

**Лабораторная работа No5. Основы
работы с Midnight Commander (mc).
Структура программы на языке
ассемблера NASM. Системные вызовы в
ОС GNU Linux**

Простейший вариант

Диана Алексеевна Садова

Содержание

1	Цель работы	6
2	Задание	7
2.1	Порядок выполнения лабораторной работы	7
2.1.1	Откройте Midnight Commander (рис.2.1)(рис.2.2)	7
2.1.2	Пользуясь клавишами вверх, вниз и Enter перейдите в каталог ~/work/arch-pc созданный при выполнении лабораторной работы No4 (рис.2.3).	8
2.1.3	С помощью функциональной клавиши F7 создайте папку lab05 (рис.2.4) и перейдите в созданный каталог(рис.2.5).	8
2.1.4	Пользуясь строкой ввода и командой touch создайте файл lab05-1.asm (рис.2.6), (рис.2.7).	9
2.1.5	С помощью функциональной клавиши F4 откройте файл lab05-1.asm для редактирования во встроенном редакторе. Как правило в качестве встроенного редактора Midnight Commander используется редакторы nano или mcedit (рис.2.8).	10
2.1.6	Введите текст программы из листинга 5.1 (можно без комментариев), сохраните изменения и закройте файл (рис.2.9).	10
2.1.7	С помощью функциональной клавиши F3 откройте файл lab05-1.asm для просмотра. Убедитесь, что файл содержит текст программы (рис.2.10).	12
2.1.8	Оттранслируйте текст программы lab05-1.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 (рис.2.11), (рис.2.12), (рис.2.13), (рис.2.14), (рис.2.15), (рис.2.16).	13
2.2	Подключение внешнего файла in_out.asm	14
2.2.1	Скачайте файл in_out.asm со страницы курса в ТУИС (рис.2.17).	16
2.2.2	Подключаемый файл in_out.asm должен лежать в том же каталоге, что и файл с программой, в которой он используется.	16
2.2.3	С помощью функциональной клавиши F6 создайте копию файла lab05-1.asm с именем lab05-2.asm. Выделите файл lab05-1.asm, нажмите клавишу F6, введите имя файла lab05-2.asm и нажмите клавишу Enter (рис.2.20).	18

2.2.4	Исправьте текст программы в файле lab05-2.asm с использование подпрограмм из внешнего файла in_out.asm (используйте подпрограммы sprintf, fread и quit) в соответствии с листингом 5.2. Создайте исполняемый файл и проверьте его работу (рис.2.21).	18
2.2.5	В файле lab5-2.asm замените подпрограмму sprintf на printf. Создайте исполняемый файл и проверьте его работу. В чем разница? (рис.2.22)	19
3	Теоретическое введение	20
4	Выполнение лабораторной работы	21
4.1	Задание для самостоятельной работы	21
4.1.1	Создайте копию файла lab5-1.asm. Внесите изменения в программу (без использования внешнего файла in_out.asm), так чтобы она работала по следующему алгоритму: (рис.2.23), (рис.2.24)	21
4.1.2	Получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите свою фамилию (рис.2.25).	22
4.1.3	Создайте копию файла lab5-2.asm. Исправьте текст программы с использование подпрограмм из внешнего файла in_out.asm, так чтобы она работала по следующему алгоритму: (рис.2.26), (рис.2.27)	23
4.1.4	Создайте исполняемый файл и проверьте его работу (рис.2.28).	24
5	Выводы	25
	Список литературы	26

Список иллюстраций

2.1	Вводим команду <code>mc</code>	7
2.2	Окно Midnight	7
2.3	Переходим в каталог <code>~/work/arch-pc</code>	8
2.4	Создаём папку <code>lab05</code>	8
2.5	Переходим в папку <code>lab05</code>	8
2.6	Создаём файл <code>lab05-1.asm</code>	9
2.7	Проверяем наличие файла <code>lab05-1.asm</code>	9
2.8	Открываем файл <code>lab05-1.asm</code> в редакторе <code>mcedit</code>	10
2.9	Вводим текст программы	11
2.10	Открываем файл <code>lab05-1.asm</code> для просмотра	12
2.11	Вводим команду	13
2.12	Проверяем корректность выполнения программы	13
2.13	Вводим команду	13
2.14	Удостоверемся в правильности выполненной программы	14
2.15	Вводим команду	14
2.16	Возводим свои имя и фамилию	14
2.17	Скачиваем файл <code>in_out.asm</code> с ТУИС	16
2.18	Копируем файл <code>in_out.asm</code> и вставляем в каталог <code>lab05</code>	17
2.19	Проверяем корректность выполненной программы	17
2.20	С помощью клавиши <code>F6</code> перемещаем файл <code>lab05-1.asm</code> в каталог <code>lab05</code> с новым именем	18
2.21	Вводим код программы	19
2.22	Редактируем и выводим работу программы в консоль	19
4.1	Создаем копию файла <code>lab5-1.asm</code> с именем <code>lab5-1_1.asm</code>	21
4.2	Вводим код программы	22
4.3	Проверяем корректность выполненной программы	22
4.4	Создаем копию файла <code>lab5-2.asm</code> с именем <code>lab5-2_1.asm</code>	23
4.5	Вводим код програмы	24
4.6	Проверяем коректность выполненной программы	24

Список таблиц

1 Цель работы

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

2 Задание

2.1 Порядок выполнения лабораторной работы

2.1.1 Откройте Midnight Commander (рис.2.1)(рис.2.2)



Рис. 2.1: Вводим команду mc

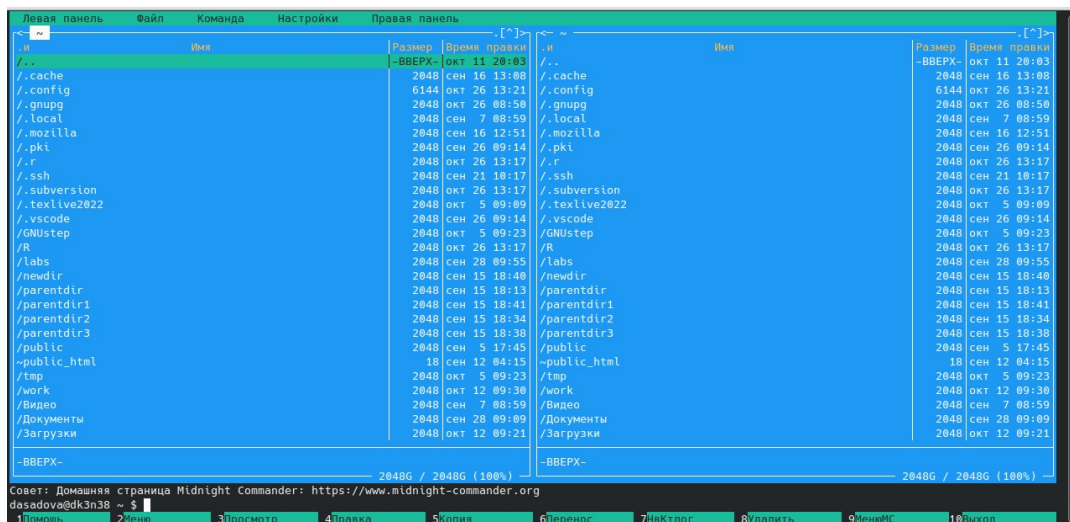


Рис. 2.2: Окно Midnight

С помощью команды mc мы переходим в Midnight и продолжаем работу

**2.1.2 Пользуясь клавишами вверх, вниз и Enter перейдите в каталог
~/work/arch-pc созданный при выполнении лабораторной
работы No4 (рис.2.3).**

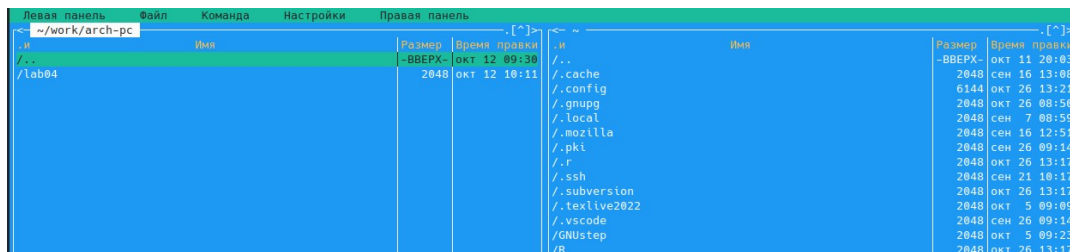


Рис. 2.3: Переходим в каталог ~/work/arch-pc

**2.1.3 С помощью функциональной клавиши F7 создайте папку lab05
(рис.2.4) и перейдите в созданный каталог(рис.2.5).**

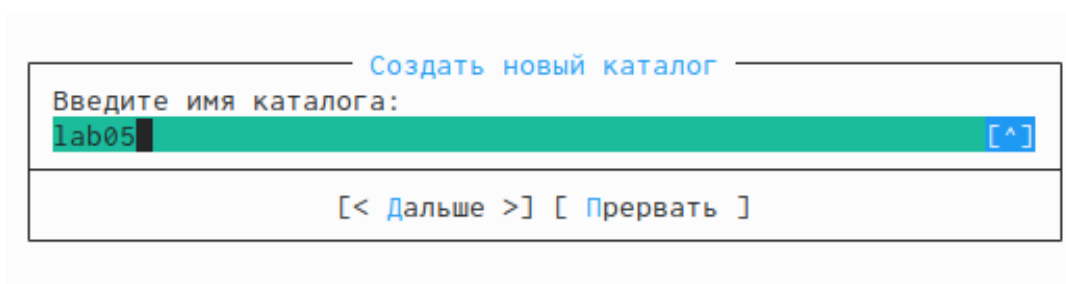


Рис. 2.4: Создаём папку lab05

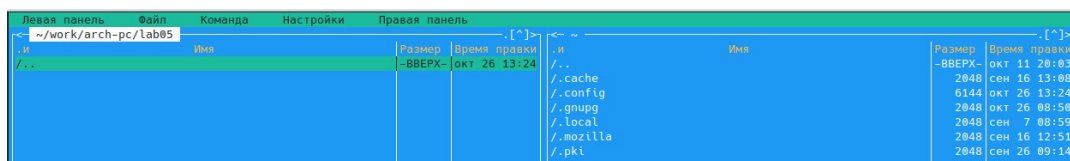


Рис. 2.5: Переходим в папку lab05

2.1.4 Пользуясь строкой ввода и командой touch создайте файл lab05-1.asm (рис.2.6), (рис.2.7).

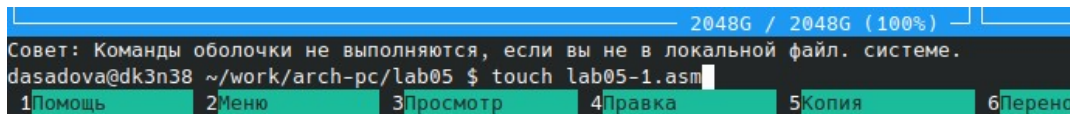


Рис. 2.6: Создаём файл lab05-1.asm

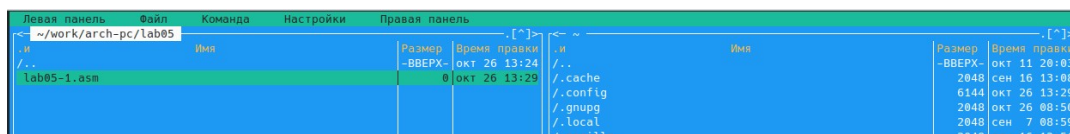


Рис. 2.7: Проверяем наличие файла lab05-1.asm

2.1.5 С помощью функциональной клавиши F4 откройте файл lab05-1.asm для редактирования во встроенном редакторе. Как правило в качестве встроенного редактора Midnight Commander используется редакторы nano или mcedit (рис.2.8).

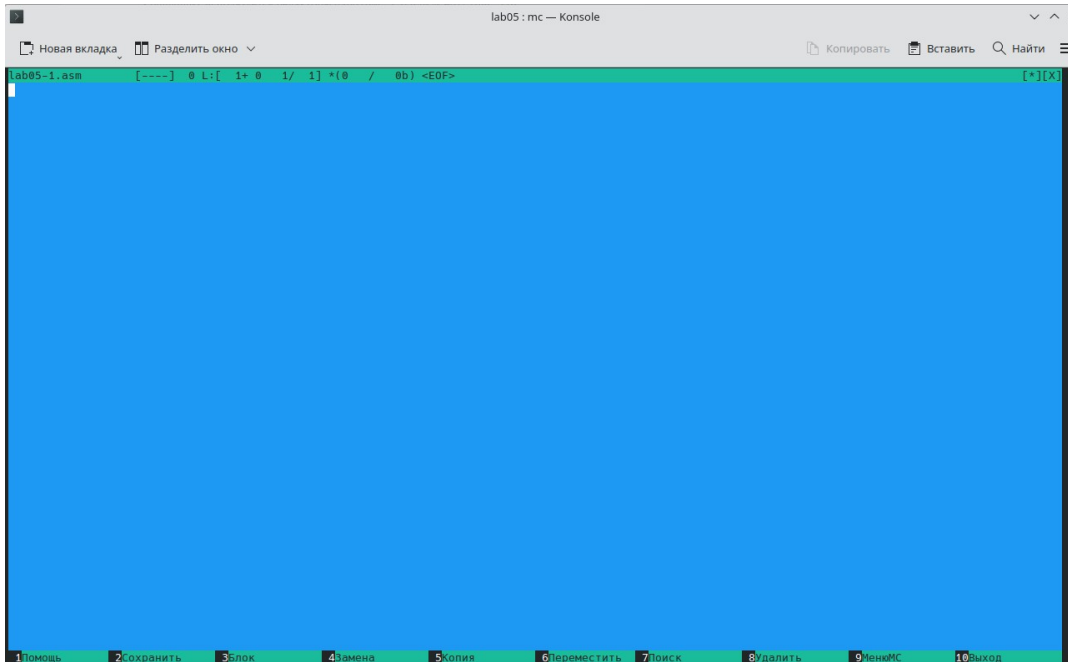


Рис. 2.8: Открываем файл lab05-1.asm в редакторе mcedit

2.1.6 Введите текст программы из листинга 5.1 (можно без комментариев), сохраните изменения и закройте файл (рис.2.9).

Листинг 5.1. Программа вывода сообщения на экран и ввода строки с клавиатуры

```

lab05-1.asm      [----] 48 L: [ 1+11 12/ 35] *(949 /2891b) 0010 0x00A
;-----
; Программы 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

```

Рис. 2.9: Вводим текст программы

2.1.7 С помощью функциональной клавиши F3 откройте файл lab05-1.asm для просмотра. Убедитесь, что файл содержит текст программы (рис.2.10).

```
/afs/.dk.sci.pfu.edu.ru/home/d/a/dasadova/work/arch-pc/lab05/lab05-1.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
;----- 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
```

Рис. 2.10: Открываем файл lab05-1.asm для просмотра

2.1.8 Оттранслируйте текст программы lab05-1.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 (рис.2.11), (рис.2.12), (рис.2.13), (рис.2.14), (рис.2.15), (рис.2.16).

```
Совет: M-! позволит вам выполнять программы и видеть вывод в просмотрщике.  
dasadova@dk3n63 ~/work/arch-pc/lab05 $ nasm -f elf lab05-1.asm
```

Рис. 2.11: Вводим команду

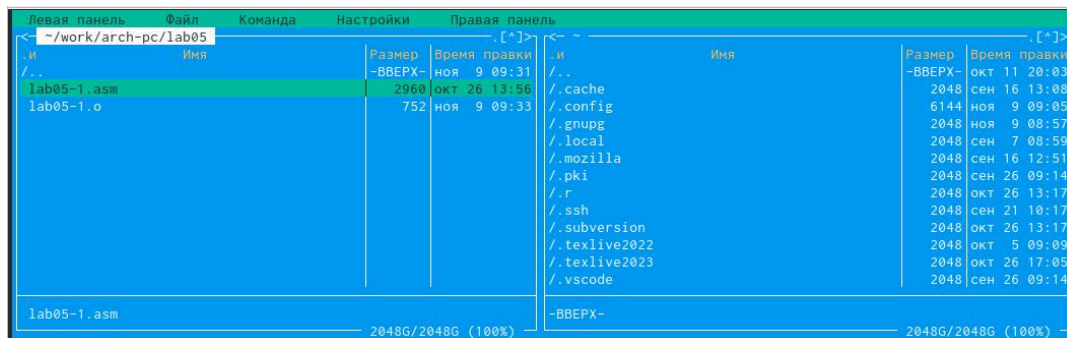


Рис. 2.12: Проверяем корректность выполнения программы

```
Совет: M-! позволит вам выполнять программы и видеть вывод в просмотрщике.  
dasadova@dk3n63 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab05-1 lab05-1.o
```

Рис. 2.13: Вводим команду

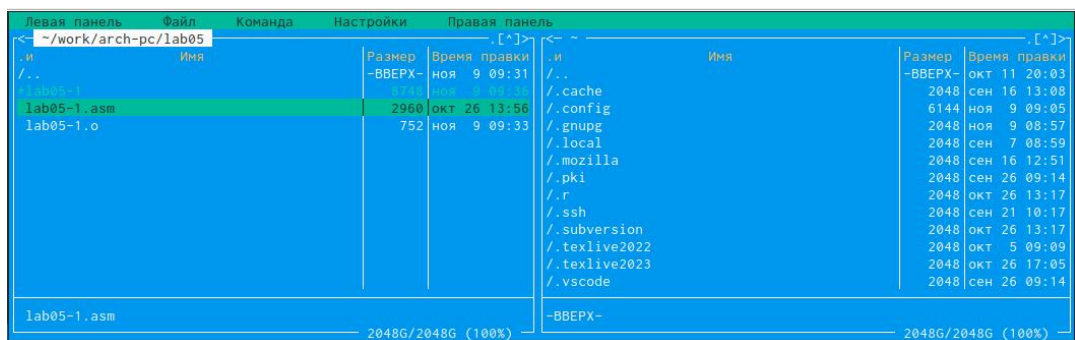


Рис. 2.14: Удостоверемся в правильности выполненной программы

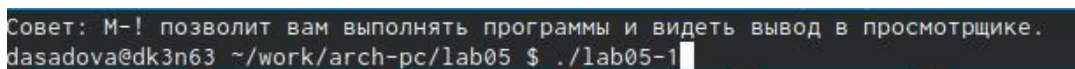


Рис. 2.15: Вводим команду

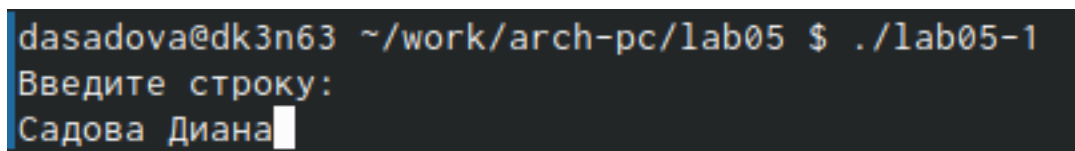


Рис. 2.16: Возводим свои имя и фамилию

2.2 Подключение внешнего файла in_out.asm

Для упрощения написания программ часто встречающиеся одинаковые участки кода (такие как, например, вывод строки на экран или выход из программы) можно оформить в виде подпрограмм и сохранить в отдельные файлы, а во всех нужных местах поставить вызов нужной подпрограммы. Это позволяет сделать основную программу более удобной для написания и чтения.

NASM позволяет подключать внешние файлы с помощью директивы `%include`, которая предписывает ассемблеру заменить эту директиву содержимым файла. Подключаемые файлы также написаны на языке ассемблера. Важно отметить, что директива `%include` в тексте программы должна стоять раньше, чем встречаются

вызовы подпрограмм из подключаемого файла. Для вызова подпрограммы из внешнего файла используется инструкция `call`, которая имеет следующий вид

```
call <function>
```

где `function` имя подпрограммы.

Для выполнения лабораторных работ используется файл `in_out.asm1`, который содержит следующие подпрограммы [4]:

- `slen` – вычисление длины строки (используется в подпрограммах печати сообщения д
- `sprint` – вывод сообщения на экран, перед вызовом `sprint` в регистр `eax` необходим
- `sprintLF` – работает аналогично `sprint`, но при выводе на экран добавляет к сообщ
- `sread` – ввод сообщения с клавиатуры, перед вызовом `sread` в регистр `eax` необходи
- `iprint` – вывод на экран чисел в формате ASCII, перед вызовом `iprint` в регистр `eax`
- `iprintLF` – работает аналогично `iprint`, но при выводе на экран после числа добав
- `atoi` – функция преобразует `ascii`-код символа в целое число и записывает результа
- `quit` – завершение программы.

2.2.1 Скачайте файл in_out.asm со страницы курса в ТУИС (рис.2.17).

Лабораторная работа №5

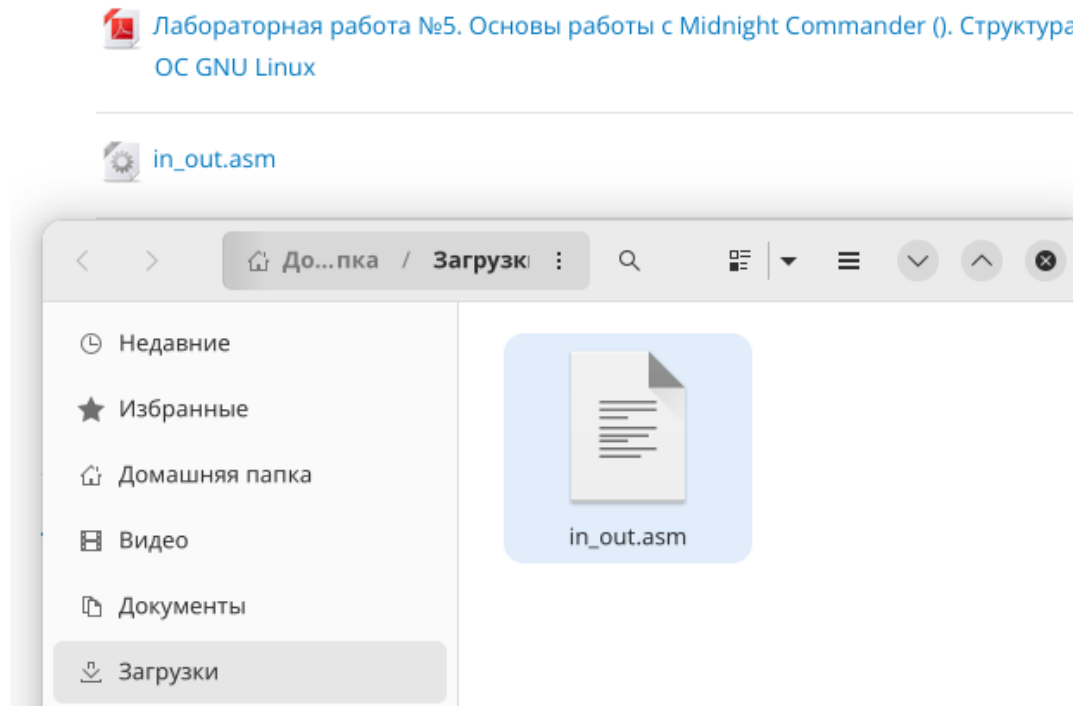


Рис. 2.17: Скачиваем файл in_out.asm с ТУИС

2.2.2 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 lab05-1.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 lab05-1.asm с помощью функциональной клавиши F5 (рис.2.18), (рис.2.19).

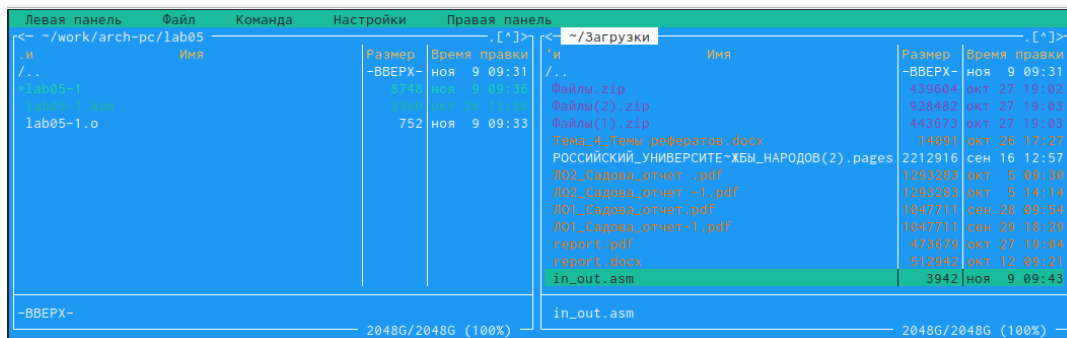


Рис. 2.18: Копируем файл `in_out.asm` и вставляем в каталог `lab05`

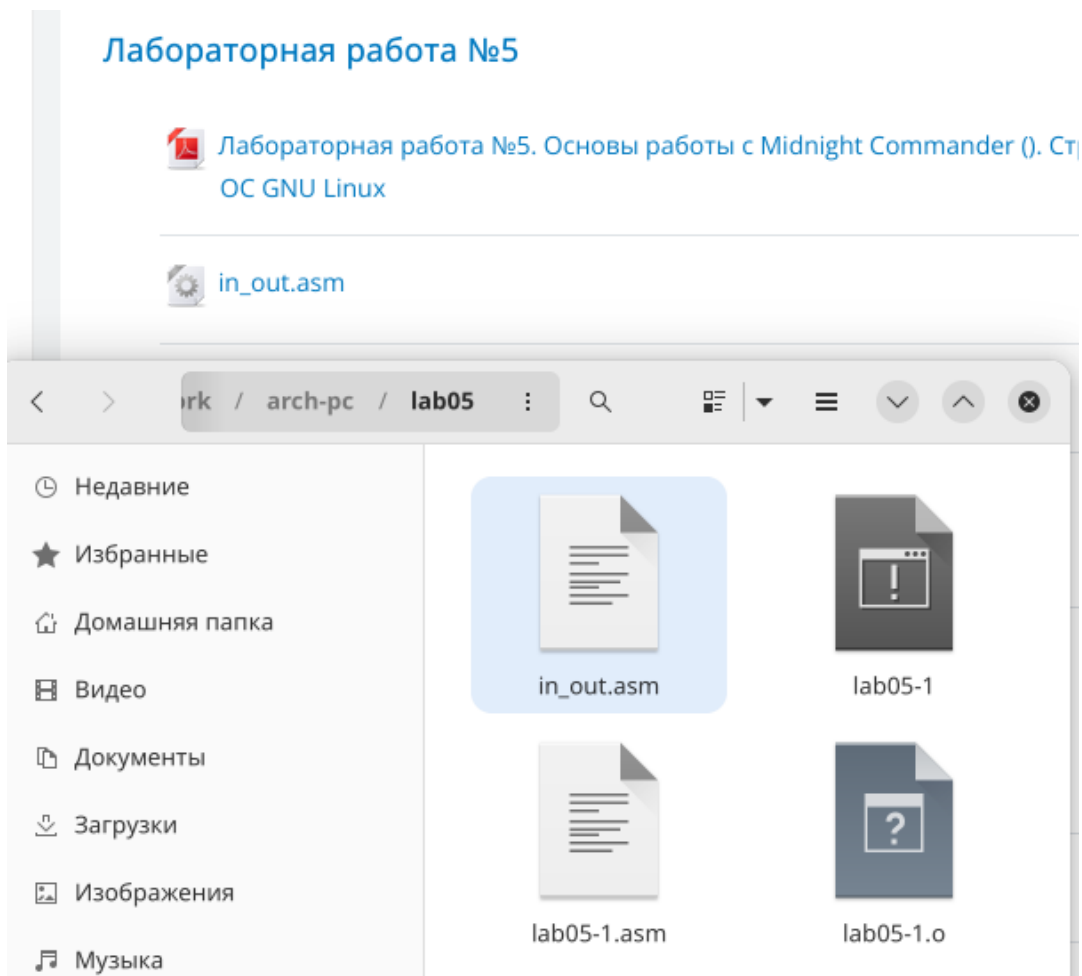


Рис. 2.19: Проверяем корректность выполненной программы

Рис. 5.7. Окно Midnight Commander. Копирование файла

2.2.3 С помощью функциональной клавиши F6 создайте копию файла lab05-1.asm с именем lab05-2.asm. Выделите файл lab05-1.asm, нажмите клавишу F6, введите имя файла lab05-2.asm и нажмите клавишу Enter (рис.2.20).

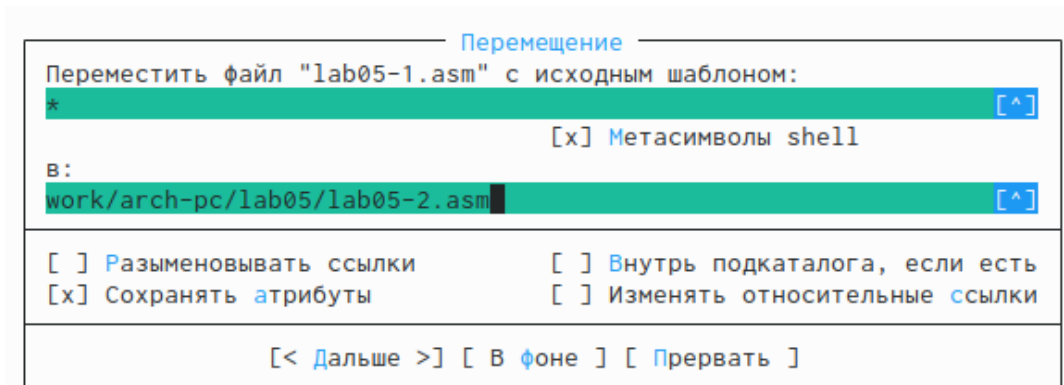


Рис. 2.20: С помощью клавиши F6 перемещаем файл lab05-1.asm в каталог lab05 с новым именем

Клавиша F6 - это клавиша для переноса файла или группы файлов из каталога, отображаемого в первой панели, в каталог, отображаемый на второй панели. Так же, возможно переписывать название файла во второй панели.

2.2.4 Исправьте текст программы в файле lab05-2.asm с использованием подпрограмм из внешнего файла in_out.asm (используйте подпрограммы sprintLF, sread и quit) в соответствии с листингом 5.2. Создайте исполняемый файл и проверьте его работу (рис.2.21).

Листинг 5.2. Программа вывода сообщения на экран и ввода строки с клавиатуры с использованием файла in_out.asm

```

lab05-2.asm      [-M--] 15 L:[ 1+12 13/ 20] *(212 / 282b) 0010 0x00A
#include 'in_out.asm'
SECTION .data
msg: DB "Введите строку:",0h

SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax, msg
call sprint

mov ebx, buf1
mov ecx, 80
call sread

call quit

```

Рис. 2.21: Вводим код программы

2.2.5 В файле lab5-2.asm замените подпрограмму sprintLF на sprint.

Создайте исполняемый файл и проверьте его работу. В чем разница? (рис.2.22)

```

dasadova@dk5n52 ~/work/arch-pc/lab05 $ nasm -f elf lab05-2.asm

dasadova@dk5n52 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab05-2 lab05-2.o

dasadova@dk5n52 ~/work/arch-pc/lab05 $ ./lab05-2
Введите строку: Садова Диана

dasadova@dk5n52 ~ $

```

Рис. 2.22: Редактируем и выводим работу программы в консоль

Ответ на вопрос: Мы записываем наши данные не на следующей строке, а после двоеточия

3 Теоретическое введение

4 Выполнение лабораторной работы

4.1 Задание для самостоятельной работы

4.1.1 Создайте копию файла lab5-1.asm. Внесите изменения в программу (без использования внешнего файла in_out.asm), так чтобы она работала по следующему алгоритму: (рис.2.23), (рис.2.24)

- вывести приглашение типа “Введите строку:”;
- ввести строку с клавиатуры;
- вывести введенную строку на экран.



Рис. 4.1: Создаем копию файла lab5-1.asm с именем lab5-1_1.asm

```

lab05-1_1.asm  [----]  0 L: [ 1+ 8  9/ 27] *(132 / 409b) 0095 0x05F
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80
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
    mov edx, 80
    int 80h
    mov eax,4
    mov ebx,1
    mov ecx,buf1
    mov edx,buf1
    int 80h
    mov eax,1
    mov ebx,0
    int 80h

```

Рис. 4.2: Вводим код программы

4.1.2 Получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите свою фамилию (рис.2.25).

```

dasadova@dk5n52 ~/work/arch-pc/lab05 $ ./lab05-1_1
Введите строку:
Садова
Садова

```

Рис. 4.3: Проверяем корректность выполненной программы

4.1.3 Создайте копию файла lab5-2.asm. Исправьте текст программы с использование подпрограмм из внешнего файла in_out.asm, так чтобы она работала по следующему алгоритму: (рис.2.26), (рис.2.27)

- вывести приглашение типа “Введите строку:”;
- ввести строку с клавиатуры;
- вывести введенную строку на экран.

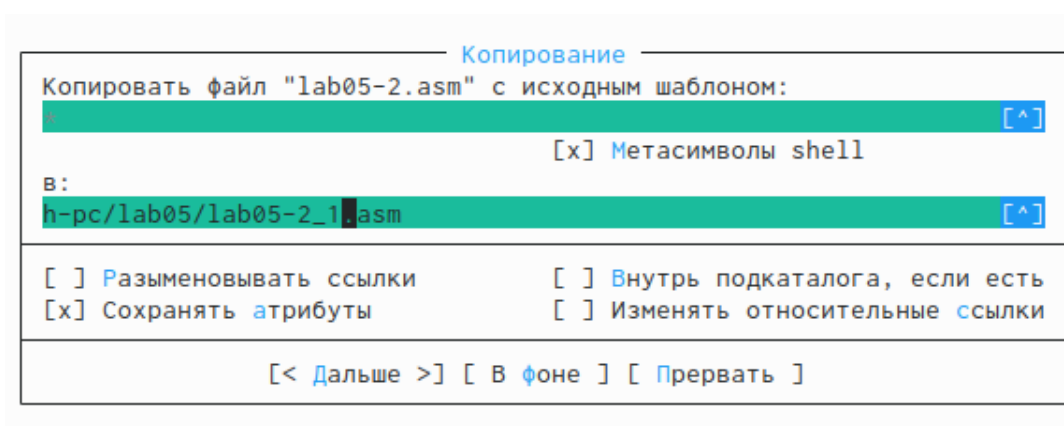


Рис. 4.4: Создаем копию файла lab5-2.asm с именем lab5-2_1.asm

```

lab05-2_1.asm      [-M--] 15 L:[ 1+11 12/ 18] *(213 / 299b) 0010 0x00A
#include "in_out.asm"
SECTION .data
msg: DB "Введите строку: ",0h
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
    mov eax, msg
    call sprint
    mov ecx, buf1
    mov edx, 80
    call sread
    mov eax, 4
    mov ebx, 1
    mov ecx, buf1
    int 80h
    call quit

```

Рис. 4.5: Вводим код программы

4.1.4 Создайте исполняемый файл и проверьте его работу (рис.2.28).

```

dasadova@dk5n52 ~/work/arch-pc/lab05 $ nasm -f elf lab05-2_1.asm
dasadova@dk5n52 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab05-2_1 lab05-2_1.o
dasadova@dk5n52 ~/work/arch-pc/lab05 $ ./lab05-2_1
Введите строку: Садова
Садова
dasadova@dk5n52 ~ $

```

Рис. 4.6: Проверяем корректность выполненной программы

5 Выводы

Приобретены практические навыки работы в Midnight Commander. Освоены инструкции языка ассемблера `mov` и `int`.

Список литературы