

Отчёт по лабораторной работе №6

Архитектура компьютера

Рогожина Надежда Александровна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
3.1	Основы работы с Midnight Commander	7
3.2	Структура программы на языке ассемблера NASM	8
3.3	Описание инструкции mov	8
3.4	Описание инструкции int	9
4	Выполнение лабораторной работы	10
5	Выполнение заданий для самостоятельной работы	21
6	Выводы	25

Список иллюстраций

4.1	Прежде всего откроем Midnight Commander	10
4.2	Перейдем в каталог, созданный при выполнении лабораторной работы №5	11
4.3	Создадим новый каталог lab06	11
4.4	Проверим создание каталога	12
4.5	Перейдем в этот каталог и создадим файл lab06.asm	12
4.6	Проверим создание файла	13
4.7	С помощью функциональной клавиши F4 откроем файл lab06-1.asm в редакторе nano	13
4.8	Введем текст программы из листинга 6.1	14
4.9	С помощью функциональной клавиши F3 проверим текст программы	14
4.10	Трансляция и компоновка	15
4.11	Скачаем файл	15
4.12	Зайдем в папку, куда файл был скачан через mc	16
4.13	Скопируем файл в каталог с файлом lab06-1.asm с помощью F5	16
4.14	С помощью F6 создадим копию файла lab06-01.asm с именем lab06-2.asm	17
4.15	Проверим создание файла	17
4.16	Исправим текст программы в файле lab06-2.asm	18
4.17	Трансляция и компоновка	18
4.18	Внесем изменения в текст программы	19
4.19	Трансляция и компоновка	19
5.1	Создадим копию исходного файла	21
5.2	Проверим изменение текста программы	22
5.3	Трансляция и компоновка	22
5.4	Создадим копию файла lab06-2.asm	23
5.5	Проверим изменение текста программы	23
5.6	Трансляция и компоновка	24

Список таблиц

3.1	Функциональные клавиши Midnight Commander	7
-----	---	---

1 Цель работы

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

2 Задание

1. Создайте копию файла lab6-1.asm. Внесите изменения в программу (без использования внешнего файла in_out.asm), так чтобы она работала по следующему алгоритму:
 - вывести приглашение типа “Введите строку:”;
 - ввести строку с клавиатуры;
 - вывести введённую строку на экран;
2. Получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите свою фамилию.
3. Создайте копию файла lab6-2.asm. Исправьте текст программы с использование подпрограмм из внешнего файла in_out.asm, так чтобы она работала по следующему алгоритму:
 - вывести приглашение типа “Введите строку:”;
 - ввести строку с клавиатуры;
 - вывести введённую строку на экран.
4. Создайте исполняемый файл и проверьте его работу.

3 Теоретическое введение

3.1 Основы работы с Midnight Commander

Midnight Commander (или просто mc) - это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управлению файловой системой, т.е. mc является файловым менеджером, позволяет сделать работу с файлами более удобной и наглядной.

Таблица 3.1: Функциональные клавиши Midnight Commander

Функциональные клавиши	Выполняемое действие
F1	Вызов контекстно-зависимой подсказки
F2	Вызов меню, созданного пользователем
F3	Просмотр файла, на который указывает подсветка в активной панели
F4	Вызов встроенного редактора для файла, на который указывает подсветка в активной панели
F5	Копирование файла или группы отмеченных файлов из каталога, отображаемого в активной панели, в каталог, отображаемый на второй панели
F6	Перенос файла или группы отмеченных файлов из каталога, отображаемого в активной панели, в каталог отображаемый на второй панели

Функциональ- ные клавиши	Выполняемое действие
F7	Создание подкаталога в каталоге, отображаемом в активной панели
F8	Удаление файла (подкаталога) или группы отмеченных файлов
F9	Вызов основного меню программы
F10	Выход из программы

3.2 Структура программы на языке ассемблера NASM

Программа на языке ассемблера NASM, как правило, состоит из трёх секций:

1. секция кода программы (SECTION .text),
2. секция инициированных (известных во время компиляции) данных (SECTION .data)
3. секция неинициализированных данных (тех, под которые во время компиляции только отводится память, а значение присваивается в ходе выполнения программы) (SECTION .bss)

3.3 Описание инструкции mov

Инструкция языка ассемблера mov предназначена для дублирования данных источника в приёмнике. В общем виде эта инструкция записывается в виде

```
mov dst,src
```

Здесь операнд dst — приёмник, а src — источник.

В качестве операнда могут выступать регистры (register), ячейки памяти (memory) и непосредственные значения (const).

3.4 Описание инструкции `int`

Инструкция языка ассемблера `int` предназначена для вызова прерывания с указанным номером. В общем виде она записывается в виде

`int n`

Здесь `n` — номер прерывания, принадлежащий диапазону 0–255.

При программировании в Linux с использованием вызовов ядра `sys_calls` `n=80h` (принято задавать в шестнадцатеричной системе счисления).

После вызова инструкции `int 80h` выполняется системный вызов какой-либо функции ядра Linux. При этом происходит передача управления ядру операционной системы. Чтобы узнать, какую именно системную функцию нужно выполнить, ядро извлекает номер системного вызова из регистра `eax`. Поэтому перед вызовом прерывания необходимо поместить в этот регистр нужный номер. Кроме того, многим системным функциям требуется передавать какие-либо параметры. По принятым в ОС Linux правилам эти параметры помещаются в порядке следования в остальные регистры процессора: `ebx`, `ecx`, `edx`. Если системная функция должна вернуть значение, то она помещает его в регистр `eax`.

4 Выполнение лабораторной работы

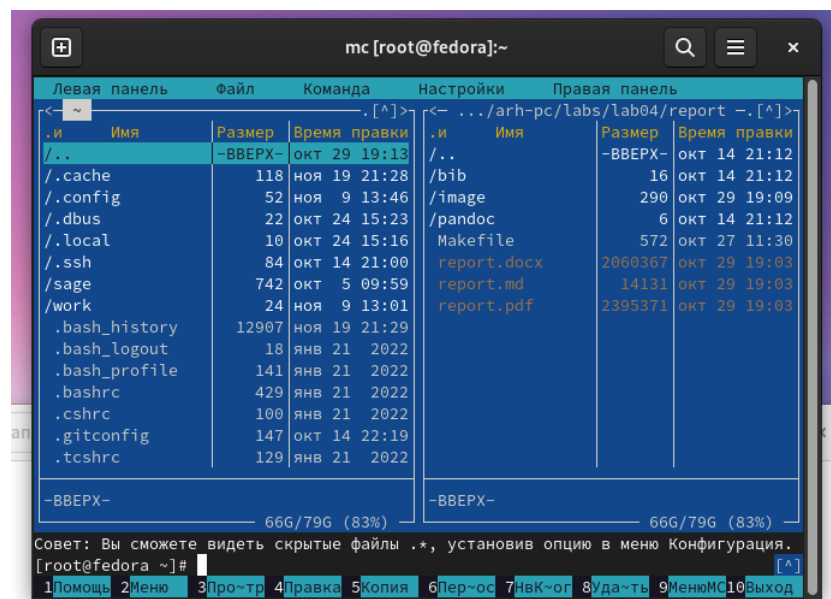


Рис. 4.1: Прежде всего откроем Midnight Commander

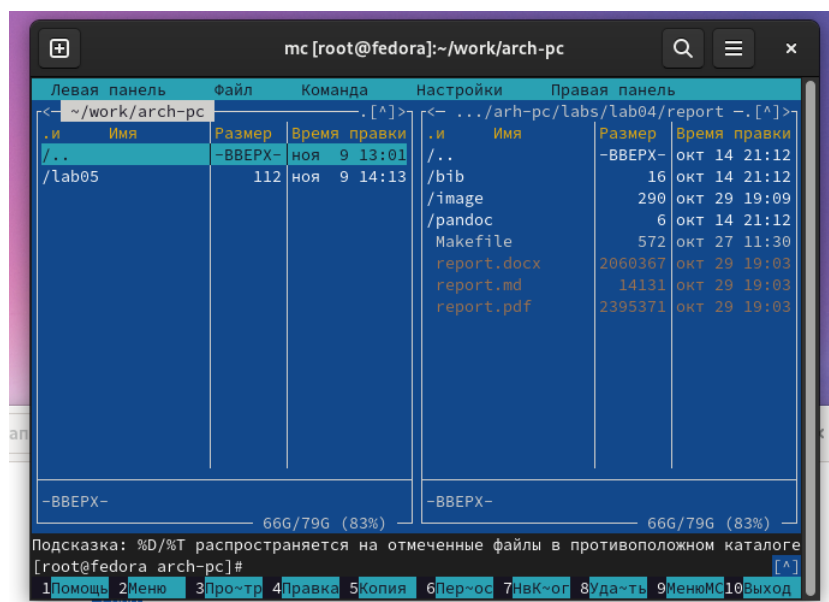


Рис. 4.2: Перейдем в каталог, созданный при выполнении лабораторной работы №5

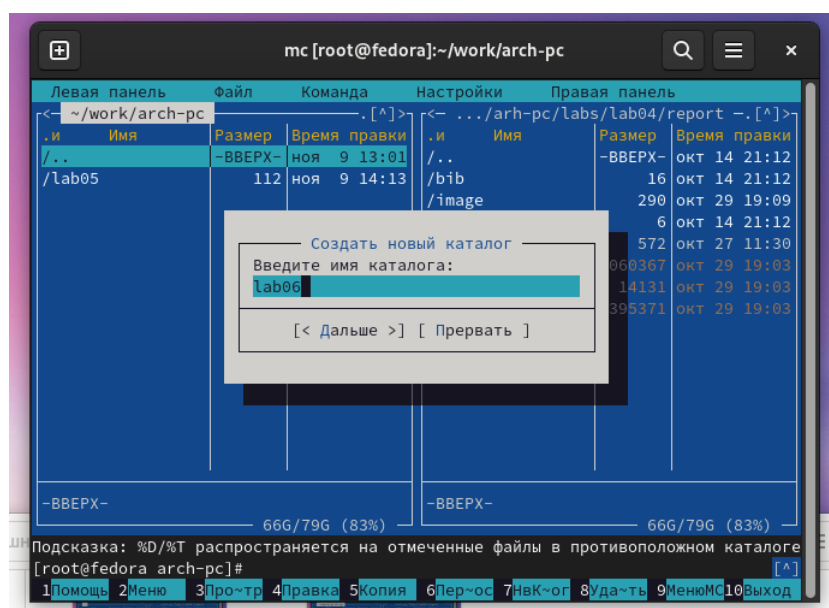


Рис. 4.3: Создадим новый каталог lab06

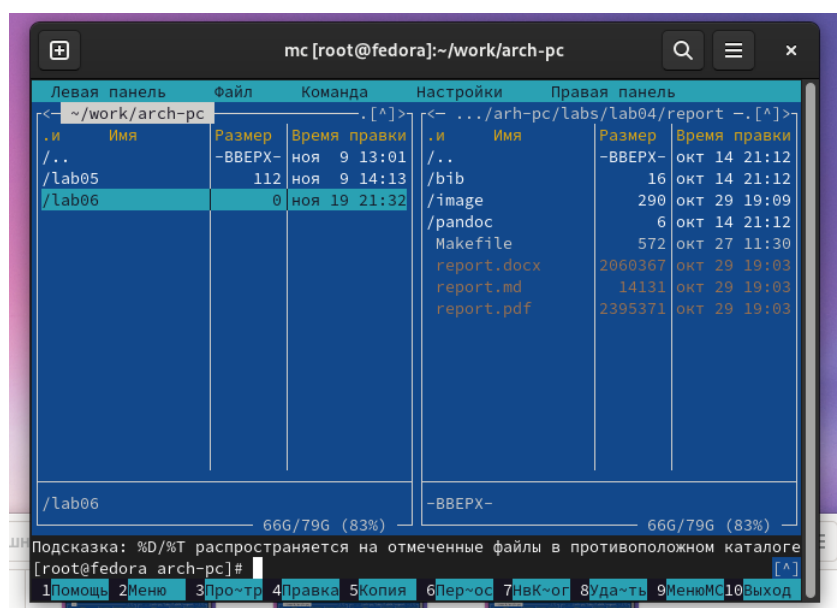


Рис. 4.4: Проверим создание каталога

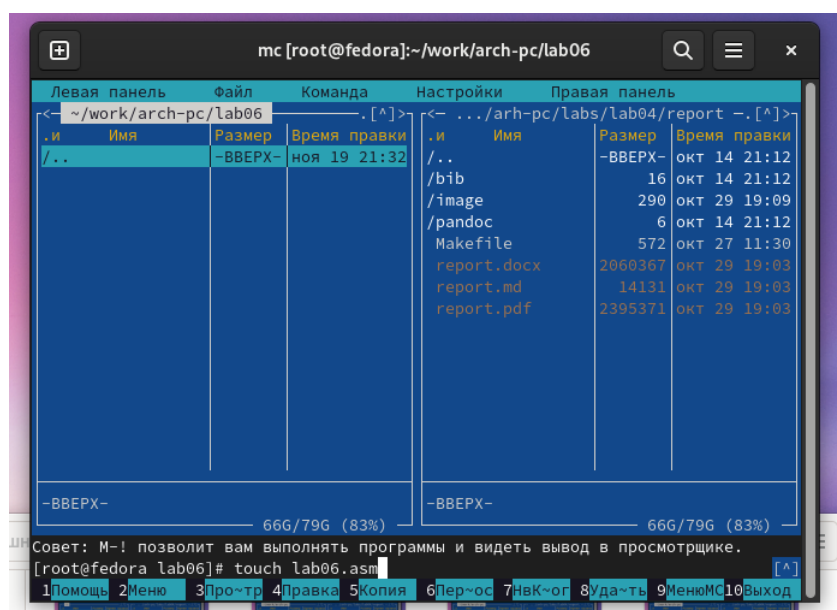


Рис. 4.5: Перейдем в этот каталог и создадим файл lab06.asm

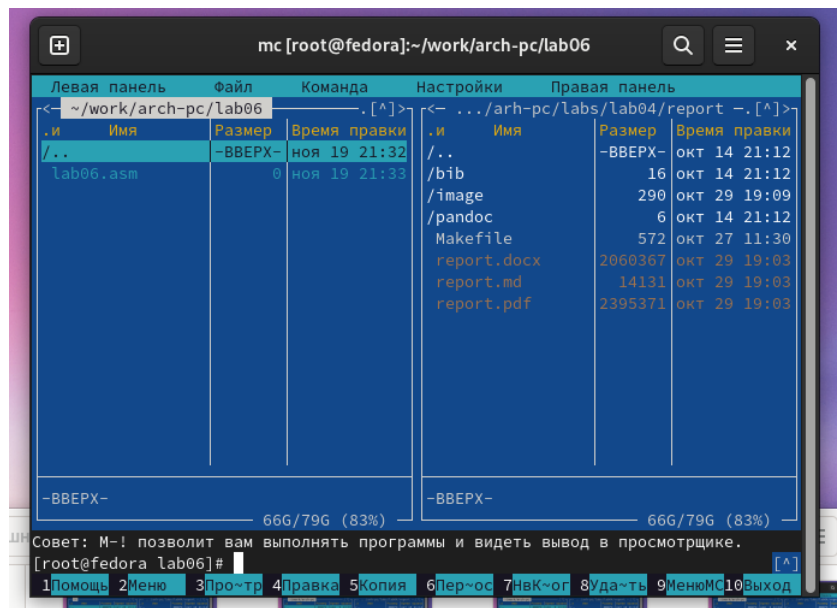


Рис. 4.6: Проверим создание файла

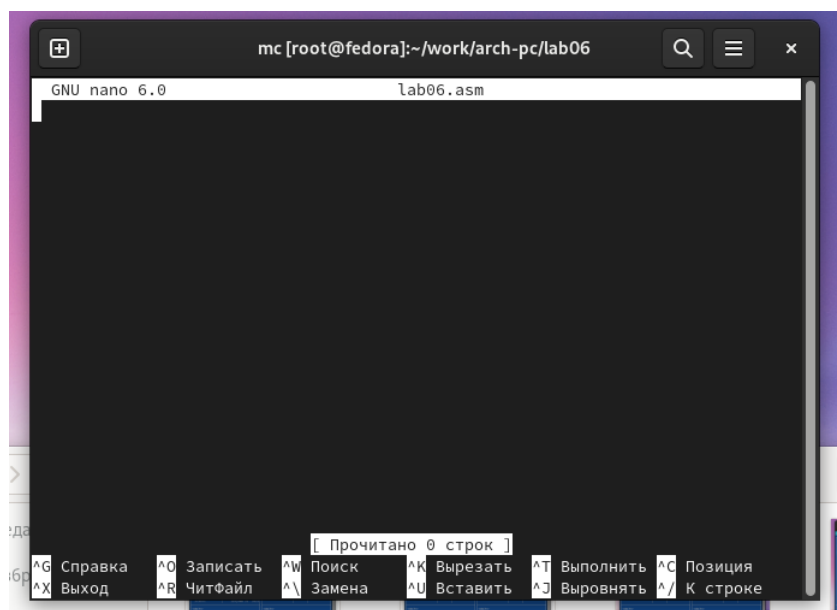


Рис. 4.7: С помощью функциональной клавиши F4 откроем файл lab06-1.asm в редакторе nano

```

mc [root@fedora]:~/work/arch-pc/lab06
GNU nano 6.0 lab06.asm
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;
;-----Объявление переменных-----
SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки

msgLen: EQU $-msg ; Длина переменной 'msg'

SECTION .bss ; Секция не иницированных данных
buf1: RESB 80 ; Буфер размером 80 байт

;----- Текст программы -----
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу

[ Прочитано 47 строк ]
^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R Читфайл ^\ Замена ^U Вставить ^J Вывернуть ^/_ К строке

```

Рис. 4.8: Введем текст программы из листинга 6.1

```

mc [root@fedora]:~/work/arch-pc/lab06
/root/work/arch-pc/lab06/lab06.asm 1111/2487 44%
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;
;-----Объявление переменных-----
SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки

msgLen: EQU $-msg ; Длина переменной 'msg'

SECTION .bss ; Секция не иницированных данных
buf1: RESB 80 ; Буфер размером 80 байт

;----- Текст программы -----
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу

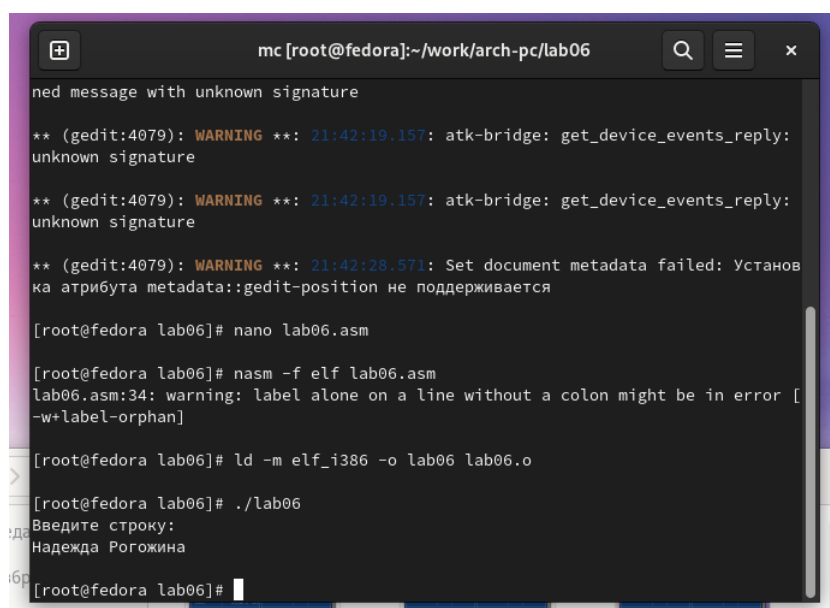
;----- Системный вызов 'write'
; После вызова инструкции 'int 80h' на экран будет

1Помощь 2Раз-рн 3Выход 4Hex 5Пер-ти 6 7Поиск 8Исх-ый 9Формат10Выход

```

Рис. 4.9: С помощью функциональной клавиши F3 проверим текст программы

Оттранслируем текст программы в объектный файл, а затем выполним компоновку и запустим получившийся исполняемый файл



A terminal window titled 'mc [root@fedora]:~/work/arch-pc/lab06' showing the compilation of an assembly file. The output includes warnings from gedit and atk-bridge, followed by the execution of 'nano lab06.asm', 'nasm -f elf lab06.asm' (with a warning about a label), and 'ld -m elf_i386 -o lab06 lab06.o'. The prompt then changes to '[root@fedora lab06]# ./lab06', and the user enters 'Надежда Рогожина'.

```
mc [root@fedora]:~/work/arch-pc/lab06
ned message with unknown signature

** (gedit:4079): WARNING **: 21:42:19.157: atk-bridge: get_device_events_reply:
unknown signature

** (gedit:4079): WARNING **: 21:42:19.157: atk-bridge: get_device_events_reply:
unknown signature

** (gedit:4079): WARNING **: 21:42:28.571: Set document metadata failed: Установ
ка атрибута metadata::gedit-position не поддерживается

[root@fedora lab06]# nano lab06.asm

[root@fedora lab06]# nasm -f elf lab06.asm
lab06.asm:34: warning: label alone on a line without a colon might be in error [
-w+label-orphan]

[root@fedora lab06]# ld -m elf_i386 -o lab06 lab06.o

[root@fedora lab06]# ./lab06
Введите строку:
Надежда Рогожина
[root@fedora lab06]#
```

Рис. 4.10: Трансляция и компоновка

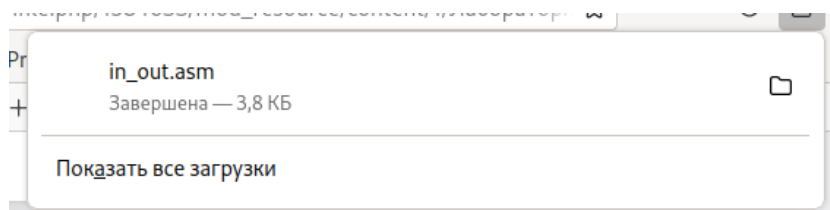


Рис. 4.11: Скачаем файл

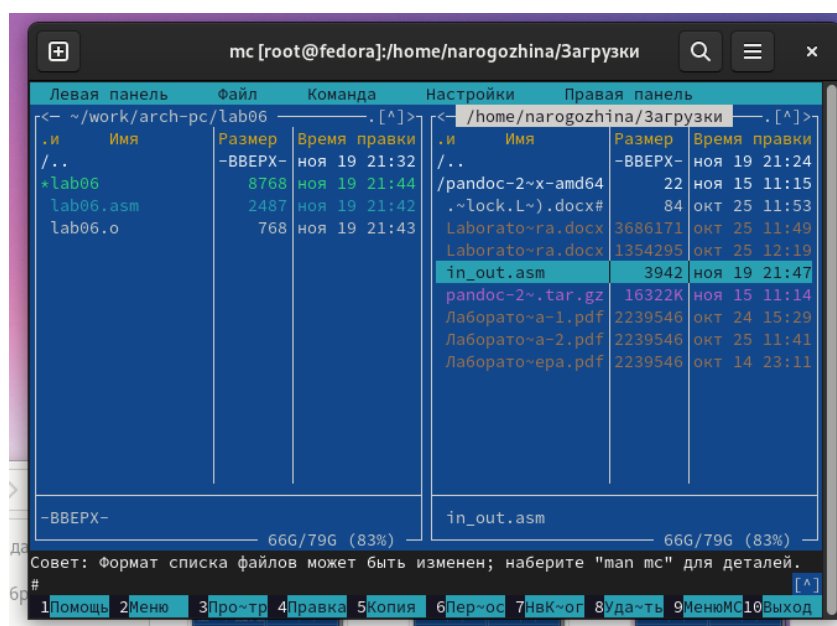


Рис. 4.12: Зайдем в папку, куда файл был скачан через mc

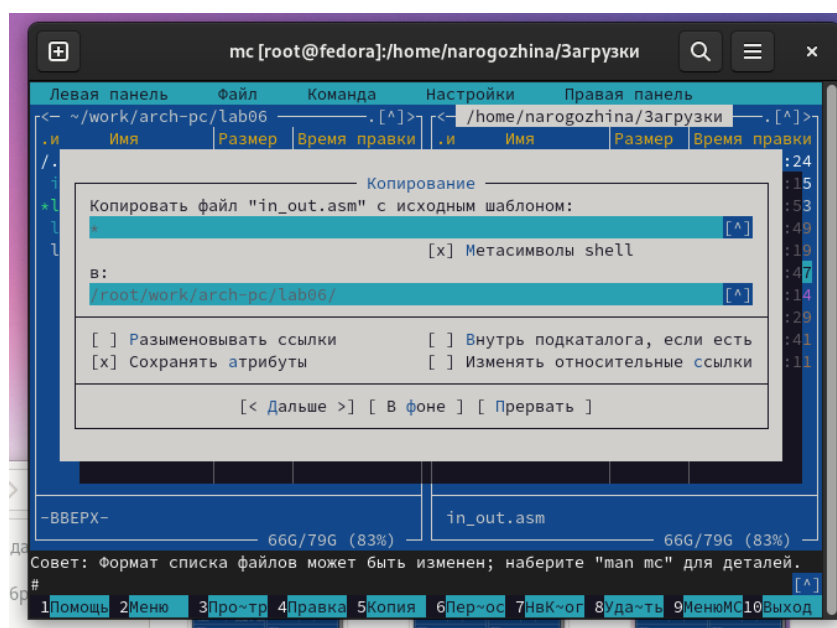


Рис. 4.13: Скопируем файл в каталог с файлов lab06-1.asm с помощью F5

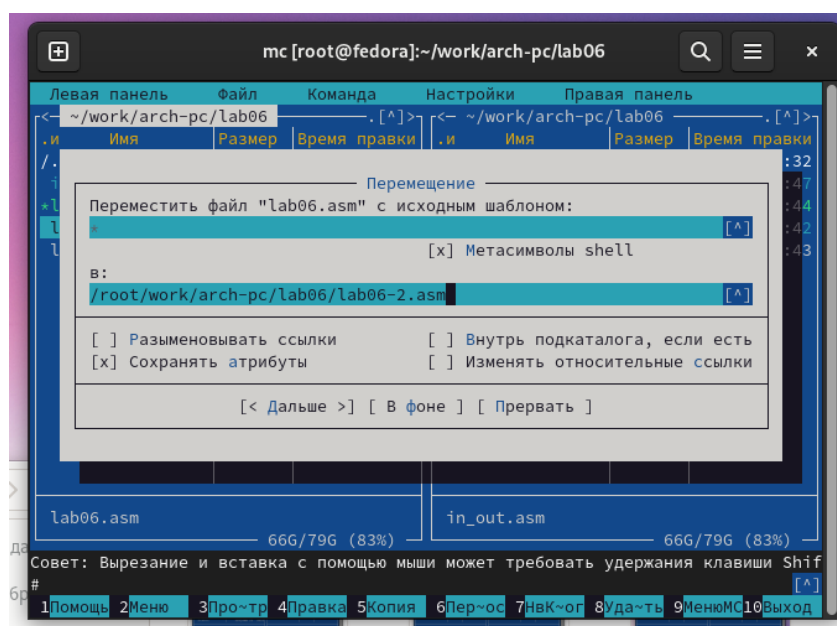
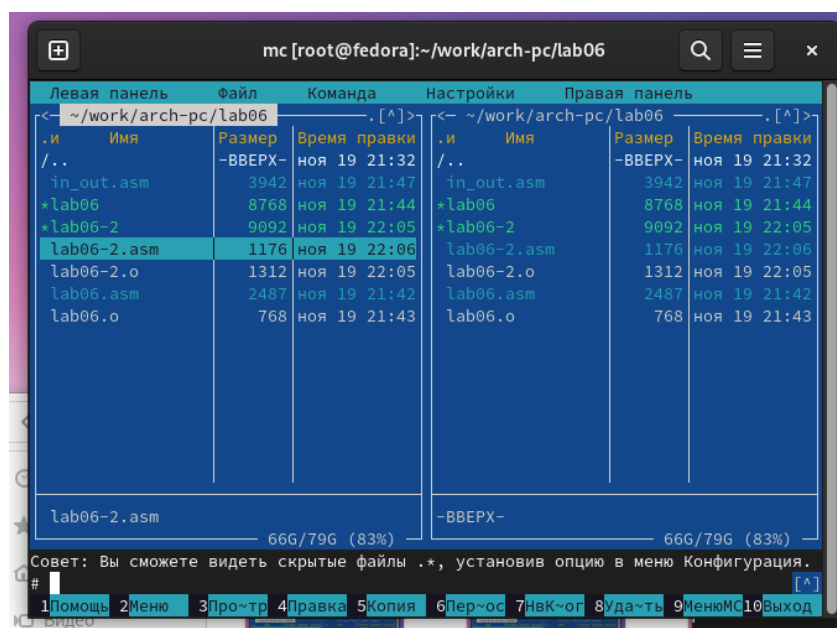
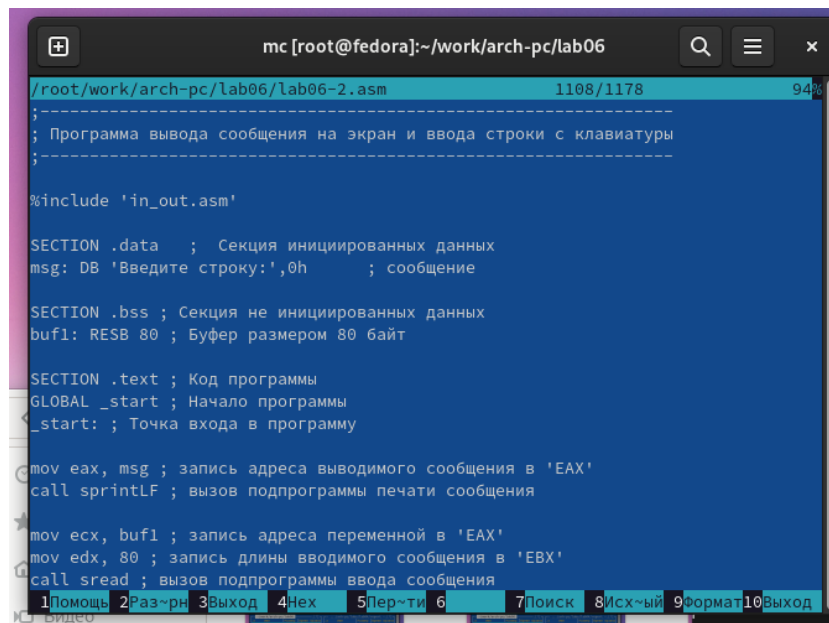


Рис. 4.14: С помощью F6 создадим копию файла lab6-01.asm с именем lab06-2.asm





```
mc [root@fedora]:~/work/arch-pc/lab06
/root/work/arch-pc/lab06/lab06-2.asm 1108/1178 94%
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----

%include 'in_out.asm'

SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку:',0h ; сообщение

SECTION .bss ; Секция не иницированных данных
buf1: RESB 80 ; Буфер размером 80 байт

SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу

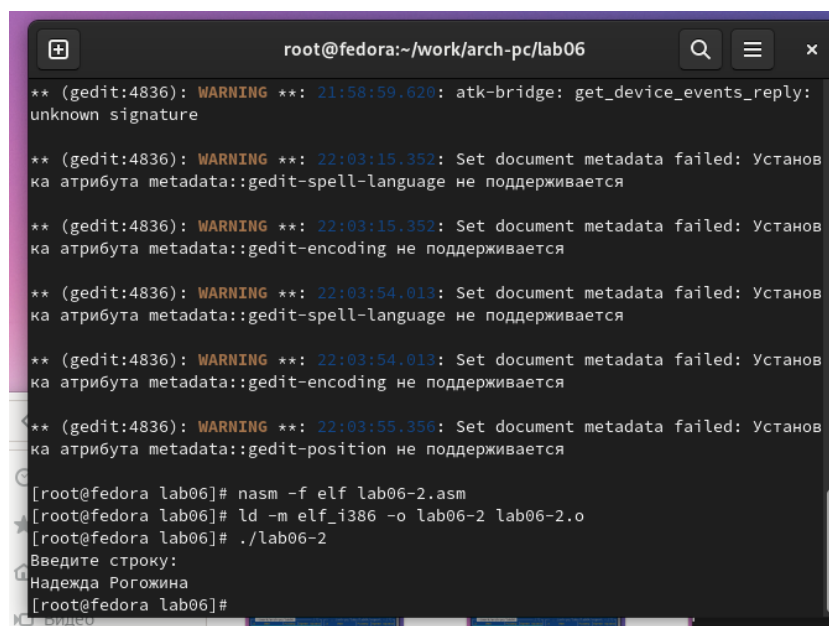
mov eax, msg ; запись адреса выводимого сообщения в 'EAX'
call sprintf ; вызов подпрограммы печати сообщения

mov ecx, buf1 ; запись адреса переменной в 'EAX'
mov edx, 80 ; запись длины вводимого сообщения в 'EBX'
call sread ; вызов подпрограммы ввода сообщения

1Помощь 2Раз~рн 3Выход 4Нех 5Пер~ти 6 7Поиск 8Исх~ый 9Формат10Выход
```

Рис. 4.16: Исправим текст программы в файле lab06-2.asm

Оттранслируем текст программы в объектный файл, а затем выполним компоновку и запустим получившийся исполняемый файл



```
root@fedora:~/work/arch-pc/lab06
** (gedit:4836): WARNING **: 21:58:59.620: atk-bridge: get_device_events_reply:
unknown signature

** (gedit:4836): WARNING **: 22:03:15.352: Set document metadata failed: Установ
ка атрибута metadata::gedit-spell-language не поддерживается

** (gedit:4836): WARNING **: 22:03:15.352: Set document metadata failed: Установ
ка атрибута metadata::gedit-encoding не поддерживается

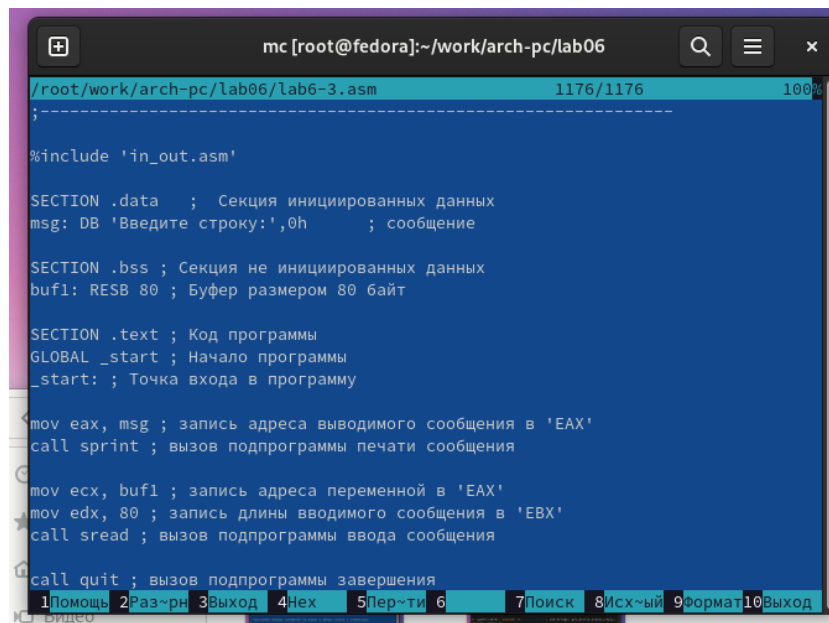
** (gedit:4836): WARNING **: 22:03:54.013: Set document metadata failed: Установ
ка атрибута metadata::gedit-spell-language не поддерживается

** (gedit:4836): WARNING **: 22:03:54.013: Set document metadata failed: Установ
ка атрибута metadata::gedit-encoding не поддерживается

** (gedit:4836): WARNING **: 22:03:55.356: Set document metadata failed: Установ
ка атрибута metadata::gedit-position не поддерживается

[root@fedora lab06]# nasm -f elf lab06-2.asm
[root@fedora lab06]# ld -m elf_i386 -o lab06-2 lab06-2.o
[root@fedora lab06]# ./lab06-2
Введите строку:
Надежда Рогожина
[root@fedora lab06]#
```

Рис. 4.17: Трансляция и компоновка



```
mc [root@fedora]:~/work/arch-pc/lab06
/root/work/arch-pc/lab06/lab6-3.asm 1176/1176 100%
;-----
%include 'in_out.asm'

SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку:',0h ; сообщение

SECTION .bss ; Секция не иницированных данных
buf1: RESB 80 ; Буфер размером 80 байт

SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу

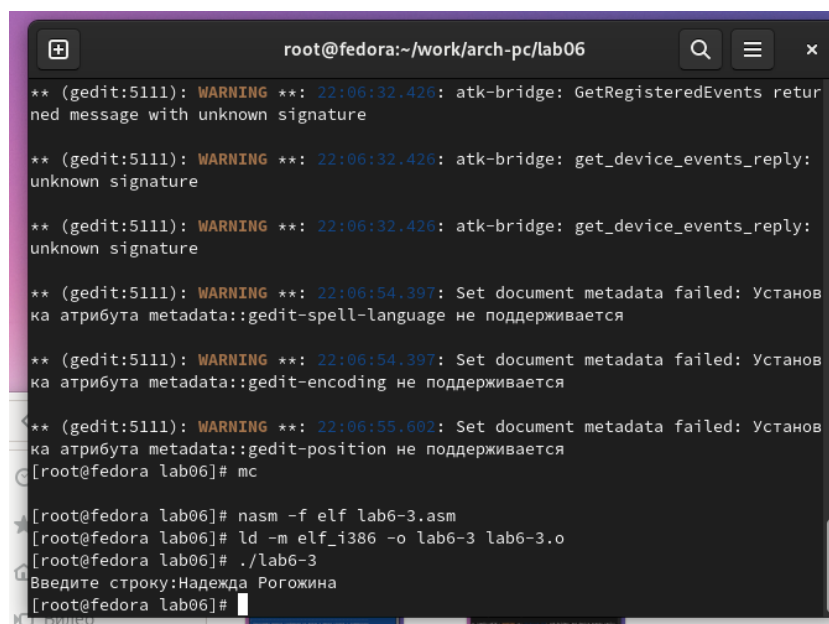
mov eax, msg ; запись адреса выводимого сообщения в 'EAX'
call sprint ; вызов подпрограммы печати сообщения

mov ecx, buf1 ; запись адреса переменной в 'EAX'
mov edx, 80 ; запись длины вводимого сообщения в 'EBX'
call sread ; вызов подпрограммы ввода сообщения

call quit ; вызов подпрограммы завершения
```

Рис. 4.18: Внесем изменения в текст программы

Оттранслируем текст программы в объектный файл, а затем выполним компоновку и запустим получившийся исполняемый файл



```
root@fedora:~/work/arch-pc/lab06
** (gedit:5111): WARNING **: 22:06:32.426: atk-bridge: GetRegisteredEvents returned message with unknown signature
** (gedit:5111): WARNING **: 22:06:32.426: atk-bridge: get_device_events_reply: unknown signature
** (gedit:5111): WARNING **: 22:06:32.426: atk-bridge: get_device_events_reply: unknown signature
** (gedit:5111): WARNING **: 22:06:54.397: Set document metadata failed: Установка атрибута metadata::gedit-spell-language не поддерживается
** (gedit:5111): WARNING **: 22:06:54.397: Set document metadata failed: Установка атрибута metadata::gedit-encoding не поддерживается
** (gedit:5111): WARNING **: 22:06:55.602: Set document metadata failed: Установка атрибута metadata::gedit-position не поддерживается
[root@fedora lab06]# mc

[root@fedora lab06]# nasm -f elf lab6-3.asm
[root@fedora lab06]# ld -m elf_i386 -o lab6-3 lab6-3.o
[root@fedora lab06]# ./lab6-3
Введите строку:Надежда Рогожина
[root@fedora lab06]#
```

Рис. 4.19: Трансляция и компоновка

После изменения текста программы, при выполнении программы больше не

происходит переноса строки.

5 Выполнение заданий для самостоятельной работы

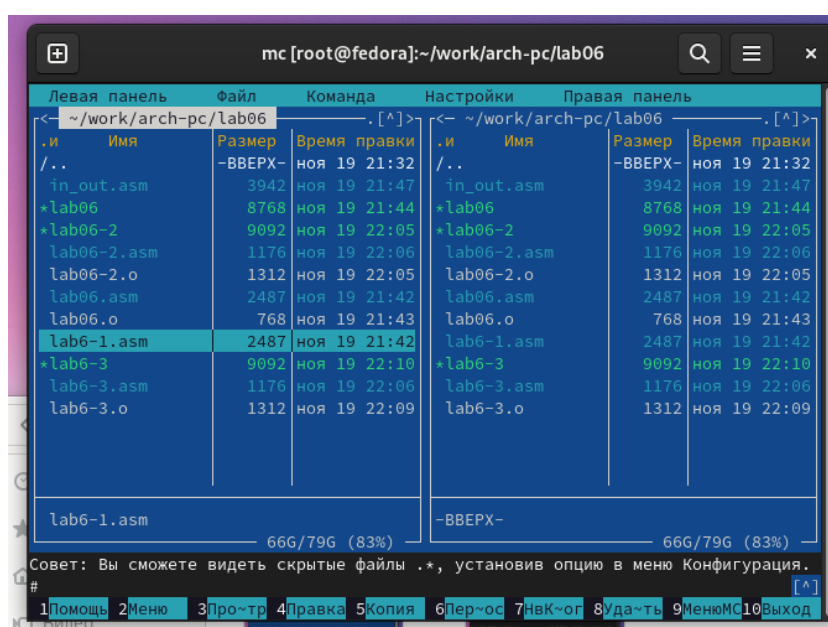
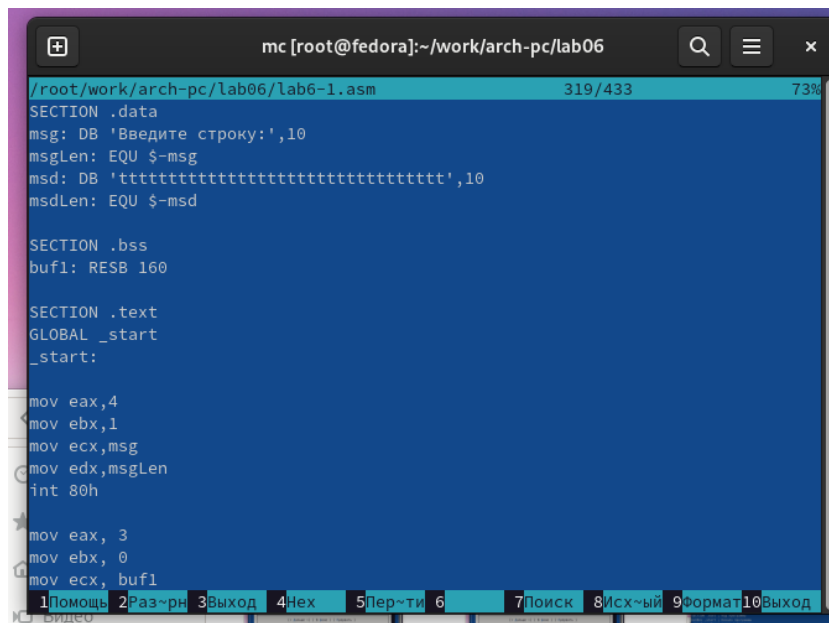


Рис. 5.1: Создадим копию исходного файла



The screenshot shows a text editor window titled 'mc [root@fedora]:~/work/arch-pc/lab06'. The file being edited is '/root/work/arch-pc/lab06/lab6-1.asm' at line 319 of 433. The code defines a data section with a message 'Введите строку:', a buffer 'buf1' of 160 bytes, and a text section starting at '_start'. The assembly instructions include moving values into registers, setting up a system call (int 80h), and preparing arguments for a read system call.

```
/root/work/arch-pc/lab06/lab6-1.asm 319/433 73%
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
msd: DB 'aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa',10
msdLen: EQU $-msd

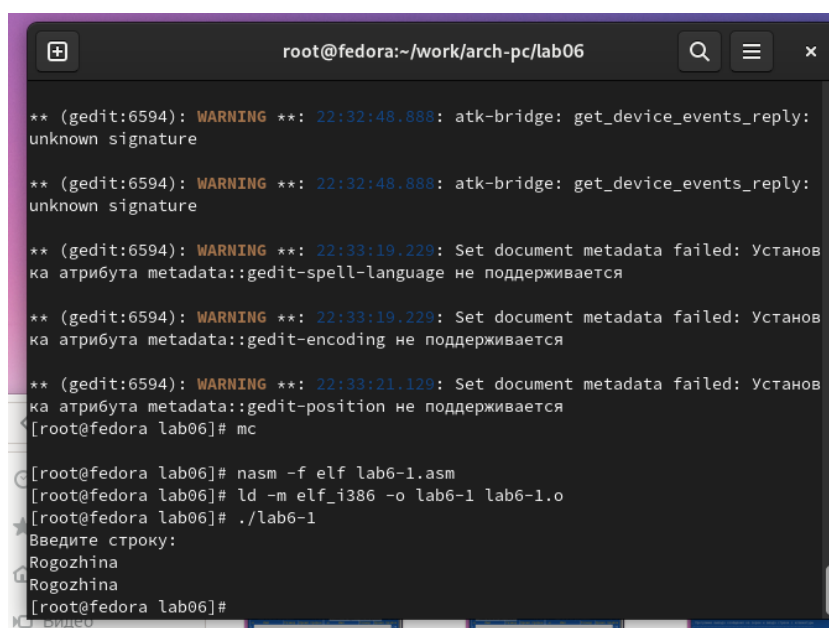
SECTION .bss
buf1: RESB 160

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
```

Рис. 5.2: Проверим изменение текста программы



The screenshot shows a terminal window with the following commands and output:

```
root@fedora:~/work/arch-pc/lab06

** (gedit:6594): WARNING **: 22:32:48.888: atk-bridge: get_device_events_reply:
unknown signature

** (gedit:6594): WARNING **: 22:32:48.888: atk-bridge: get_device_events_reply:
unknown signature

** (gedit:6594): WARNING **: 22:33:19.229: Set document metadata failed: Установ
ка атрибута metadata::gedit-spell-language не поддерживается

** (gedit:6594): WARNING **: 22:33:19.229: Set document metadata failed: Установ
ка атрибута metadata::gedit-encoding не поддерживается

** (gedit:6594): WARNING **: 22:33:21.129: Set document metadata failed: Установ
ка атрибута metadata::gedit-position не поддерживается
[root@fedora lab06]# mc

[root@fedora lab06]# nasm -f elf lab6-1.asm
[root@fedora lab06]# ld -m elf_i386 -o lab6-1 lab6-1.o
[root@fedora lab06]# ./lab6-1
Введите строку:
Rogozhina
Rogozhina
[root@fedora lab06]#
```

Рис. 5.3: Трансляция и компоновка

Таким образом, после запуска программы, мы удостоверились, что все работает как нужно

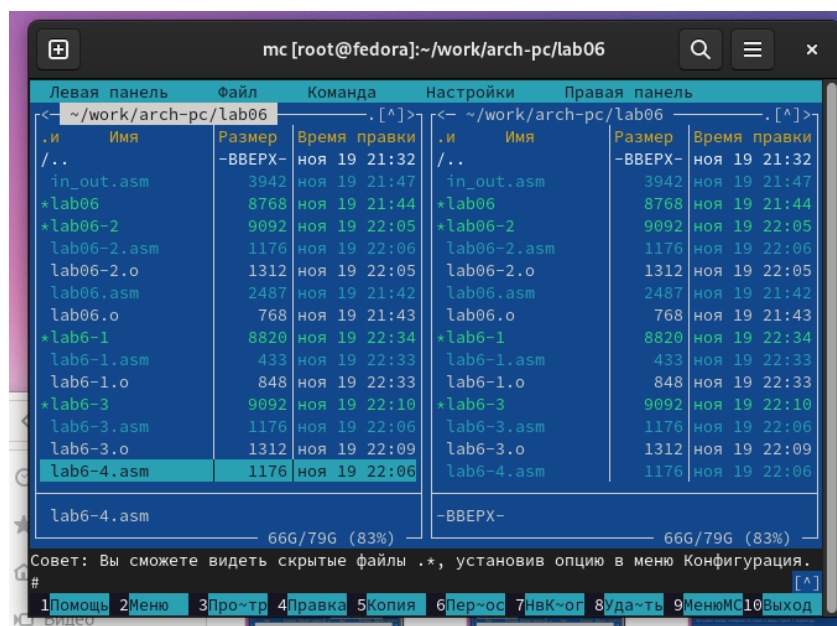


Рис. 5.4: Создадим копию файла lab06-2.asm

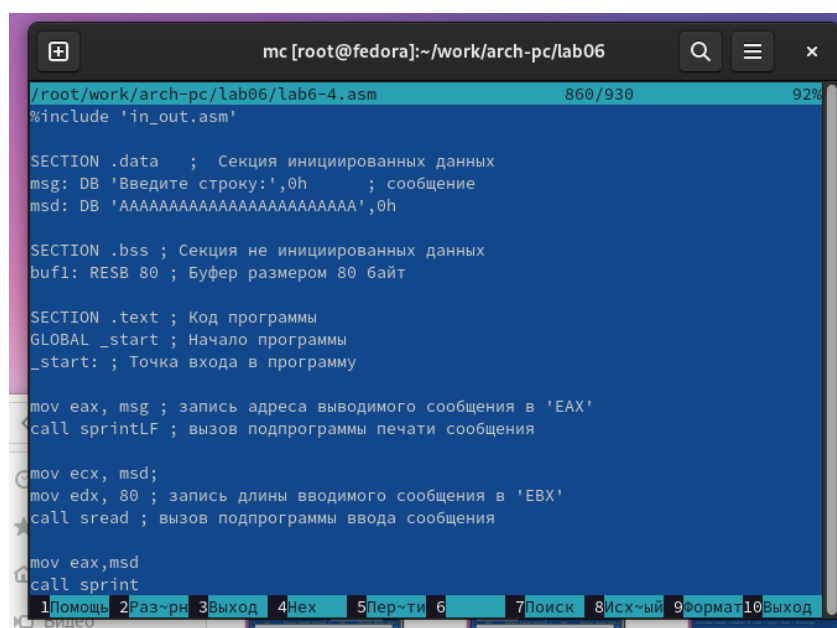
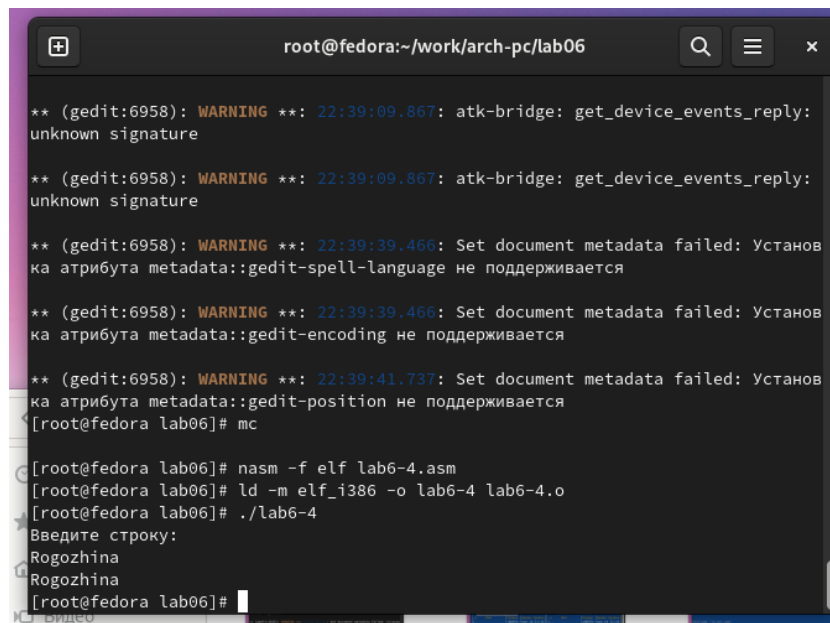


Рис. 5.5: Проверим изменение текста программы



```
root@fedora:~/work/arch-pc/lab06

** (gedit:6958): WARNING **: 22:39:09.867: atk-bridge: get_device_events_reply:
unknown signature

** (gedit:6958): WARNING **: 22:39:09.867: atk-bridge: get_device_events_reply:
unknown signature

** (gedit:6958): WARNING **: 22:39:39.466: Set document metadata failed: Установ
ка атрибута metadata::gedit-spell-language не поддерживается

** (gedit:6958): WARNING **: 22:39:39.466: Set document metadata failed: Установ
ка атрибута metadata::gedit-encoding не поддерживается

** (gedit:6958): WARNING **: 22:39:41.737: Set document metadata failed: Установ
ка атрибута metadata::gedit-position не поддерживается
[root@fedora lab06]# mc

[root@fedora lab06]# nasm -f elf lab6-4.asm
[root@fedora lab06]# ld -m elf_i386 -o lab6-4 lab6-4.o
[root@fedora lab06]# ./lab6-4
Введите строку:
Rogozhina
Rogozhina
[root@fedora lab06]#
```

Рис. 5.6: Трансляция и компоновка

После запуска программы мы убедились, что все работает

6 Выводы

Таким образом, по выполнению лабораторной работы мы научились работать с Midnight Commander.

1. МС - это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управлению файловой системы. Она упрощает работу с файлами, делает ее более удобной и наглядной.
2. Такие операции как (например) копирование, перемещение, удаление файлов можно выполнить и с помощью bash, и с помощью mc.
3. Типичный формат записи команд NASM имеет вид:

[метка:] мнемокод [операнд {, операнд}] [; комментарий]

4. Секция bss содержит переменные, для которых **не задано** начальное значение. Секция data содержит переменные, для которых **задано** начальное значение.
5. Для объявления инициализированных данных в секции .data используются следующие директивы:
 - DB - определяет переменную размером в 1 байт
 - DW - определяет переменную размером в 2 байта (слово)
 - DD - определяет переменную размером в 4 байта (двойное слово)
 - DQ - определяет переменную размером в 8 байт (четверное слово)

- DT - определяет переменную размером в 10 байт
6. Инструкция **mov eax, esi** пересылает значение регистра esi в регистр eax.
 7. int 80 - вызов ядра.