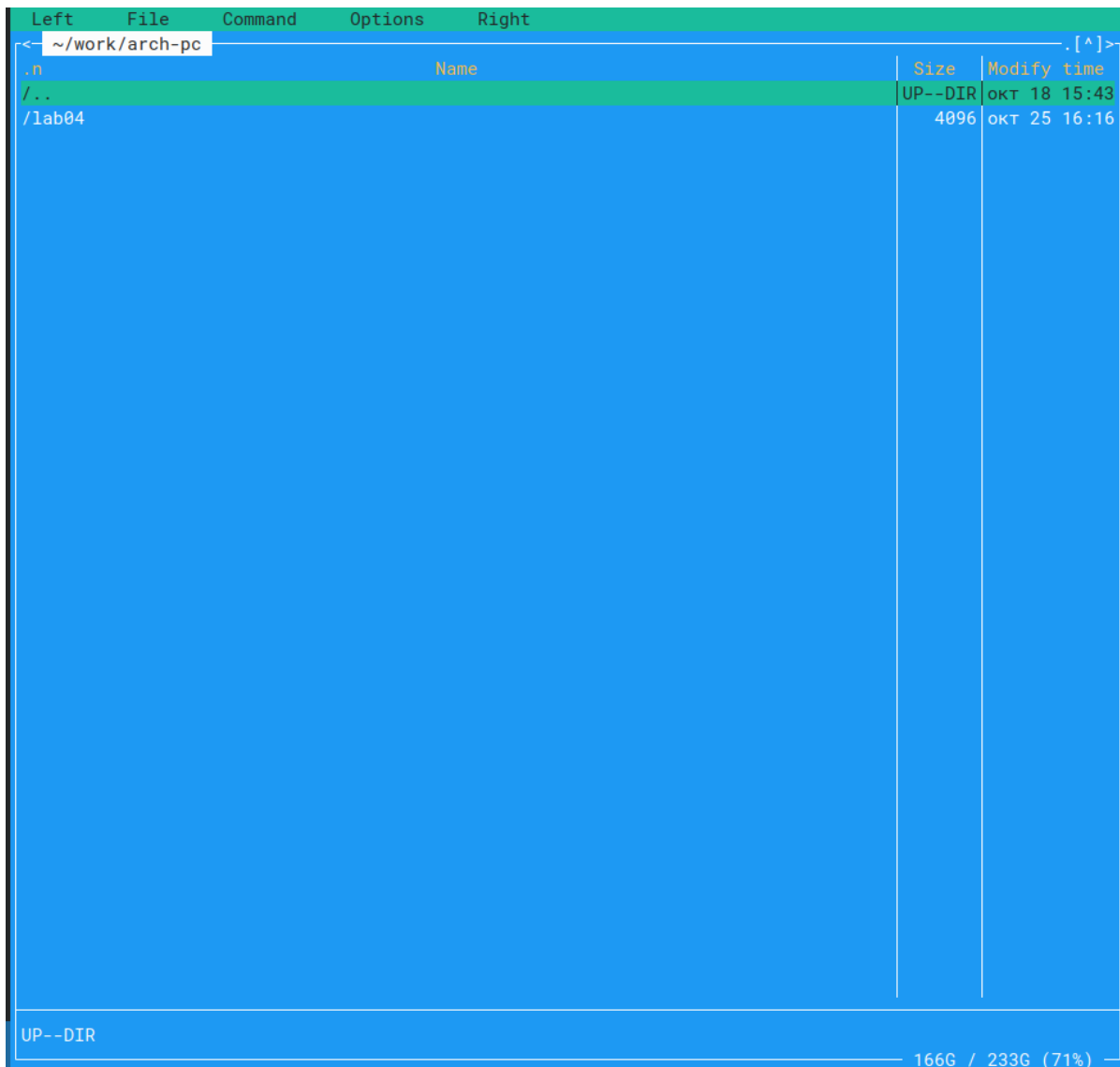


Рис. 3.2: Переход в каталог arch-рс

В Midnight Commander создадим новый каталог lab05(рис.3).

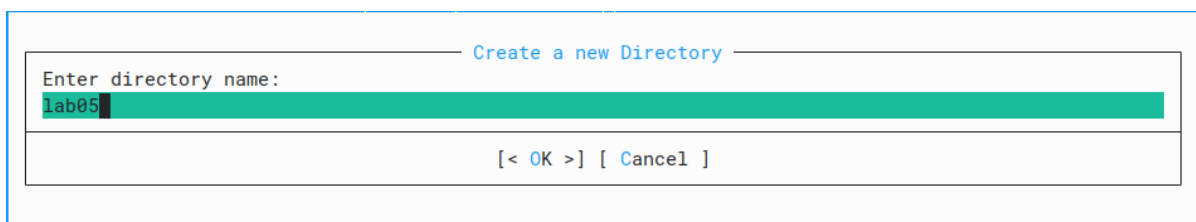


Рис. 3.3: Создание каталога в Midnight Commander

Перейдём в созданный нами каталог(рис.4).

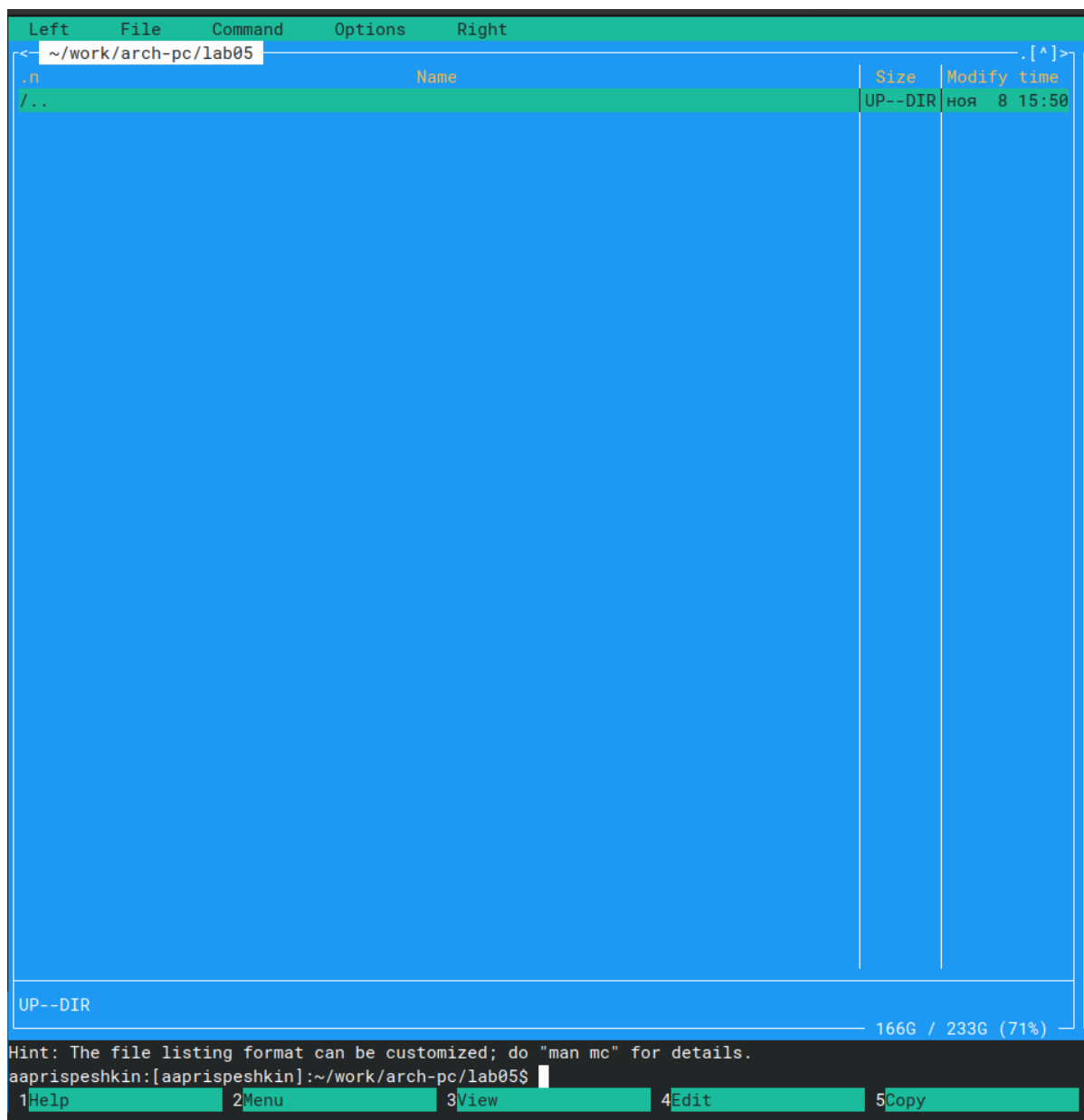


Рис. 3.4: Переход в созданный нами каталог

Утилитой touch создадим файл lab5-1.asm(рис.5).



Рис. 3.5: Создание файла lab5-1.asm

Убедимся что файл действительно создан(рис.6).

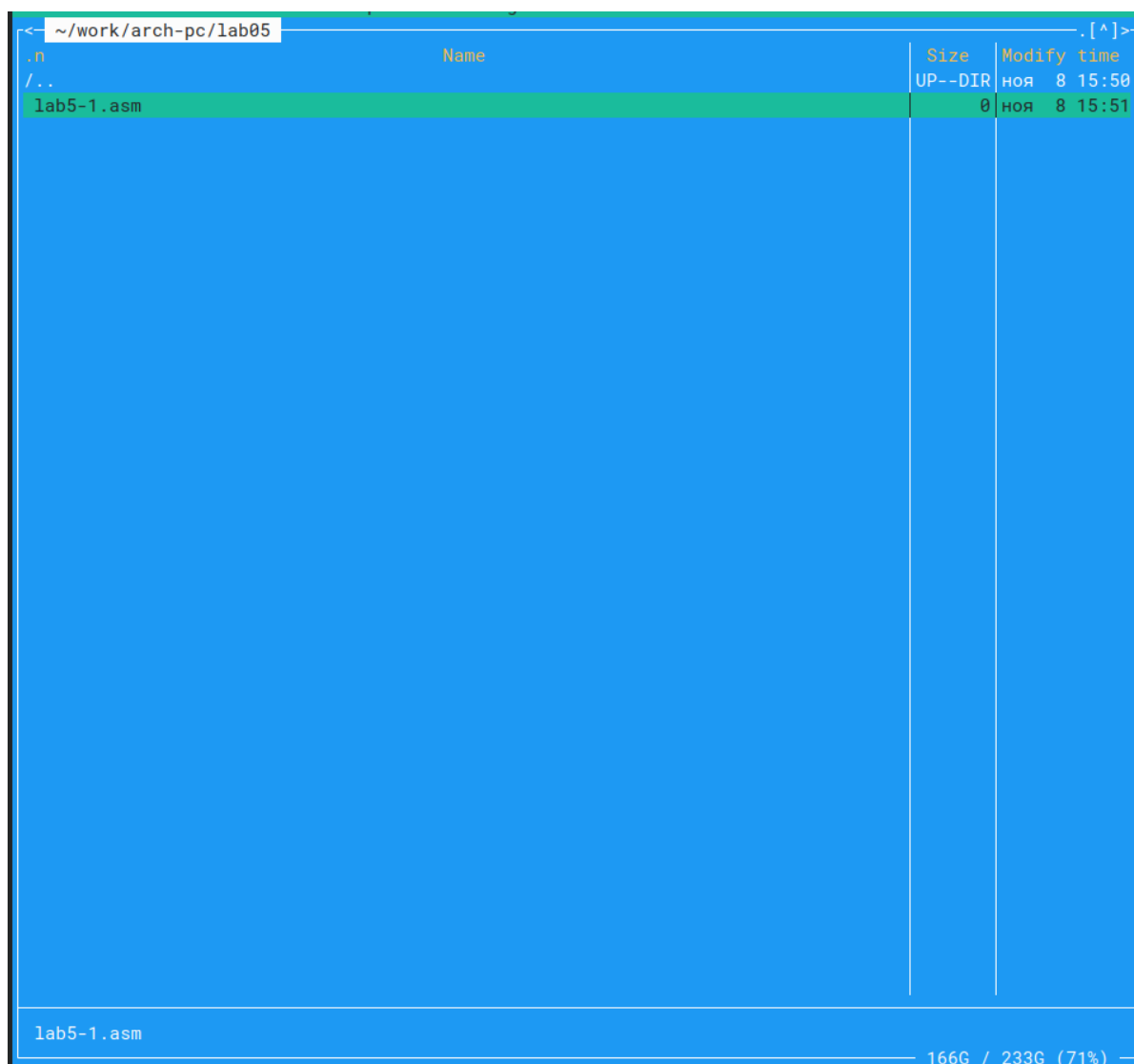


Рис. 3.6: Проверка правильности создания файла

Откроем файл lab5-1.asm во встроенный редакторе mcedit(рис.7).

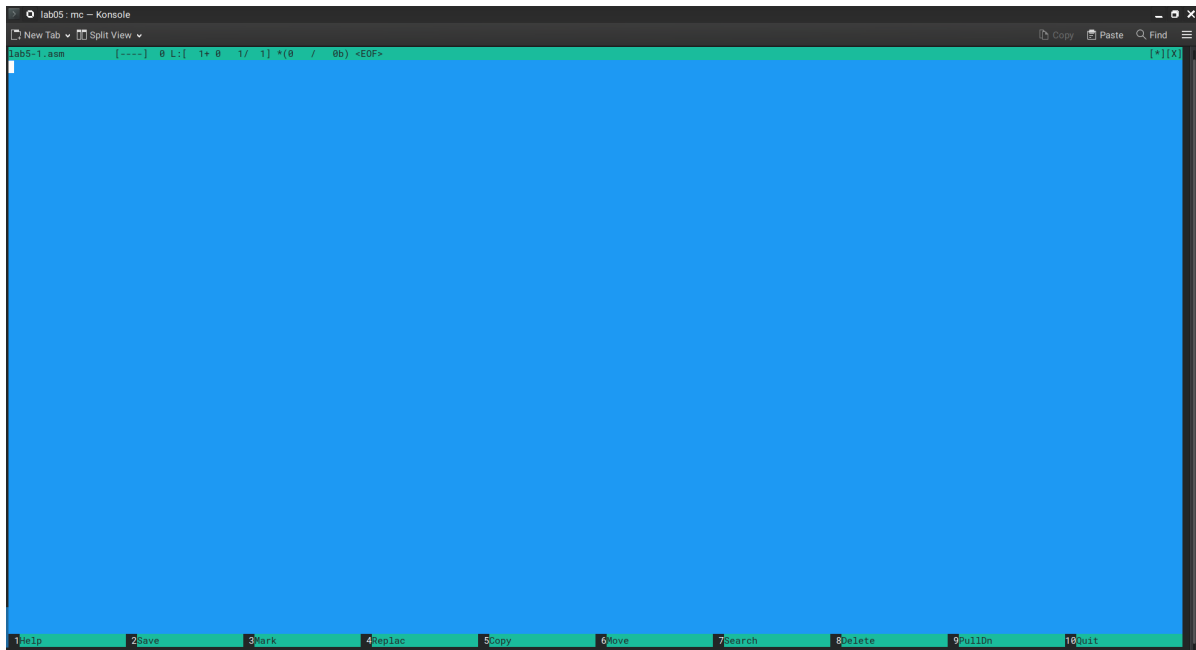


Рис. 3.7: Открытие файла в редакторе mcedit

Напишем код по образцу(рис.8).

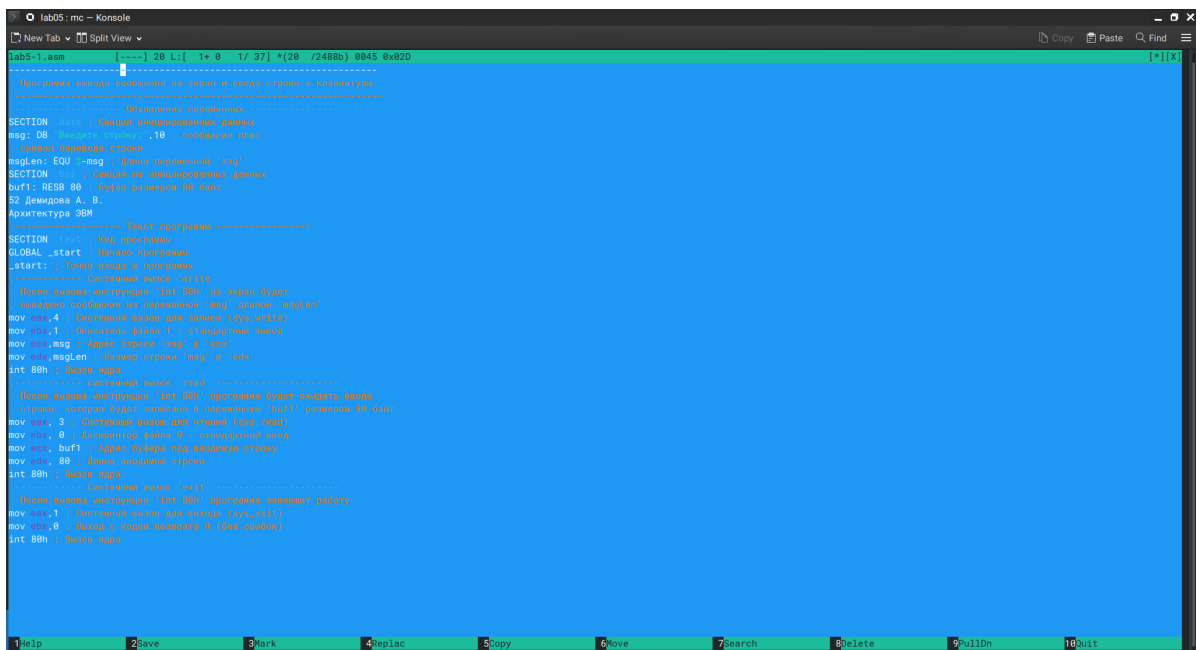


Рис. 3.8: Код, написанный по образцу

В Midnight Commander откроем файл для просмотра и убедимся что правильно написали код(рис.9).

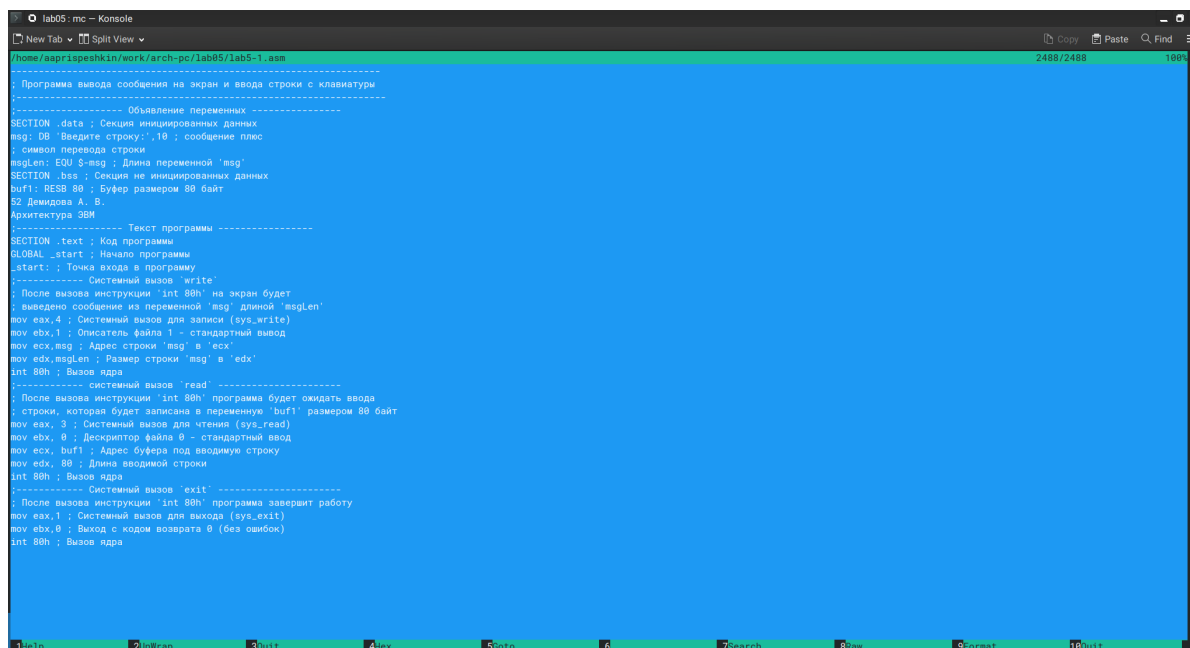


Рис. 3.9: Файл, открытый для просмотра

Оттранслируем и скомпилируем lab5-1.asm, затем запустим полученный файл для проверки его работоспособности(рис.10).

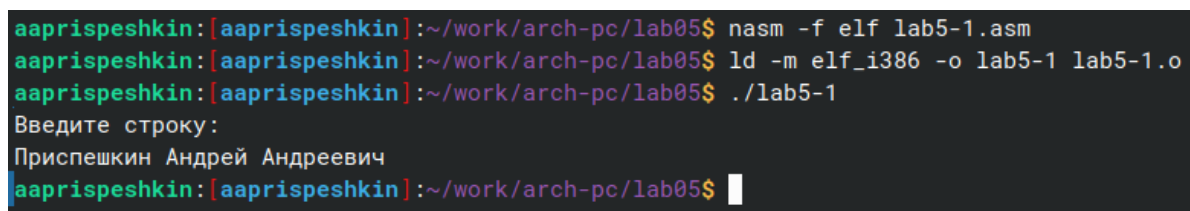


Рис. 3.10: Трансляция, компоновка и запуск lab5-1.asm

Скопируем файл lab5-1.asm в этот же каталог и назовём его lab5-2.asm(рис.11).

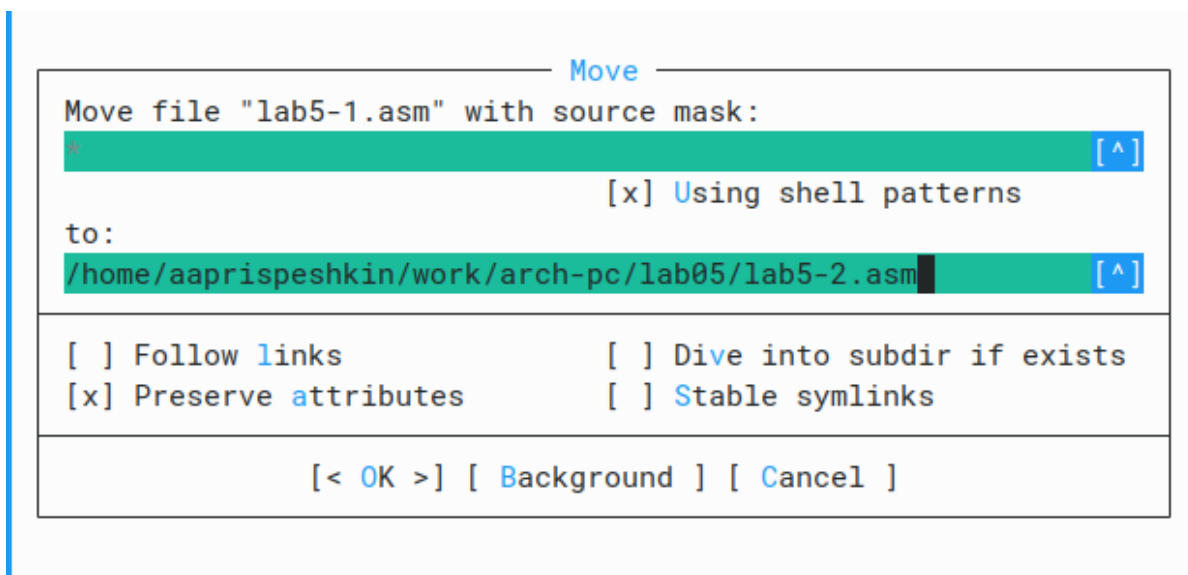


Рис. 3.11: Копирование файла в Midnight Commander

Убедимся в правильности копирования файла(рис.12).

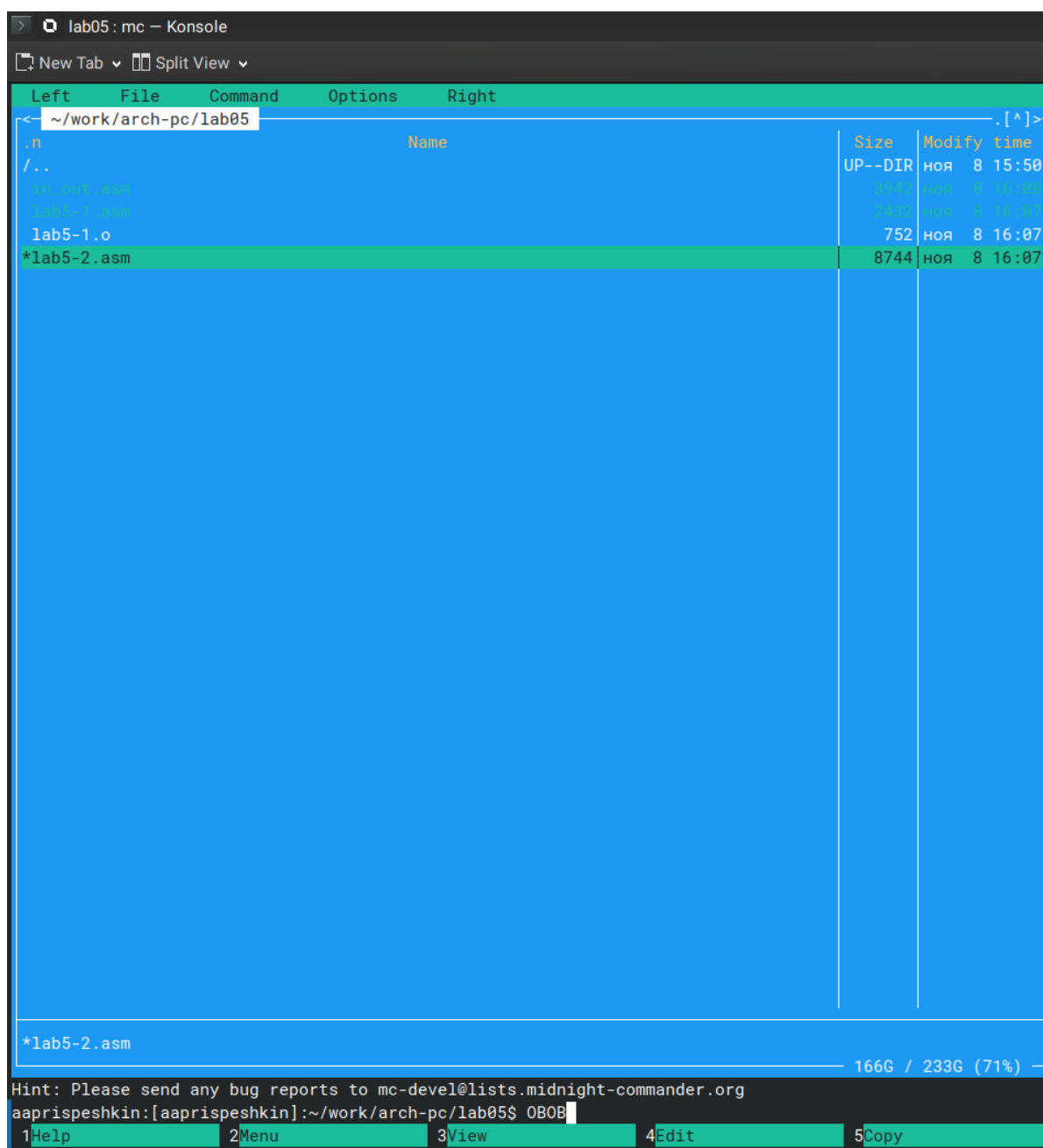


Рис. 3.12: Файл lab5-2.asm в рабочем каталоге

По образцу напишем код, включим в него внешний файл `in_out.asm` и воспользуемся доступными нам подпрограммами(рис.13).


```

lab5-1.asm [----] 3 L: [ 1+14 15/ 22] *(819 /13136) 0832 0x020 [F]IX
SECTION .text : Default code section
msg: DB "Введите строку: ", 10 ; заголовок строки
msgLen: EQU $-msg ; длина переменной 'msg'
SECTION .data : Default data section
buf1: RESB 80 ; буфер размером 80 байт
SECTION .bss : б/с
GLOBAL _start ; Начало программы
_start: ; точка входа в программу
mov eax, 4 ; Системный вызов для вывода (sys_write)
mov ebx, 1 ; Файловый дескриптор 1 - стандартный вывод
mov ecx, msg ; Адрес строки 'msg' в 'ecx'
mov edx, msgLen ; Размер строки 'msg' в 'edx'
int 80h ; Язык ассемблера

mov ebx, 3 ; Системный вызов для чтения (sys_read)
mov ecx, 0 ; Дескриптор файла 0 - стандартный ввод
mov edx, buf1 ; Адрес буфера под вводную строку
mov esi, 80 ; Длина ожидаемой строки
int 80h ; Язык ассемблера

mov eax, 1 ; Системный вызов для выхода (sys_exit)
mov ebx, 0 ; Выход: 0 - нормальное завершение (0 - без ошибок)
int 80h ; Язык ассемблера

```

Рис. 3.17: Код в файле lab5-1.asm

Проверим его работоспособность(рис.18).

```

aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Приспешкин

```

Рис. 3.18: Проверка работоспособности кода

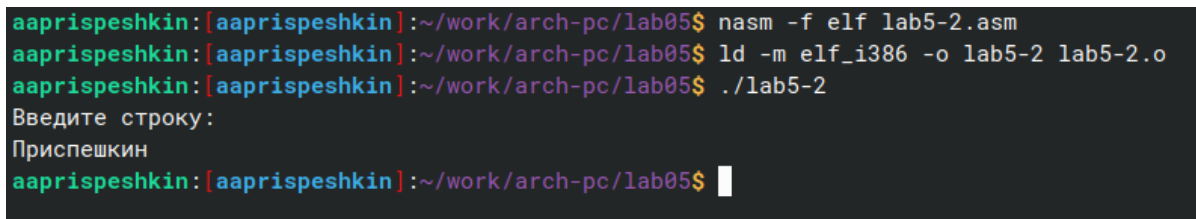
Скопируем файл lab5-2.asm и напишем там код для написания строки, запроса ввода от пользователя и вывода введённой строки на экран, в этот раз пользуясь подпрограммами из in_out.asm(рис.19)



```
lab5-2.asm [M--] 13 L: 1* 1 2/ 181 *(35 / 267b) 8010 0x00A [*][X]
include "lab5.inc"
SECTION ".text"
msg: DB "Prispeshkin", 0h
SECTION ".data"
buf1: RESB 80
SECTION ".bss"
GLOBAL _start
_start:
mov eax, msg
call sprint
mov ecx, buf1
mov edx, 80
call sread
mov ebx, 4
mov ebx, 1
mov ecx, buf1
int 80h
call quit
```

Рис. 3.19: Написанный в lab5-2.asm код

Проверим работоспособность написанного нами кода(рис.20).



```
aapripeshkin:[aapripeshkin]:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
aapripeshkin:[aapripeshkin]:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
aapripeshkin:[aapripeshkin]:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Приспешкин
aapripeshkin:[aapripeshkin]:~/work/arch-pc/lab05$
```

Рис. 3.20: Проверка работоспособности lab5-2.asm

4 Выводы

Я научился работать с программой Midnight Commander, а также освоил инструкции ассемблера `mov` и `int` и научился пользоваться внешними программами.

Список литературы

Лабораторная работа №6