

Лабораторная работа №5

Архитектура вычислительных систем

Басманова Дарья Кириллова

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
4	Выполнение самостоятельной работы	12
5	Выводы	14

Список иллюстраций

3.1	1.png	6
3.2	2.png	7
3.3	3.png	8
3.4	4.png	9
3.5	5.png	9
3.6	6.png	10
3.7	7.png	11
4.1	8.png	13

1 Цель работы

Преобретение практических навыков работы с программой ассемблер, Midnight Commander.

2 Задание

1. Создать копию файла и внести изменения в программу, так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введённую строку на экран.
2. Получить исполняемый файл и проверить его работу. На приглашение ввести строку введите свою фамилию.
3. Создать копию файла и исправить 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 введённую строку на экран

3 Выполнение лабораторной работы

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 3.1)

- 1) Откроем Midnight Commander и с помощью функциональной клавиши F7 создадим папку lab05 и перейдем в данный каталог.

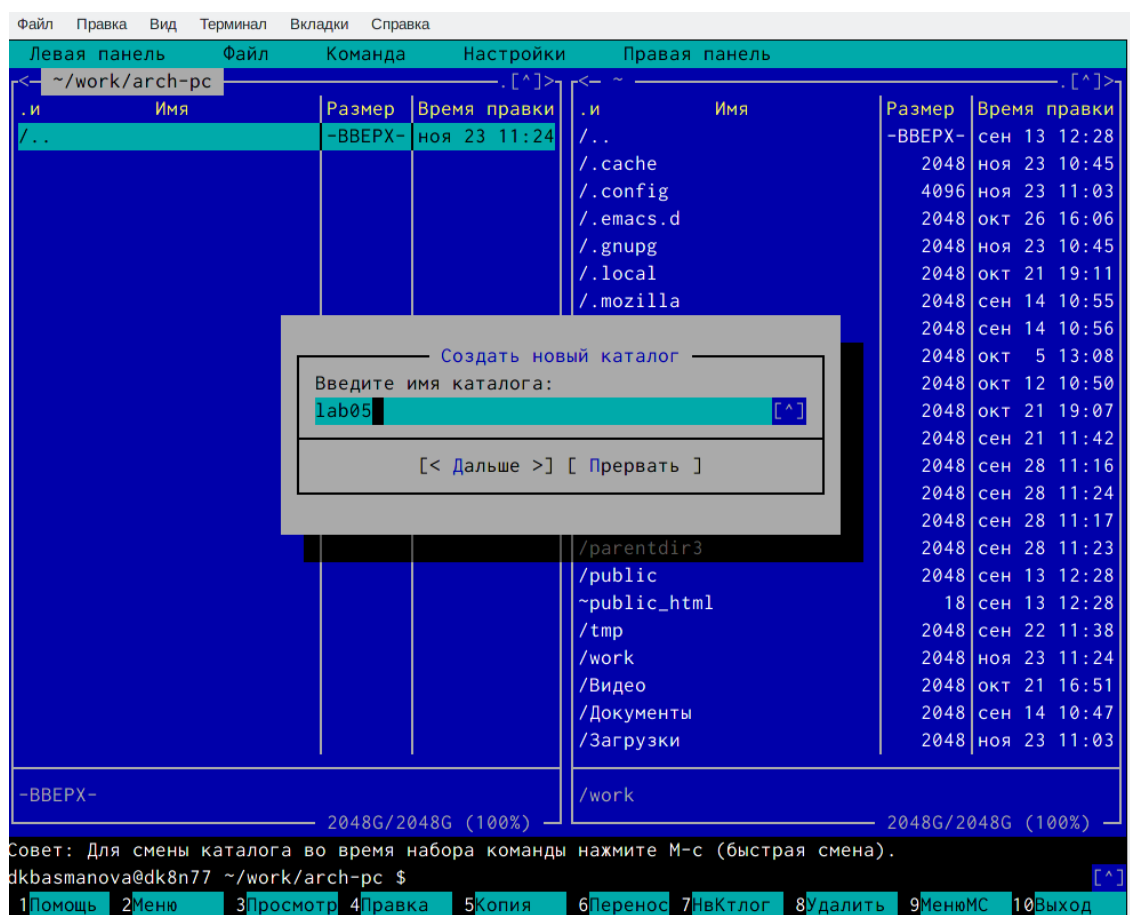


Рис. 3.1: 1.png

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 3.2)

2) Пользуясь строкой ввода и командой touch создадим файл lab5-1.asm.

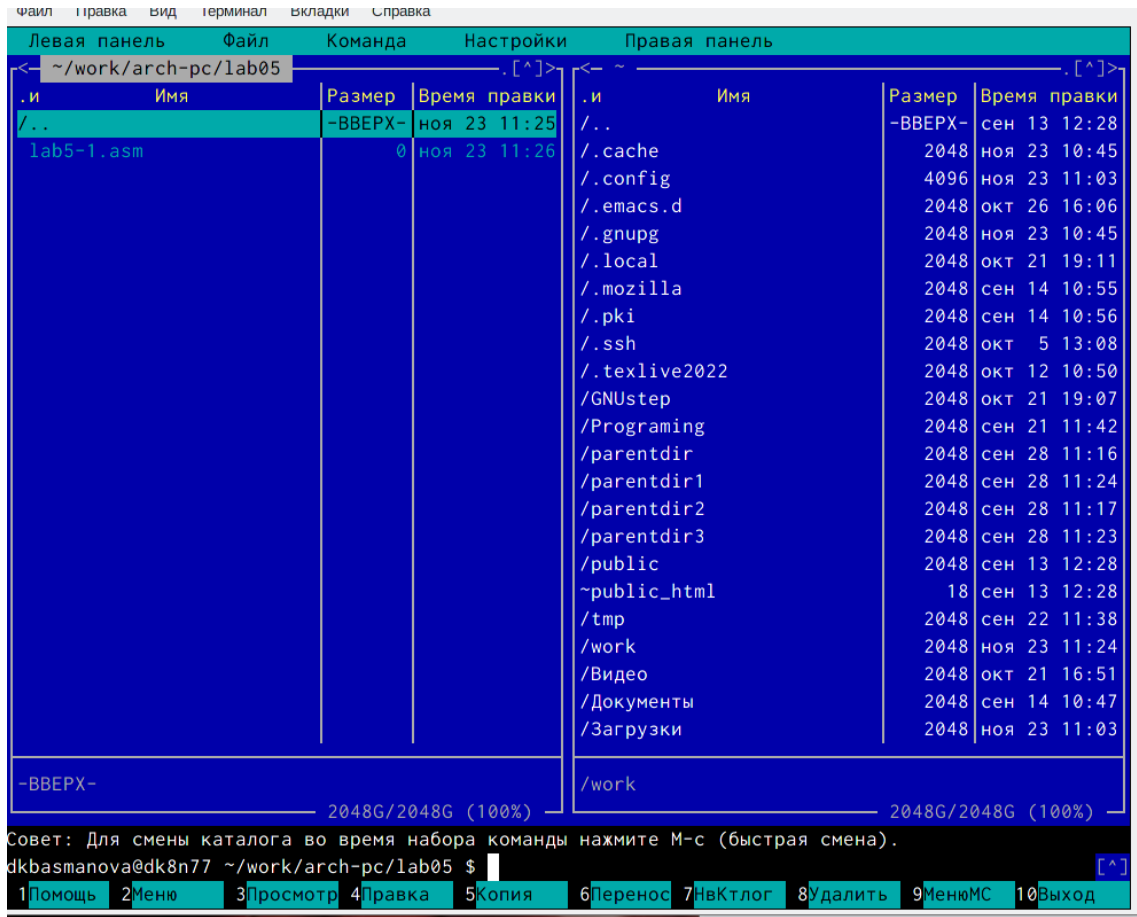


Рис. 3.2: 2.png

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 3.3)

3) С помощью функциональной клавиши F4 откроем файл lab6-1.asm для редактирования во встроенном редакторе.

```
lab5-1.asm      [-M--] 24 L: [ 1+ 0 1/ 51] *(24 /3387b) 0045 0x02D      [*][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 ; Дескриптор файла 0 - стандартный ввод
mov ecx, buf1 ; Адрес буфера под вводимую строку
mov edx, 80 ; Длина вводимой строки
1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перейти 7Поиск 8Удалить 9МенюМС 10Выход
```

Рис. 3.3: 3.png

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 3.4)

4) Сохраним изменения


```
lab5-1.asm      [-M--] 32 L: [ 1+12 13/ 51] *(871 /3387b) 0010 0x00A      [*][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

----- Сохранить файл -----
тверждаете запись файла "/afs/.dk.sci.pfu.edu.ru/home/d/k/dkbasmanova/work/arch-pc/lab05/lab5-1.as

[ Сохранить ] [ Прервать ]

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

Рис. 3.4: 4.png

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 3.5)

- 5) Оттранслируем текст программы lab6-1.asm в объектный файл. Выполняем компоновку и запускаем. Вводим ФИО.

```
dkbasmanova@dk2n22 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o
dkbasmanova@dk2n22 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ ./lab5-1
Введите строку:
Басманова Дарья Кирилловна
dkbasmanova@dk2n22 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $
```

Рис. 3.5: 5.png

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 3.6)

- 6) Скачаем файл in_out.asm со страницы в ТУИС и переместим его в файл с программой.

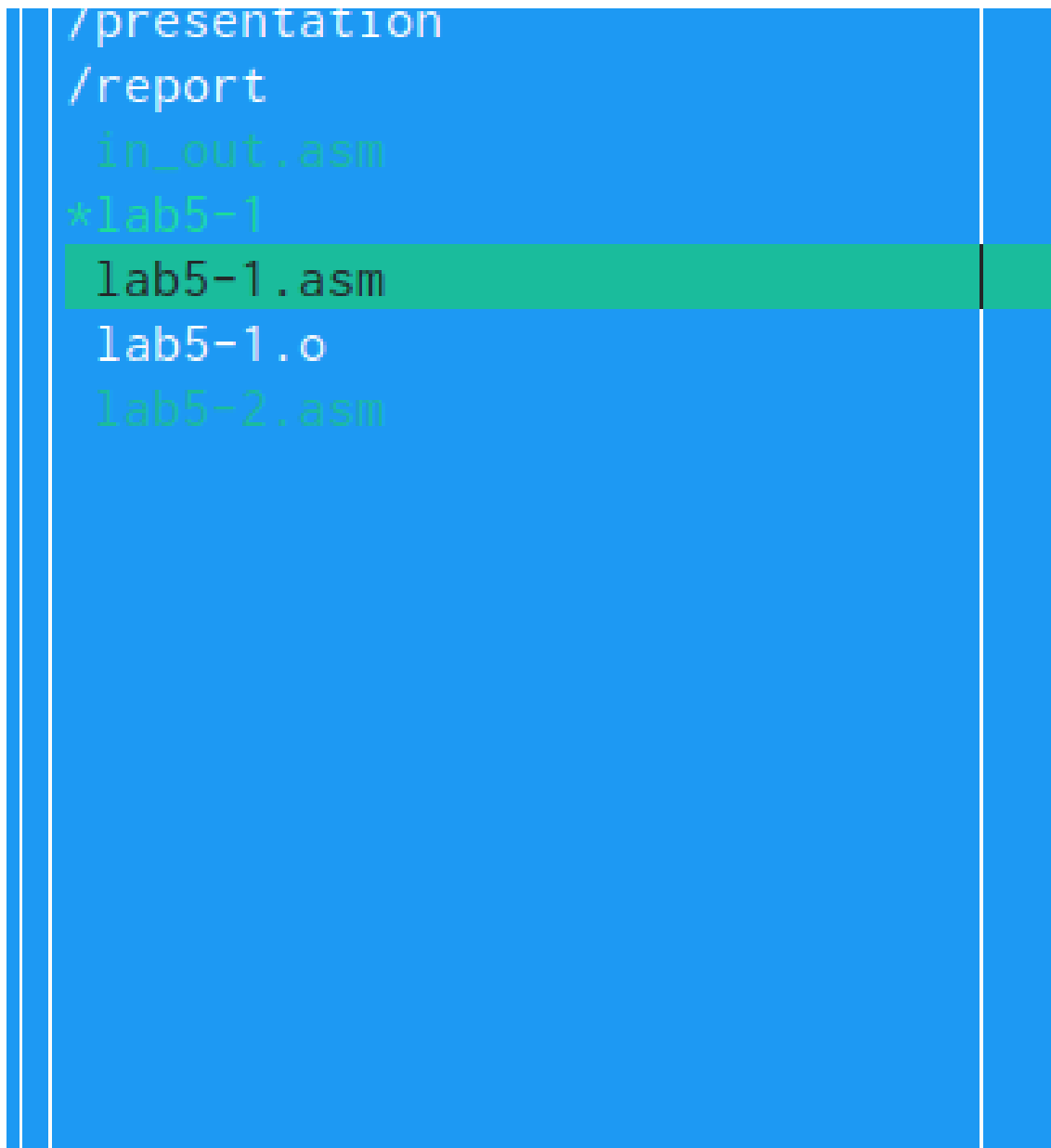


Рис. 3.6: 6.png

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 3.7)

- 7) С помощью функциональной клавиши F6 создаем копию файла lab5-1.asm

с именем lab5-2.asm.

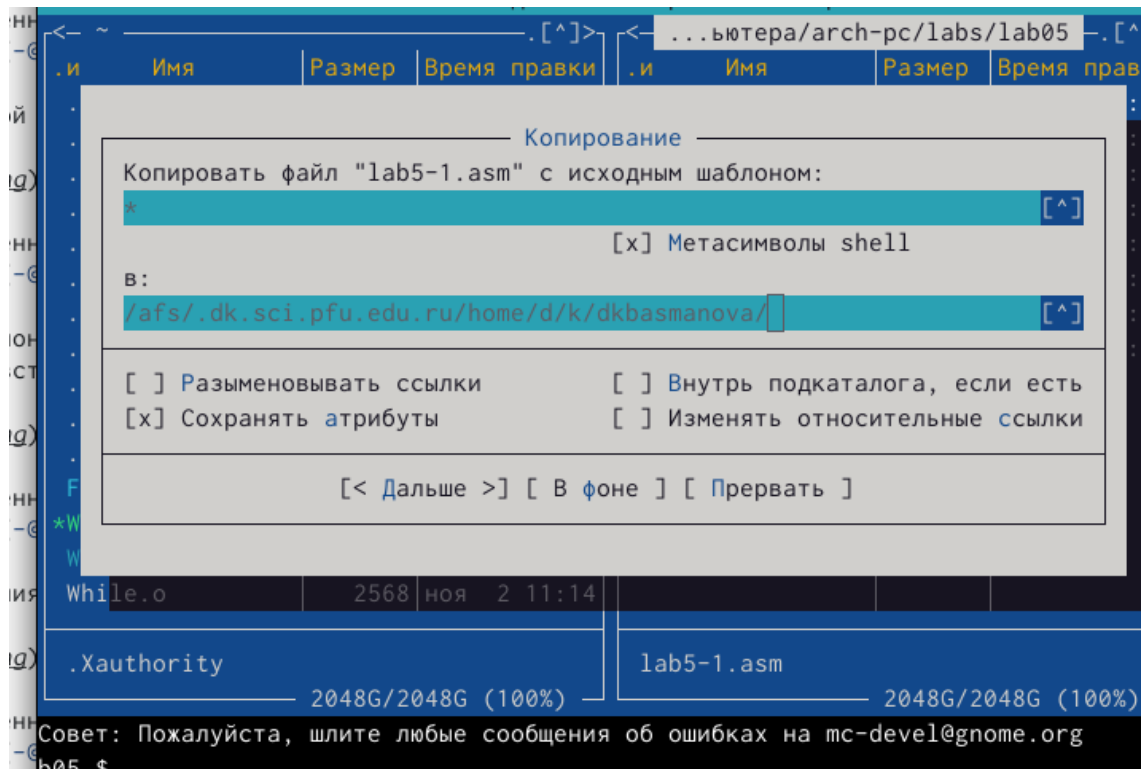


Рис. 3.7: 7.png

- 8) При замене подпрограммы `sprintf` на `spint` текст выводится на той же строке, где расположена фраза “Введите строку”, а не на отдельной.

4 Выполнение самостоятельной работы

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 4.1)

- 1) Создадим копию файла lab5-1.asm и назовем его lab5-3.asm и внесем изменения, так чтобы она работала по определенному алгоритму.

```
...rk/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05/lab5-2.asm Измен  
;-----  
; Программа вывода сообщения на экран и ввода строки с клавиатуры  
;-----  
#include 'in_out.asm' ; подключение внешнего файла  
  
SECTION .data ; Секция инициированных данных  
msg: DB 'Введите строку: ',0h ; сообщение  
SECTION .bss ; Секция не инициированных 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 sprintLF ; 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
```

^G Справка	^O Записать	^W Поиск	^K Вырезать	^T Выполнить	M-U Отмена
^X Выход	^R ЧитФайл	^V Замена	^U Вставить	^C Позиция	M-E Повтор

Рис. 4.1: 8.png

5 Выводы

После выполнения работы я получила практические знания работы в Midnight Commander. А так же освоила инструкции языка ассемблера mov и int