

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЁТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5**

*дисциплина: Архитектура вычислительных систем*

Студент: Спиридонова Алина Артёмовна

Группа: НБИбд-03-22

**МОСКВА**

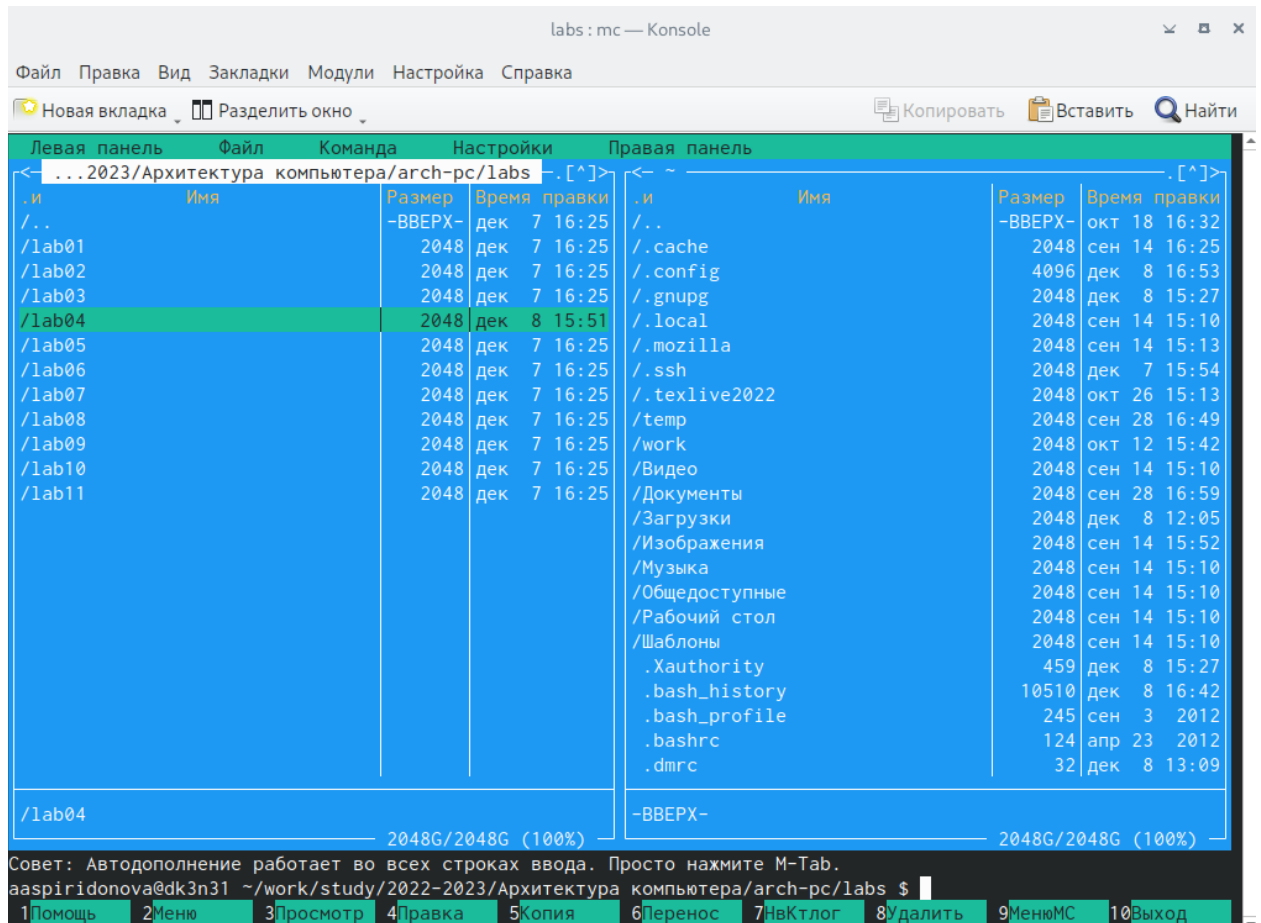
2022 г.

## Цель работы:

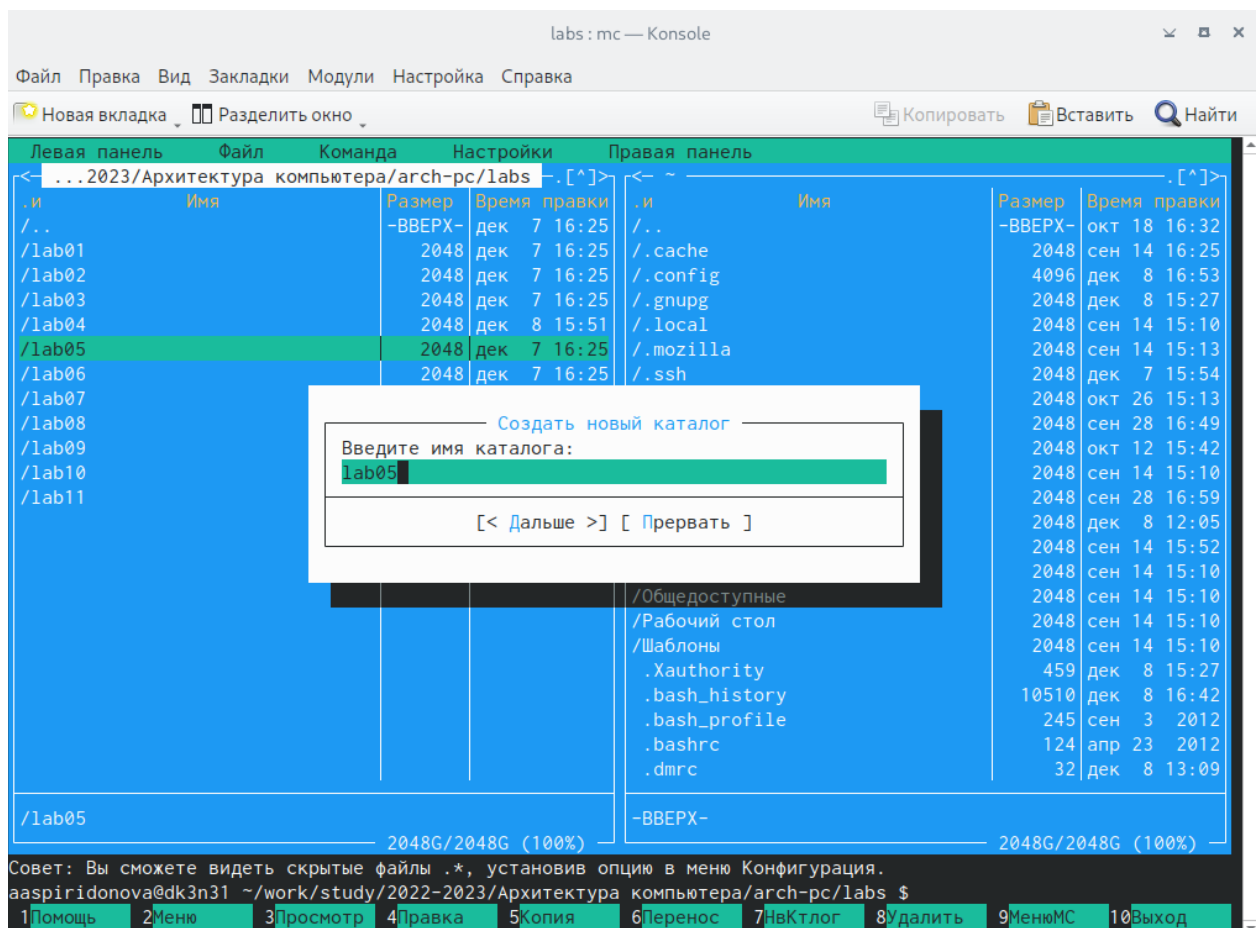
Приобретение практических навыков работы в Midnight Commander.  
Освоение инструкций языка ассемблера mov и int.

## Ход работы:

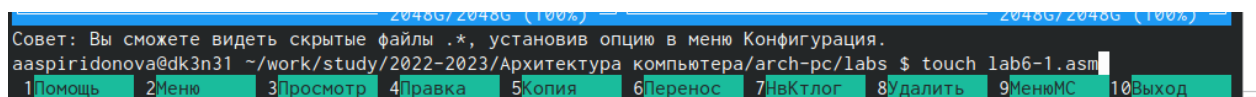
1. Откройте Midnight Commander.
2. Пользуясь клавишами ↑, ↓ и Enter перейдите в каталог ~/work/arch-pc созданный при выполнении лабораторной работы No4.



3. С помощью функциональной клавиши F7 создайте папку lab05 и перейдите в созданный каталог.



4. Пользуясь строкой ввода и командой touch создайте файл lab5-1.asm



5. С помощью функциональной клавиши F4 откройте файл lab5-1.asm для редактирования во встроенном редакторе. Как правило в качестве встроенного редактора Midnight Commander используется редакторы nano или mcedit.

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

```
labs : mc — Konsole

Файл  Правка  Вид  Закладки  Модули  Настройка  Справка

Новая вкладка  Разделить окно

lab5-1.asm  [M~] 20 L: [ 1+31 32/ 36] *(2100/2434b) 1085 0x43D
;-----
; 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
```

7. С помощью функциональной клавиши F3 откройте файл lab5-1.asm для просмотра. Убедитесь, что файл содержит текст программы.

```
labs : mc — Konsole

Файл  Правка  Вид  Закладки  Модули  Настройка  Справка

Новая вкладка  Разделить окно

/afs/.dk.sci.pfu.edu.ru/home/a/a/aaspiridonova/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab5-1.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 ; 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
```

8. Оттранслируйте текст программы lab5-1.asm в объектный файл. Выполните компоновку объектного файла и запустите получившийся

исполняемый файл. Программа выводит строку 'Введите строку:' и ожидает ввода с клавиатуры. На запрос введите Ваши ФИО.

```
user@dk4n31:~$ nasm -f elf lab5-1.asm
```

```
user@dk4n31:~$ ld -m elf_i386 -o lab5-1 lab5-1.o
```

```
user@dk4n31:~$ ./lab5-1
```

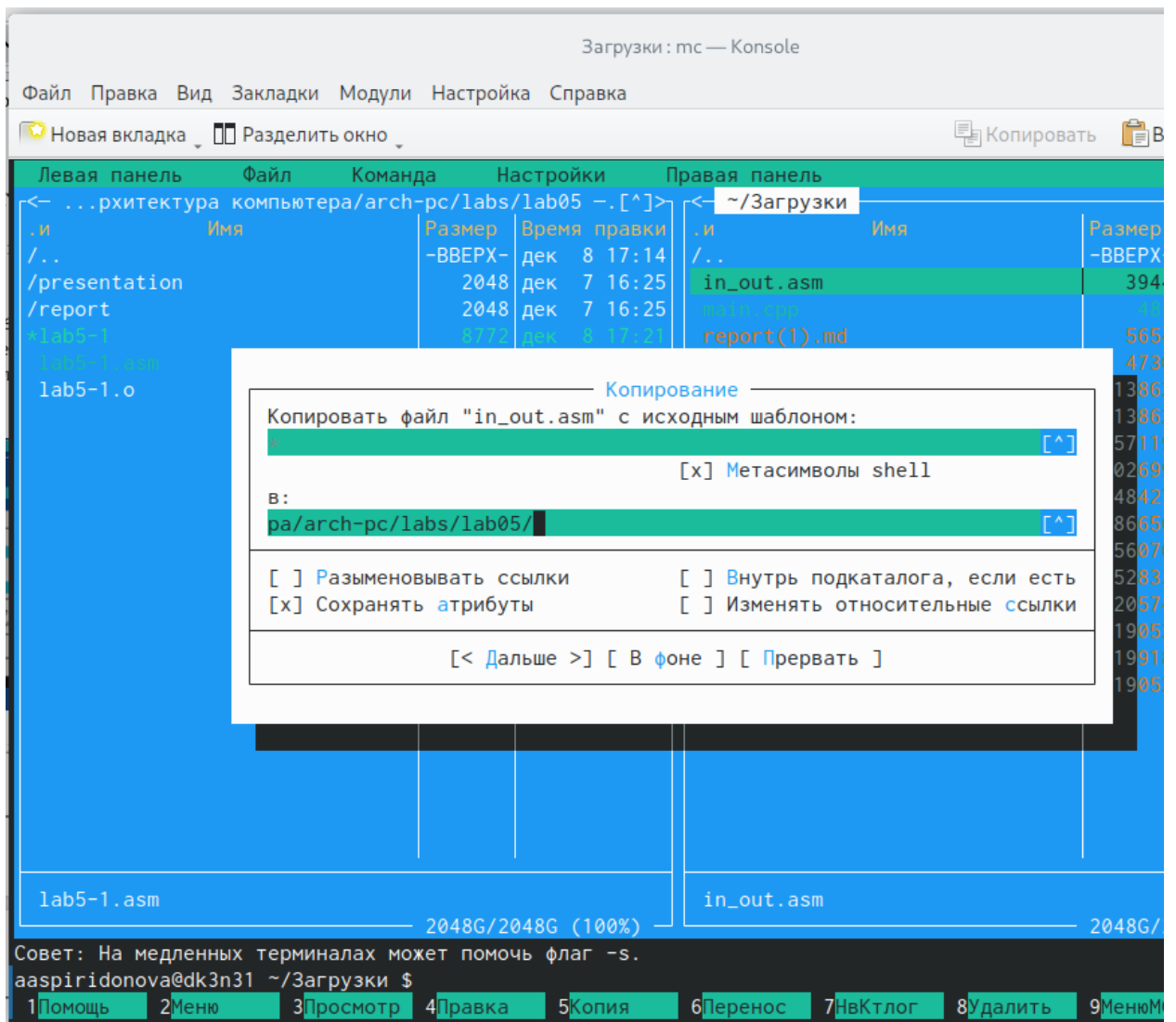
Введите строку:

Имя пользователя

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

```
aaspiridonova@dk3n31 ~ $ mc
aaspiridonova@dk3n31 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ nasm -f elf lab5-1.asm
lab5-1.asm:26: warning: label alone on a line without a colon might be in error [-w+label-orphan]
aaspiridonova@dk3n31 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o
aaspiridonova@dk3n31 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ ./lab5-1
Введите строку:
Спиридонова Алина
```

9. Скачайте файл in\_out.asm со страницы курса в ТУИС. Подключаемый файл in\_out.asm должен лежать в том же каталоге, что и файл с программой, в которой он используется.



10. С помощью функциональной клавиши F6 создайте копию файла lab5-1.asm с именем lab5-2.asm. Выделите файл lab5-1.asm, нажмите клавишу F6, введите имя файла lab5-2.asm и нажмите клавишу Enter.

```
lab05: mc — Konsole
Файл Правка Вид Закладки Модули Настройка Справка
Новая вкладка Разделить окно
GNU nano 6.3 /afs/dk.sci.pfu.edu.ru/home/a/aaspiridonova/work/study/2022-2023/Архитектура_компьютера/arch-pc/labs/lab05/lab5-2.asm
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
include "in_out.asm" ; подключение внешнего файла
section .data ; Секция инициализированных данных
msg: db "Введите строку: ", 0h ; сообщение
buff: resb 80 ; буфер размером 80 байт
section .text ; Код программы
extern _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, buff ; запись адреса переменной в 'ECX'
mov edx, 80 ; запись длины выводимого сообщения в 'EDX'
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения
```

11. Исправьте текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in\_out.asm (используйте подпрограммы sprintf, sread и quit) в соответствии с листингом 6.2. Создайте исполняемый файл и проверьте его работу.

```
Приложения Места gedit
Открыть in_out.asm
1;----- slen -----
2; Функция вычисления длины сообщения
3slen:
4    push    ebx
5    mov     ebx, eax
6
7nextchar:
8    cmp     byte [eax], 0
9    jz      finished
10   inc     eax
11   jmp     nextchar
12
13finished:
14   sub     eax, ebx
15   pop     ebx
16   ret
17
18;----- sprintf -----
19; Функция печати сообщения
20; входные данные: mov eax, <message>
21sprintf:
22   push    edx
23   push    ecx
24   push    ebx
25   push    eax
26   call    slen
27
28   mov     edx, eax
29   pop     eax
30
31   mov     ecx, eax
32   mov     ebx, 1
33   mov     eax, 4
34   int     80h
35
36   pop     ebx
37   pop     ecx
38   pop     edx
39   ret
40
41
42;----- sprintfLF -----
43; Функция печати сообщения с переводом строки
44; входные данные: mov eax, <message>
45sprintfLF:
46   call    sprintf
47
```

**Вывод: В ходе выполнения лабораторной работы я приобрела практические навыки работы в Midnight Commander, а также освоила инструкции языка ассемблера mov и int.**