

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5**

дисциплина: Архитектура вычислительных систем

Студент: Уханаева Сансара Зоригтуевна

Студ. билет № 1132229047

Группа: НБИбд-03-22

**МОСКВА**

2022 г.

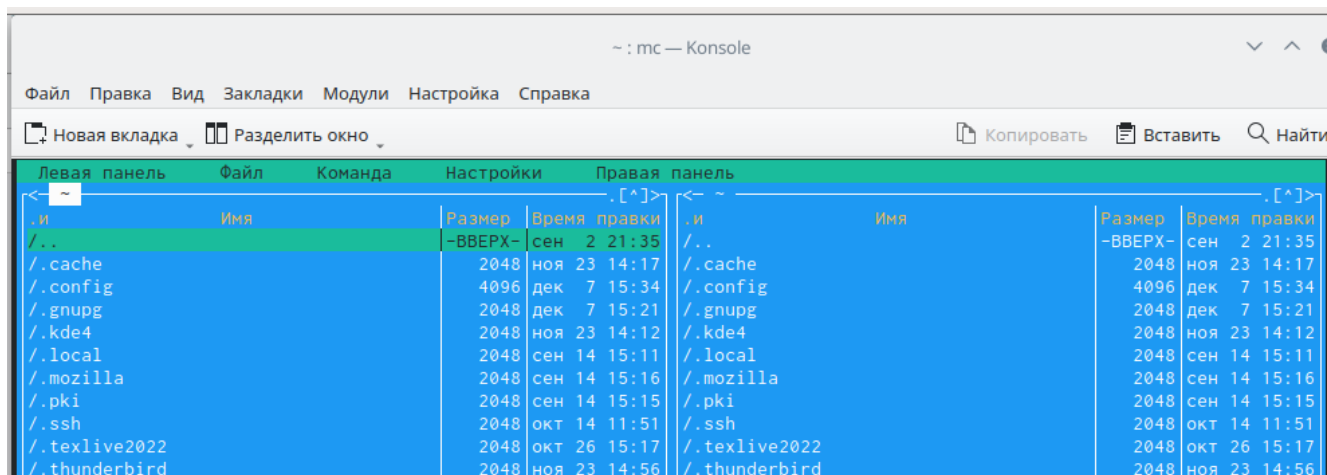
## Цель работы:

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

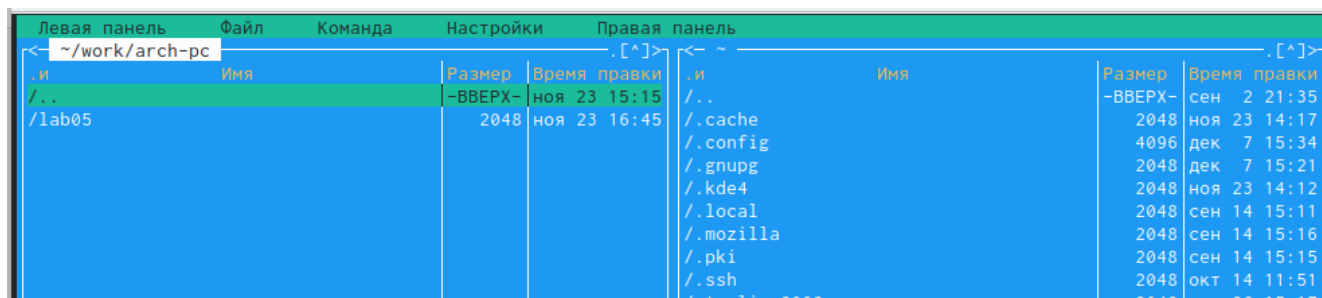
## Выполнение лабораторной работы:

### 1. Откроем Midnight Commander

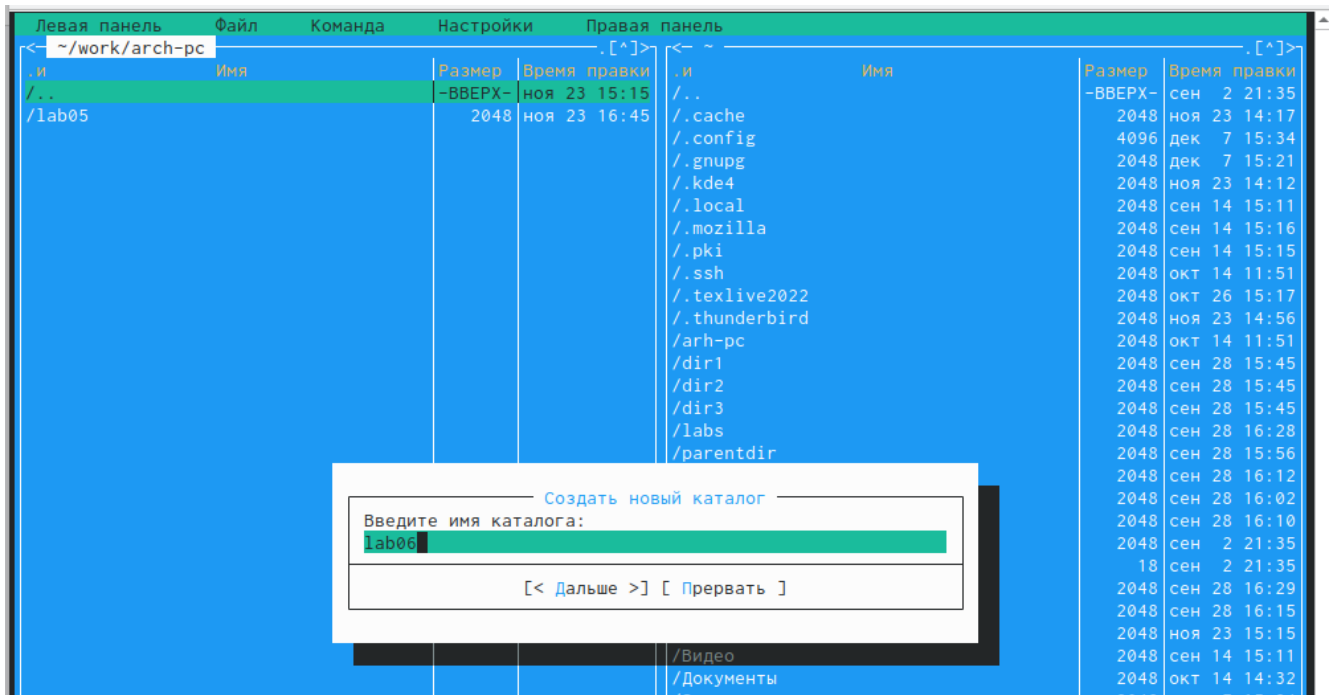
```
user@dk4n31:~$ mc
```



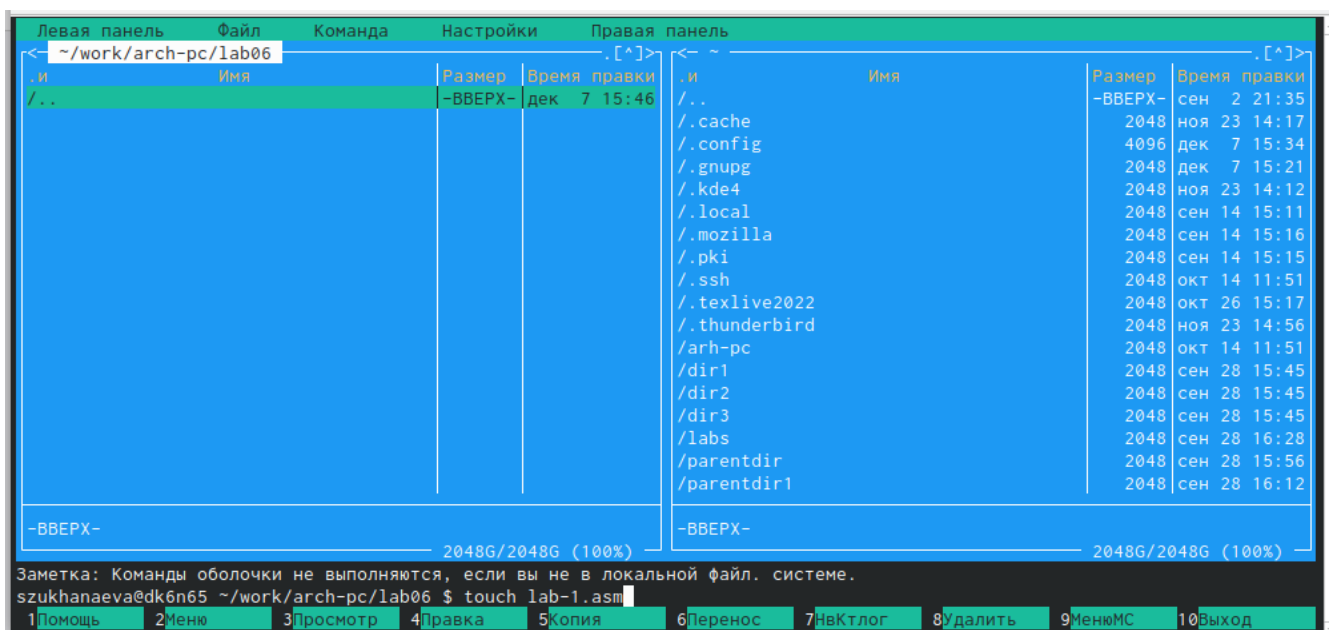
2. Пользуясь клавишами ↑, ↓ и Enter перейдем в каталог ~/work/archpc созданный при выполнении лабораторной работы №5

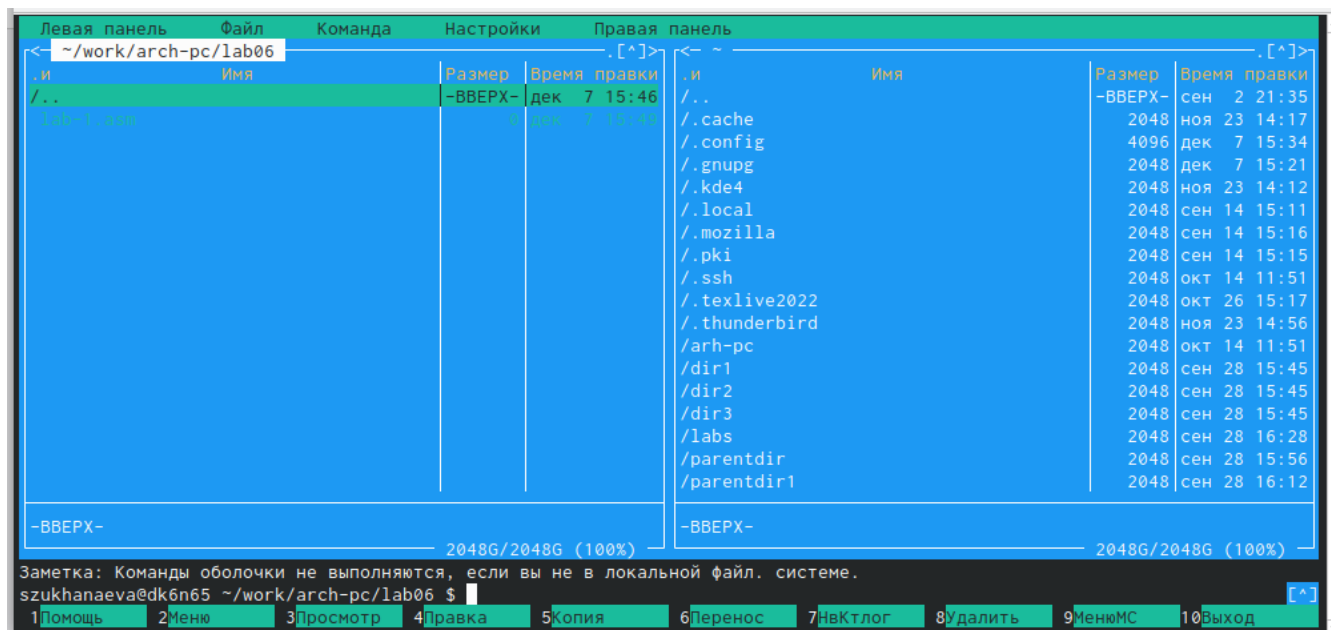


3. С помощью функциональной клавиши F7 создадим папку lab06 и перейдем в созданный каталог.

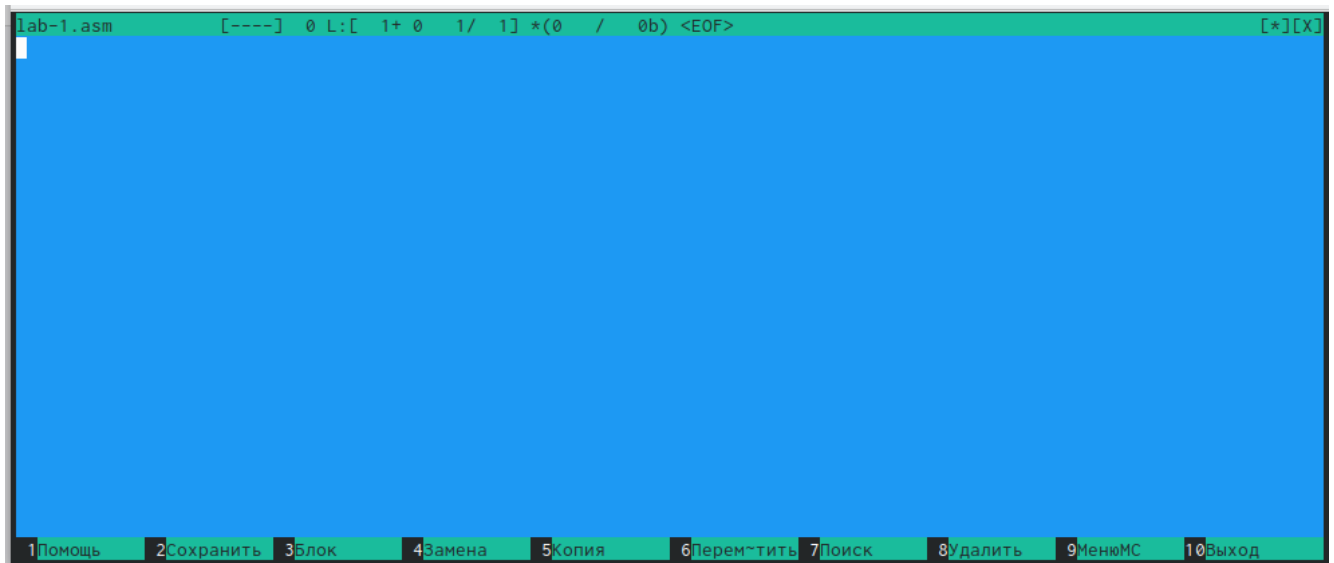


4. Пользуясь строкой ввода и командой touch создадим файл lab6-1.asm





5. С помощью функциональной клавиши F4 откроем файл `lab6-1.asm` для редактирования во встроенном редакторе. Как правило в качестве встроенного редактора Midnight Commander используется редакторы `nano` или `mcedit`



6. Введем текст программы из листинга 6.1 (можно без комментариев), сохраните изменения и закройте файл.

```
lab-1.asm [-M--] 22 L: [ 1+ 8 9/ 45] *(597 /2700b) 0032 0x020 [*][X]
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
;----- Объявление переменных -----
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'

SECTION .bss ; Секция не инициализированных 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
; выведено сообщение из переменной 'msg' длиной 'msgLen'

mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла 1 - стандартный вывод
mov ecx,msg ; Адрес строки 'msg' в 'ecx'
mov edx,msgLen ; Размер строки 'msg' в 'edx'
int 80h ; Вызов ядра

;----- системный вызов 'read' -----
; После вызова инструкции 'int 80h' программа будет ожидать ввода
; строки, которая будет записана в переменную 'buf1' размером 80 байт

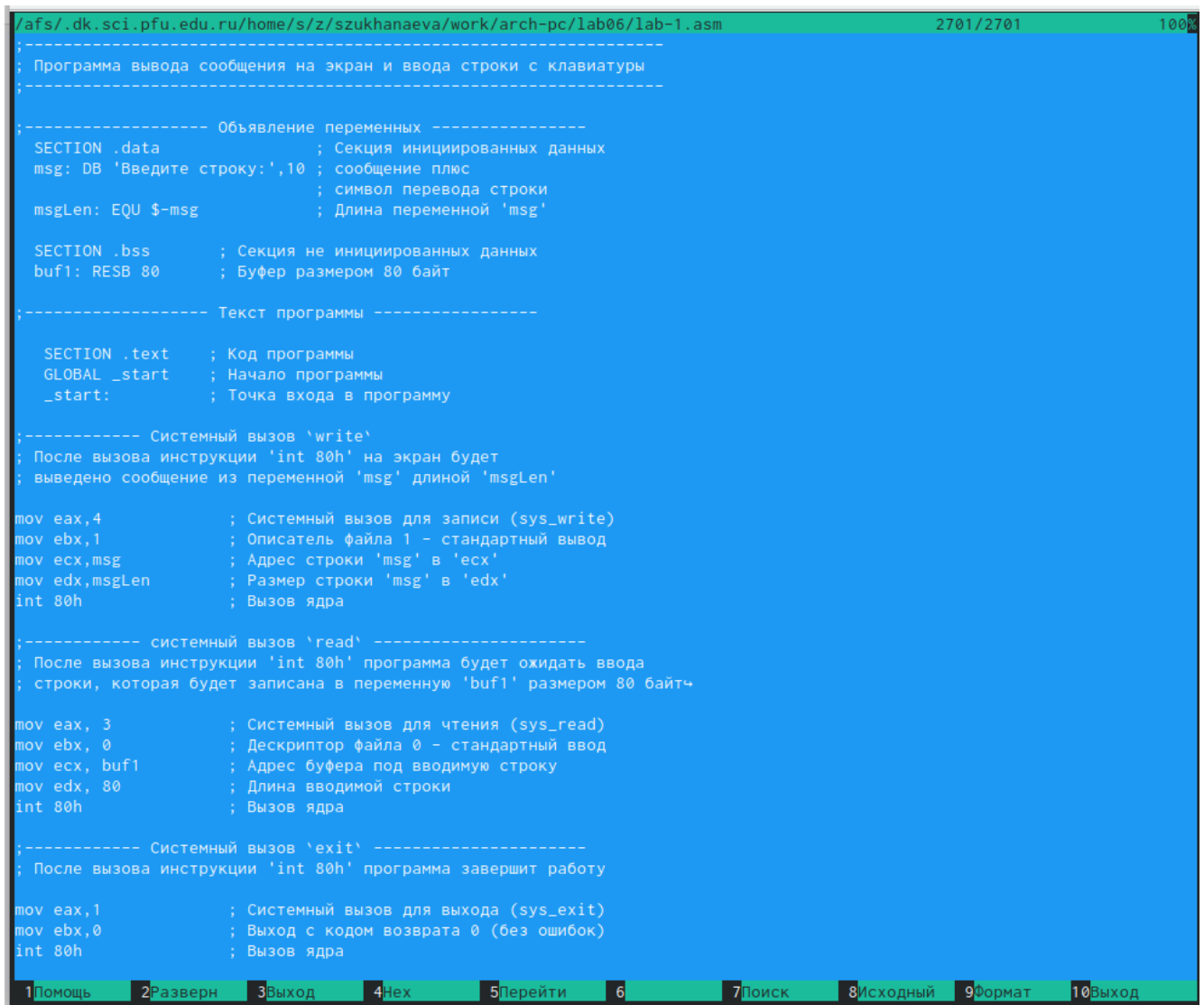
mov eax,3 ; Системный вызов для чтения (sys_read)
mov ebx,0 ;Descriptor файла 0 - стандартный ввод
mov ecx,buf1 ; Адрес буфера под вводимую строку
mov edx,80 ; Длина вводимой строки
int 80h ; Вызов ядра

;----- Системный вызов 'exit' -----
; После вызова инструкции 'int 80h' программа завершит работу

mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра

1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Переместить 7Поиск 8Удалить 9МенюМС 10Выход
```

7. С помощью функциональной клавиши F3 откроем файл lab6-1.asm для просмотра. Убедимся, что файл содержит текст программы.



```
/afs/.dk.sci.pfu.edu.ru/home/s/z/szukhanaeva/work/arch-pc/lab06/lab-1.asm 2701/2701 100%
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
;----- Объявление переменных -----
SECTION .data                ; Секция инициализированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
                                ; символ перевода строки
msgLen: EQU $-msg            ; Длина переменной 'msg'

SECTION .bss                 ; Секция не инициализированных 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
; выведено сообщение из переменной 'msg' длиной 'msgLen'

mov eax,4                    ; Системный вызов для записи (sys_write)
mov ebx,1                    ; Описатель файла 1 - стандартный вывод
mov ecx,msg                  ; Адрес строки 'msg' в 'ecx'
mov edx,msgLen               ; Размер строки 'msg' в 'edx'
int 80h                      ; Вызов ядра

;----- системный вызов 'read' -----
; После вызова инструкции 'int 80h' программа будет ожидать ввода
; строки, которая будет записана в переменную 'buf1' размером 80 байт

mov eax,3                    ; Системный вызов для чтения (sys_read)
mov ebx,0                    ; Дескриптор файла 0 - стандартный ввод
mov ecx,buf1                 ; Адрес буфера под вводимую строку
mov edx,80                   ; Длина вводимой строки
int 80h                      ; Вызов ядра

;----- Системный вызов 'exit' -----
; После вызова инструкции 'int 80h' программа завершит работу

mov eax,1                    ; Системный вызов для выхода (sys_exit)
mov ebx,0                    ; Выход с кодом возврата 0 (без ошибок)
int 80h                      ; Вызов ядра

1Помощь 2Разверн 3Выход 4Нех 5Перейти 6 7Поиск 8Исходный 9Формат 10Выход
```

8. Оттранслируем текст программы lab6-1.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл. Программа выводит строку 'Введите строку:' и ожидает ввода с клавиатуры. На запрос введем свое ФИО.

```
user@dk4n31:~$ nasm -f elf lab6-1.asm
```

```
user@dk4n31:~$ ld -m elf_i386 -o lab6-1 lab6-1.o
```

```
user@dk4n31:~$ ./lab6-1
```

Введем строку:

Имя пользователя

```
user@dk4n31:~$
```

```


szukhanaeva@dk6n65 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
szukhanaeva@dk6n65 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
szukhanaeva@dk6n65 ~/work/arch-pc/lab06 $ ./lab6-1
Введите строку:
Уханаева Сансара Зоригтуевна
szukhanaeva@dk6n65 ~/work/arch-pc/lab06 $

```

### 5.3.1. Подключение внешнего файла in\_out.asm

9. Скачайте файл in\_out.asm со страницы курса в ТУИС.

## Лабораторная работа №5




Лабораторная работа  
№5

Основы  
работы с  
Midnight  
Commander  
(mc).  
Структура  
программы  
на языке  
ассемблера  
NASM

Отметить как пройденное

---



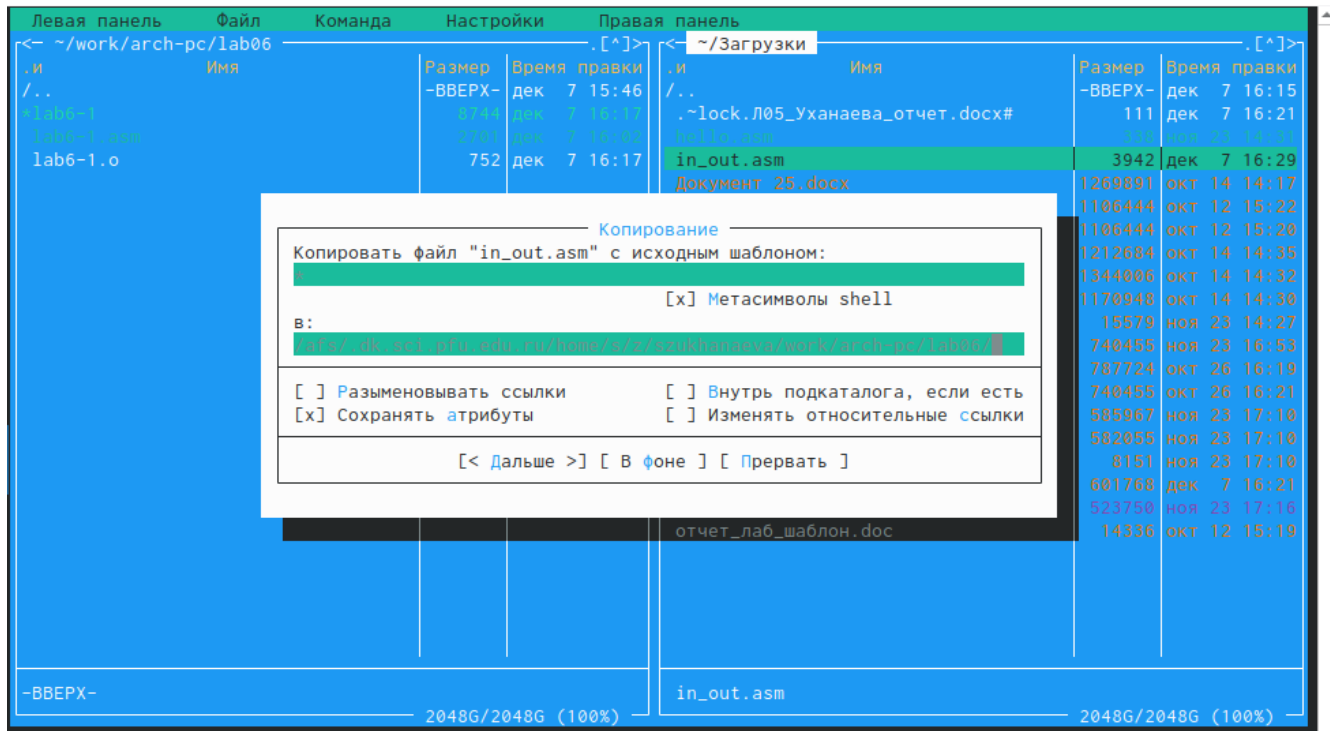
in\_out.asm

10. Подключаемый файл in\_out.asm должен лежать в том же каталоге, что и файл с программой, в которой он используется. В одной из панелей mc откройте каталог с файлом lab6-1.asm. В другой панели каталог со скаченным файлом in\_out.asm (для перемещения между панелями используйте Tab). Скопируйте файл in\_out.asm в каталог с файлом lab6-1.asm с помощью функциональной клавиши F5

Левая панель	Файл	Команда	Настройки	Правая панель
~/work/arch-pc/lab06				~/Загрузки
..	Имя	Размер	Время правки	..
lab6-1		-ВВЕРХ-	дек 7 15:46	..
lab6-1.asm		8744	дек 7 16:17	lock.L05_Уханаева_отчет.docx#
lab6-1.o		2701	дек 7 16:17	in_out.asm
		752	дек 7 16:17	in_out.asm
				Документ 25.docx
				L01_Уханаева_отчет-1.pdf
				L01_Уханаева_отчет.pdf
				L02_Уханаева_отчет-1.pdf
				L02_Уханаева_отчет.docx
				L02_Уханаева_отчет.pdf
				L03_Уханаева_отчет(1).docx
				L03_Уханаева_отчет-1.pdf
				in_out.asm
-ВВЕРХ-		2048G/2048G (100%)		2048G/2048G (100%)

Совет: Автодополнение: M-Tab (или Esc+Tab). Для получения списка нажать дважды.

11. С помощью функциональной клавиши F6 создайте копию файла lab6-1.asm с именем lab6-2.asm. Выделим файл lab6-1.asm, нажмем клавишу F6, введем имя файла lab6-2.asm и нажмем клавишу Enter.



12. Исправим текст программы в файле lab6-2.asm с использованием одпрограмм из внешнего файла in\_out.asm (используйте подпрограммы sprintf, sread и quit) в соответствии с листингом 6.2. Создадим исполняемый файл и проверим его работу.

```
GNU nano 6.3 /afs/.dk.sci.pfu.edu.ru/home/s/z/szukhanaeva/work/arch-pc/lab06/lab6-2.asm
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----

#include 'in_out.asm'          ; подключение внешнего файла

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 'ECX'
mov edx, 80                    ; запись длины вводимого сообщения в 'EDX'
call sread                     ; вызов подпрограммы ввода сообщения

call quit                     ; вызов подпрограммы завершения
```



13. В файле lab6-2.asm заменим подпрограмму sprintLF на sprint. Создадим исполняемый файл и проверьте его работу.

```
lab6-2.asm
~/work/arch-pc/lab06
Сохранить

1 ;-----
2 ; Программа вывода сообщения на экран и ввода строки с клавиатуры
3 ;-----
4
5 %include 'in_out.asm'          ; подключение внешнего файла
6
7 SECTION .data                  ; Секция инициированных данных
8 msg: DB 'Введите строку: ',0h ; сообщение
9
10 SECTION .bss                   ; Секция не инициированных данных
11 buf1: RESB 80                   ; Буфер размером 80 байт
12
13 SECTION .text                   ; Код программы
14     GLOBAL _start                ; Начало программы
15     _start:                       ; Точка входа в программу
16
17 mov eax, msg                    ; запись адреса выводимого сообщения в 'EAX'
18 call sprint                      ; вызов подпрограммы печати сообщения
19
20 mov ecx, buf1                   ; запись адреса переменной в 'EAX'
21 mov edx, 80                     ; запись длины вводимого сообщения в 'EBX'
22 call sread                      ; вызов подпрограммы ввода сообщения
23
24 call quit                      ; вызов подпрограммы завершения
```

### Задание для самостоятельной работы

1. Создадим копию файла lab6-1.asm. Внесем 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 введенную строку на экран.

```

GNU nano 6.3 /afs/.dk.sci.pfu.edu.ru/home/s/z/szukhanaeva/work/arch-pc/lab06/lab6-1.asm
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
;----- Объявление переменных -----
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки

msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не инициализированных 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
; выведено сообщение из переменной 'msg' длиной 'msgLen'
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла 1 - стандартный вывод
mov ecx,msg ; Адрес строки 'msg' в 'ecx'
mov edx,msgLen ; Размер строки 'msg' в 'edx'
int 80h ; Вызов ядра

;----- системный вызов 'read' -----
; После вызова инструкции 'int 80h' программа будет ожидать ввода
; строки, которая будет записана в переменную 'buf1' размером 80 байт
mov eax,3 ; Системный вызов для чтения (sys_read)
mov ebx,0 ; Описатель файла 0 - стандартный ввод

```

2. Получим исполняемый файл и проверим его работу. На приглашение ввести строку введем свою фамилию.

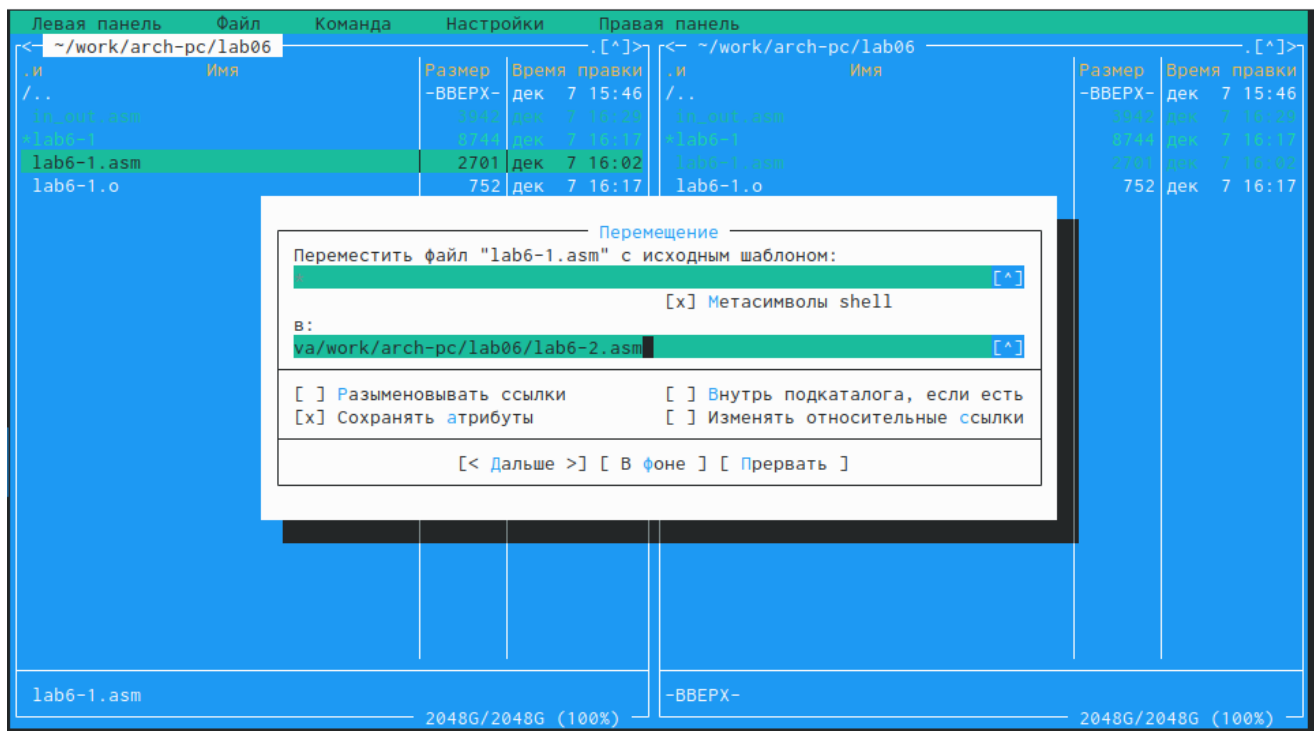
```

szukhanaeva@dk6n65 ~/work/arch-pc/lab06 $ nasm -f elf lab6-1.asm
szukhanaeva@dk6n65 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-1 lab6-1.o
szukhanaeva@dk6n65 ~/work/arch-pc/lab06 $ ./lab6-1
Введите строку:
Уханаева Сансара Зоригтуевна

```

3. Создадим копию файла lab6-2.asm. Исправим 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.



4. Создадим исполняемый файл и проверьте его работу.

```
55szukhanaeva@dk6n65 ~/work/arch-pc/lab06 nasm -f elf lab6-2.asm
szukhanaeva@dk6n65 ~/work/arch-pc/lab06 $ ld -m elf_i386 -o lab6-2 lab6-2.o
szukhanaeva@dk6n65 ~/work/arch-pc/lab06 $ ./lab6-2
Введите строку: Уханаева Сансара Зоригтуевна
```

**Вывод:** Я приобрела практические навыки работы в Midnight Commander. А также освоила инструкции языка ассемблера mov и int.