

Отчет по лабораторной работе №5

Простейший вариант

Чигладзе Майя Владиславовна

Содержание

1	Цель работы	5
2	Порядок выполнения лабораторной работы	6
2.1	Задание 1	6
2.2	Задание 2	6
2.3	Задание 3	7
2.4	Задание 4	8
2.5	Задание 5	8
2.6	Задание 6	9
2.7	Задание 7	10
2.8	Задание 8	11
3	Подключение внешнего файла in_out.asm	14
3.1	Задание 9	14
3.2	Задание 10	14
3.3	Задание 11	15
3.4	Задание 12	16
3.5	Задание 13	17
4	Задание для самостоятельной работы	20
4.1	Задание 1	20
4.2	Задание 2	21
4.3	Задание 3	22
4.4	Задание 4	22
5	Выводы	24
	Список литературы	25

Список иллюстраций

2.1	Открываем Midnight Commander	6
2.2	Переход в другой каталог	7
2.3	Создание каталога	7
2.4	Переход в созданный каталог	8
2.5	Создание файла	8
2.6	Переход во встроенный редактор	9
2.7	Ввод текста программы	10
2.8	Открытие файла	11
2.9	Оттранслирование текста программы	11
2.10	Выполнение компоновки	12
2.11	Запуск файла	13
2.12	Ввод с клавиатуры	13
3.1	Скачивание файла с ТУИС	14
3.2	Копирование файла	15
3.3	Создание копии файла	15
3.4	Исправление текста	16
3.5	Проверка работы	17
3.6	Изменение файла	18
3.7	Запуск файла	19
4.1	Открытие копии файла	21
4.2	Конвертирование и запуск программы	22
4.3	Создании копии файла	22
4.4	Проверяем работу	23

Список таблиц

1 Цель работы

Изучение инструкций языка ассемблера `mov` и `int` в процессе освоения практических навыков работы с Midnight Commander.

2 Порядок выполнения лабораторной работы

2.1 Задание 1

Откроем Midnight Commander (рис. [2.1])

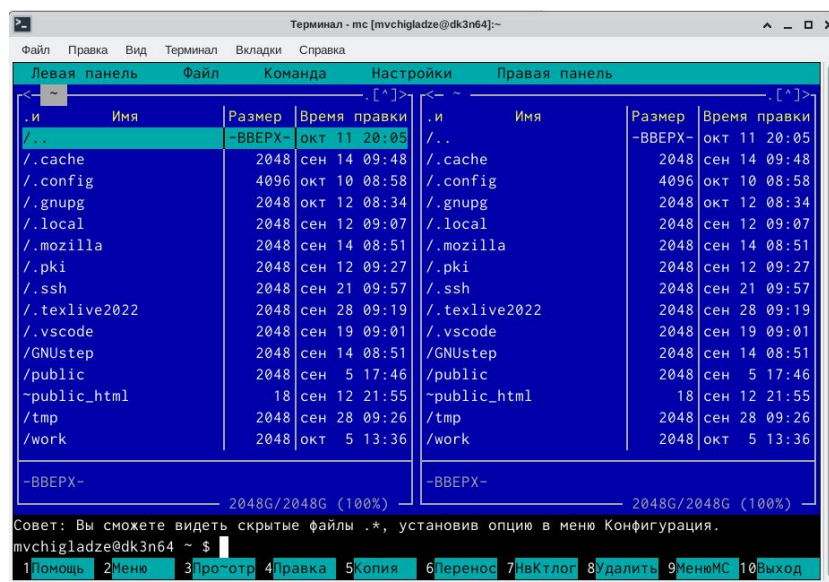


Рис. 2.1: Открываем Midnight Commander

2.2 Задание 2

Пользуясь клавишами **⌘**, **⌘** и **Enter** перейдем в каталог `~/work/arch-pc`, созданный при выполнении лабораторной работы №4 (рис. [2.2])

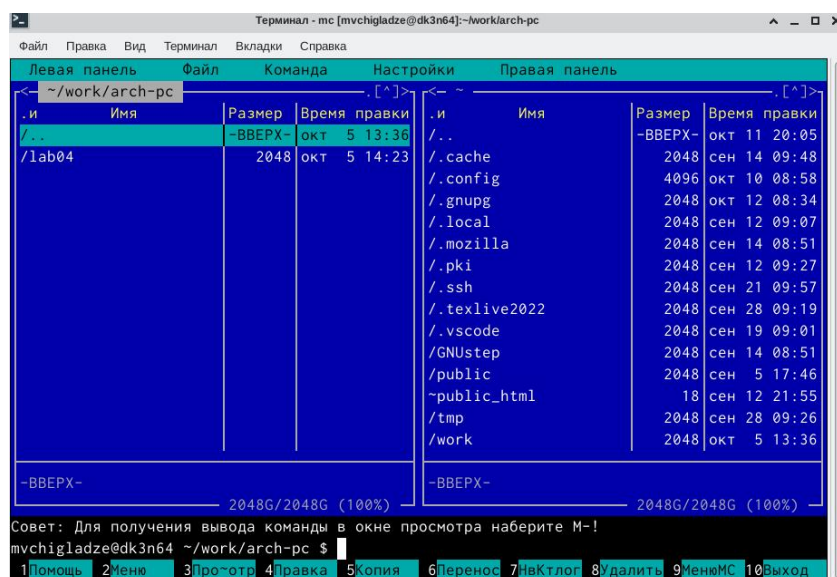


Рис. 2.2: Переход в другой каталог

2.3 Задание 3

С помощью функциональной клавиши F7 создадим папку lab05 (рис. [2.3]) и перейдем в созданный каталог (рис. [2.4])

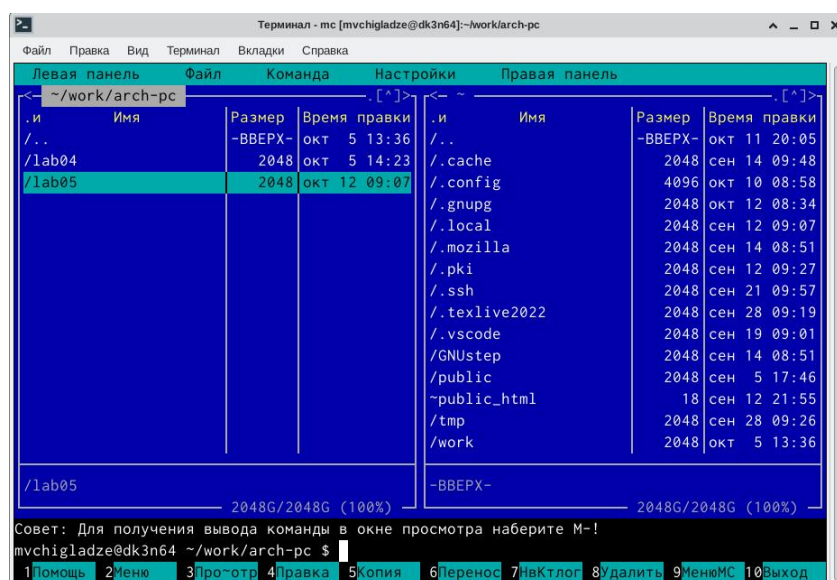


Рис. 2.3: Создание каталога

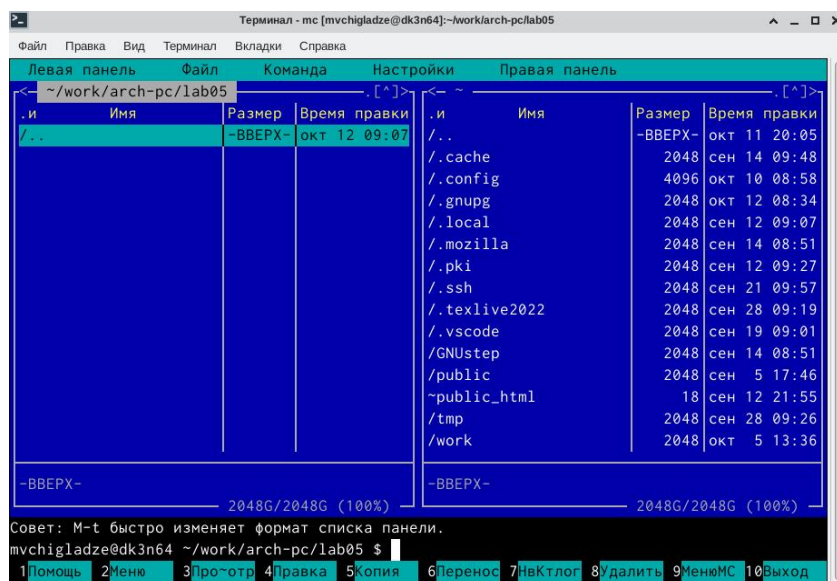


Рис. 2.4: Переход в созданный каталог

2.4 Задание 4

Пользуясь строкой ввода и командой `touch` создадим файл `lab5-1.asm` (рис. [2.5])

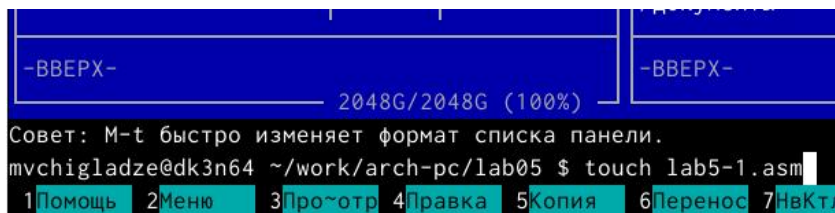


Рис. 2.5: Создание файла

2.5 Задание 5

С помощью функциональной клавиши `F4` откроем файл `lab5-1.asm` для редактирования во встроенном редакторе (рис. [2.6])

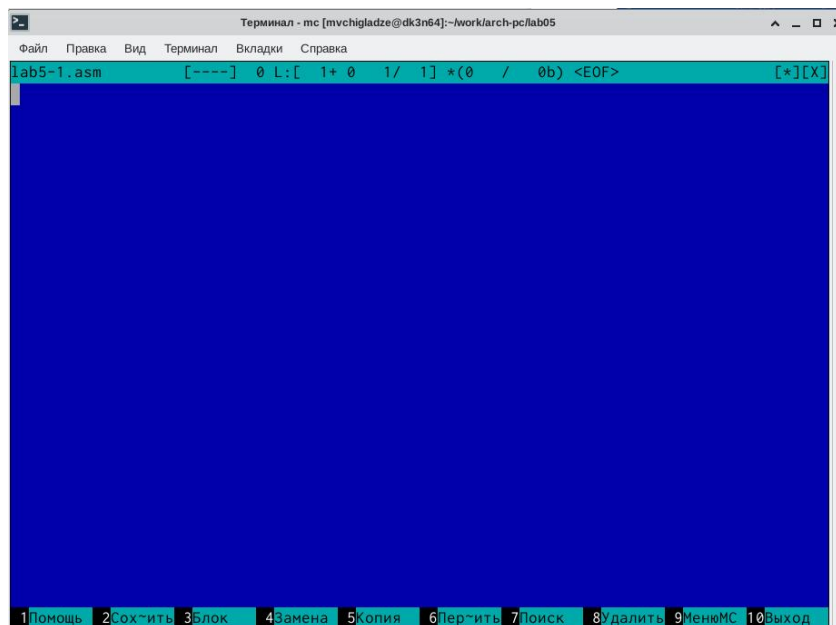
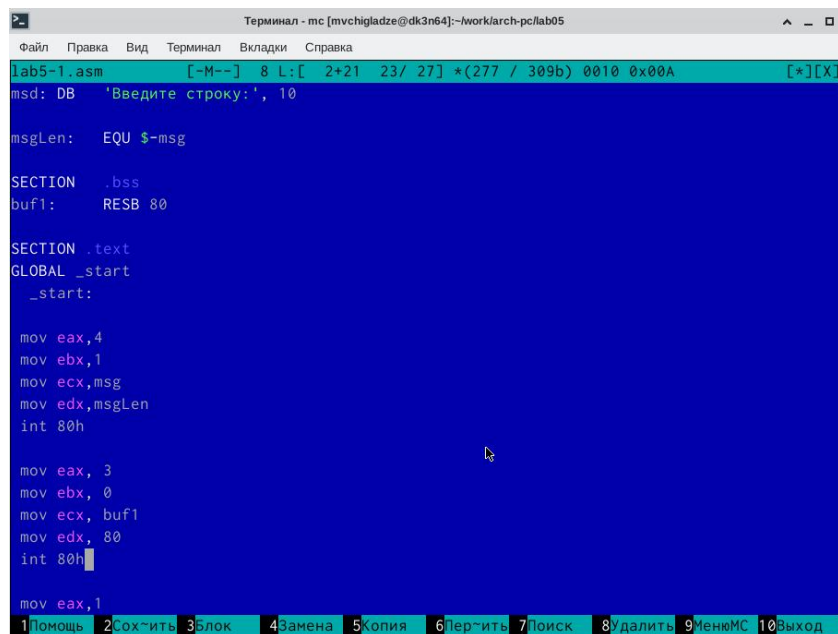


Рис. 2.6: Переход во встроенный редактор

2.6 Задание 6

Введем текст программы из листинга, сохраним изменения и закроем файл (рис. [2.7])



```
lab5-1.asm [-M--] 8 L: [ 2+21 23/ 27] *(277 / 309b) 0010 0x00A [*][X]
msd: 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

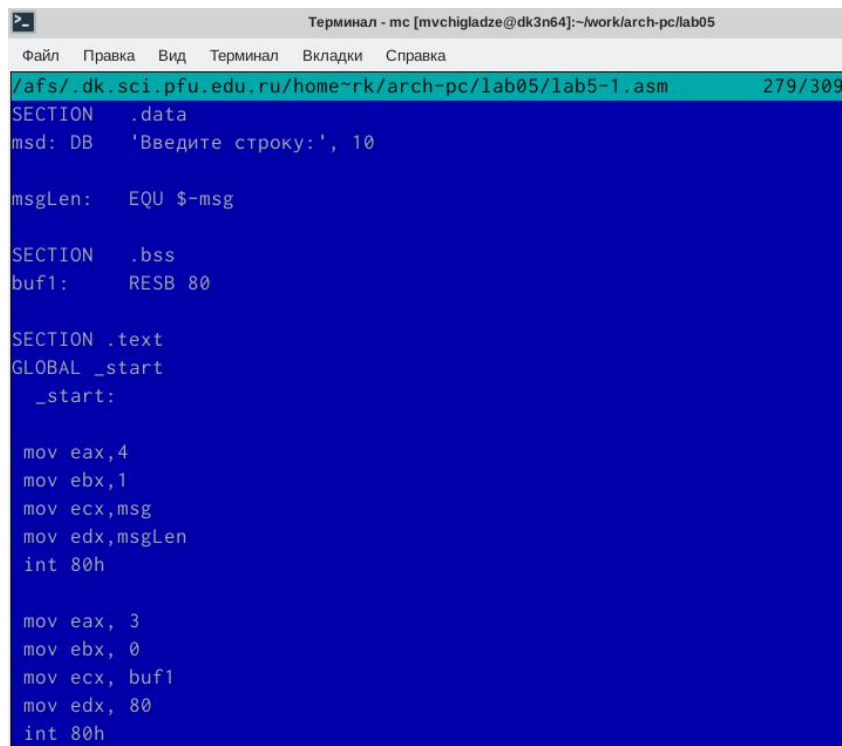
mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h

mov eax, 1
```

Рис. 2.7: Ввод текста программы

2.7 Задание 7

С помощью функциональной клавиши F3 откроем файл lab5-1.asm для просмотра. Убедимся, что файл содержит текст программы (рис. [2.8])



```
Терминал - mc [mvchigladze@dk3n64]:~/work/arch-pc/lab05
Файл  Правка  Вид  Терминал  Вкладки  Справка
/afs/.dk.sci.pfu.edu.ru/home~rk/arch-pc/lab05/lab5-1.asm 279/309
SECTION .data
msd: 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
```

Рис. 2.8: Открытие файла

2.8 Задание 8

Оттранслируем текст программы lab5-1.asm в объектный файл (рис. [2.9]). Выполним компоновку объектного файла (рис. [2.10]) и запустим получившийся исполняемый файл (рис. [2.11]). Программа выводит строку 'Введите строку:' и ожидает ввода с клавиатуры. На запрос введу свои ФИО (рис. [2.12])

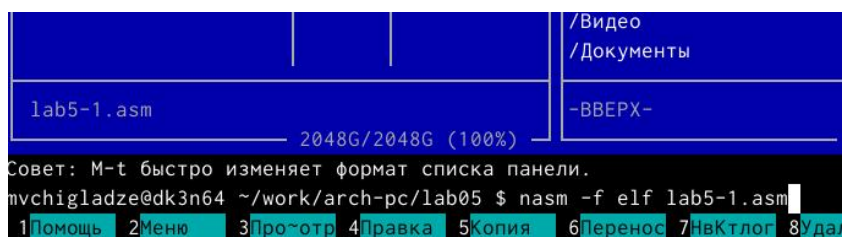


Рис. 2.9: Оттранслирование текста программы

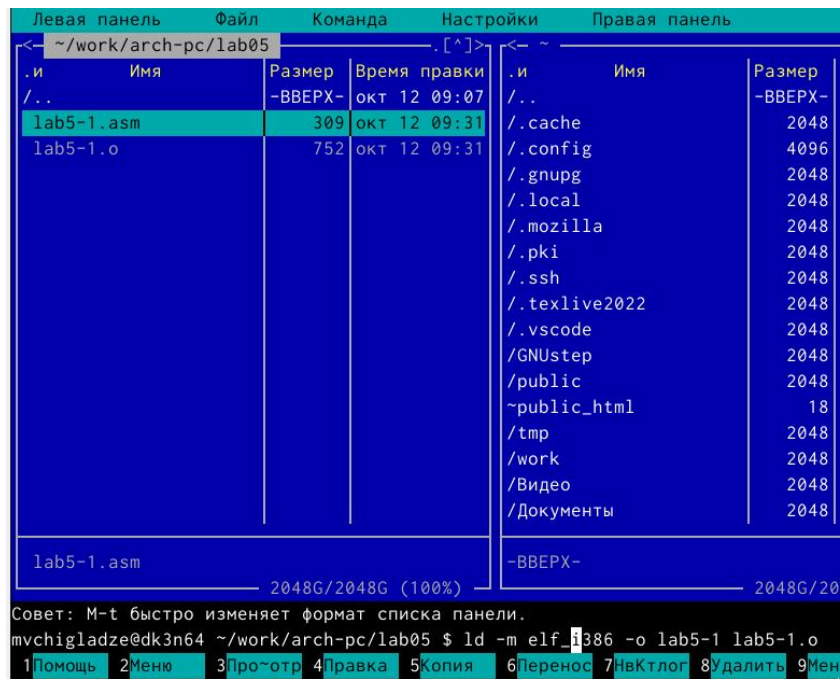


Рис. 2.10: Выполнение компоновки

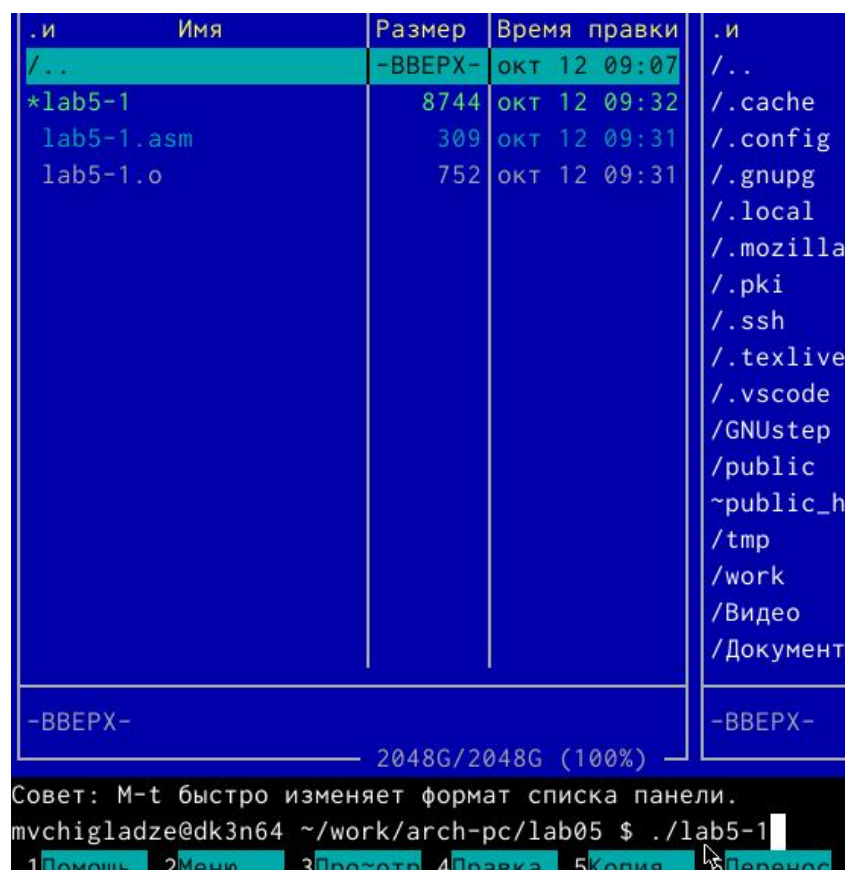


Рис. 2.11: Запуск файла

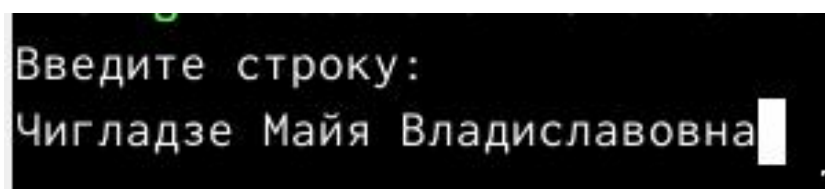


Рис. 2.12: Ввод с клавиатуры

3 Подключение внешнего файла

in_out.asm

3.1 Задание 9

Скачаем файл in_out.asm со страницы курса в ТУИС (рис. [3.1])



Рис. 3.1: Скачивание файла с ТУИС

3.2 Задание 10

Копируем подключаемый файл in_out.asm в каталог lab05 (рис. [3.2])

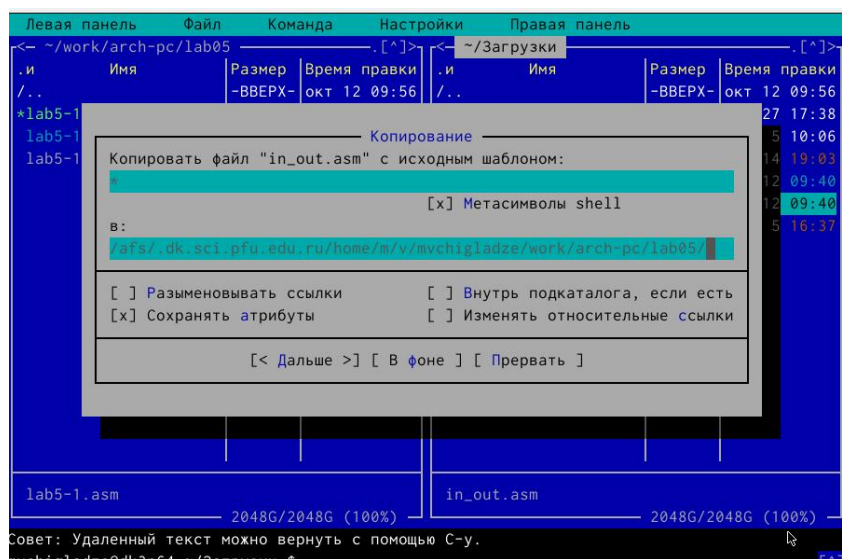


Рис. 3.2: Копирование файла

3.3 Задание 11

С помощью функциональной клавиши F6 создам копию файла lab5-1.asm с именем lab5-2.asm (рис. [3.3])

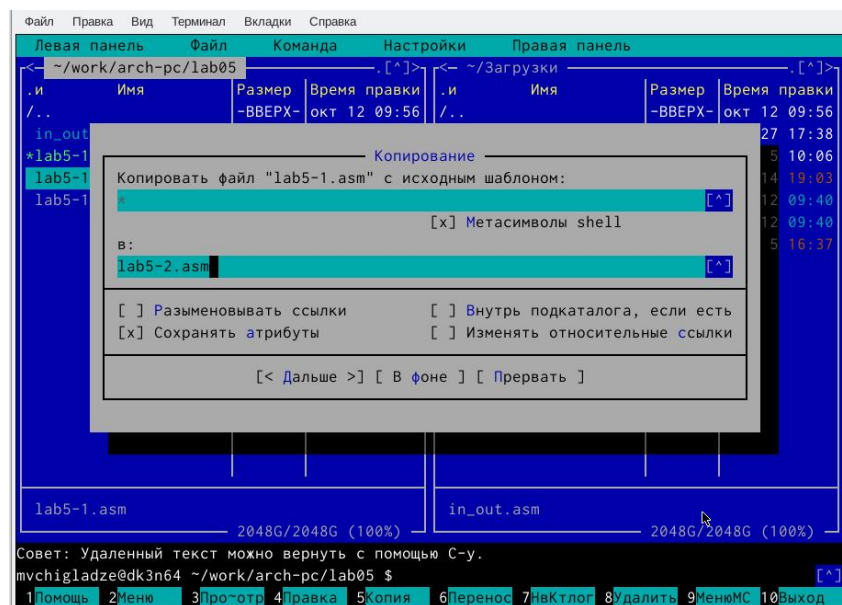


Рис. 3.3: Создание копии файла

3.4 Задание 12

Исправим текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in_out.asm в соответствии с листингом (рис. [3.4]). Создам исполняемый файл и проверю его работу (рис. [3.5])

```
7ats7.uk.sc1.ptu.edu.ru/home/m/v/mvcp1g1k  
%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 sprintLF  
  
    mov eax, buf1  
    mov edx, 80  
  
    call sread  
  
    call quit
```

Рис. 3.4: Исправление текста


```
mvchigladze@dk3n64 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
mvchigladze@dk3n64 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
mvchigladze@dk3n64 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку:
Чигладзе Майя Владиславовна
mvchigladze@dk3n64 ~/work/arch-pc/lab05 $
```

Рис. 3.5: Проверка работы

3.5 Задание 13

В файле lab5-2.asm заменим подпрограмму sprintLF на sprint (рис. [3.6]). Создадим исполняемый файл и проверим его работу (рис. [3.7]). Разница состоит в отступе после заранее написанной фразы и нашим вводом

```
msg: DB    'Введите строку:

SECTION    .bss
buf1:      RESB 80

SECTION .text
GLOBAL _start
_start:

    mov eax,msg
    call sprint

    mov eax, buf1
    mov edx, 80

    call sread
```

Рис. 3.6: Изменение файла

```
mvchigladze@dk3n64 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку:Чигладзе Майя
mvchigladze@dk3n64 ~/work/arch-pc/lab05 $
```

Рис. 3.7: Запуск файла

4 Задание для самостоятельной работы

4.1 Задание 1

Создадим копию файла lab5-1.asm и назовем lab5-3.asm (рис. [4.1]). Внесем изменения в программу (без использования внешнего файла in_out.asm), так чтобы она работала по определенному алгоритму

```
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, 80
int 80h

mov eax, 1
mov ebx, 0
int 80h
```

1Помощь 2Сохранить 3Блок 4Замена

Рис. 4.1: Открытие копии файла

4.2 Задание 2

Получим исполняемый файл и проверим его работу. На приглашение ввести строку введу свою фамилию (рис. [4.2]).

```

mvchigladze@dk3n64 ~/work/arch-pc/lab05 $ nasm -f elf lab5-3.asm
mvchigladze@dk3n64 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-3 lab5-3.o
mvchigladze@dk3n64 ~/work/arch-pc/lab05 $ ./lab5-3
Введите строку:
Чигладзе
Чигладзе
mvchigladze@dk3n64 ~/work/arch-pc/lab05 $

```

Рис. 4.2: Конвертирование и запуск программы

4.3 Задание 3

Создадим копию файла lab5-2.asm и назовем lab5-4.asm. Исправим текст программы с использованием подпрограмм из внешнего файла in_out.asm, так чтобы она работала по заданному алгоритму (рис. [4.3]).

```

mov eax,msg
call sprintf

mov ecx, buf1
mov edx, 80
call sread

mov eax, buf1
call sprintf

call quit

```

Рис. 4.3: Создании копии файла

4.4 Задание 4

Создадим исполняемый файл и проверим его работу (рис. [4.4]).

```
mvchigladze@dk3n64 ~/work/arch-pc/lab05 $ nasm -f elf lab5-4.asm
mvchigladze@dk3n64 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-4 lab5-4.o
mvchigladze@dk3n64 ~/work/arch-pc/lab05 $ ./lab5-4
Введите строку:
Чигладзе
Чигладзе
```

Рис. 4.4: Проверяем работу

5 Выводы

В ходе лабораторной работы, я изучила инструкции языка ассемблера `mov` и `int` в процессе освоения практических навыков работы с Midnight Commander.

Список литературы