Отчет по лабораторной работе № 5

Дисциплина: Архитектура компьютера

Дроздова Дарья Игоревна

Содержание

1 Цель работы **3**

2 Выполнение лабораторной работы **4**

3 Выводы **12**

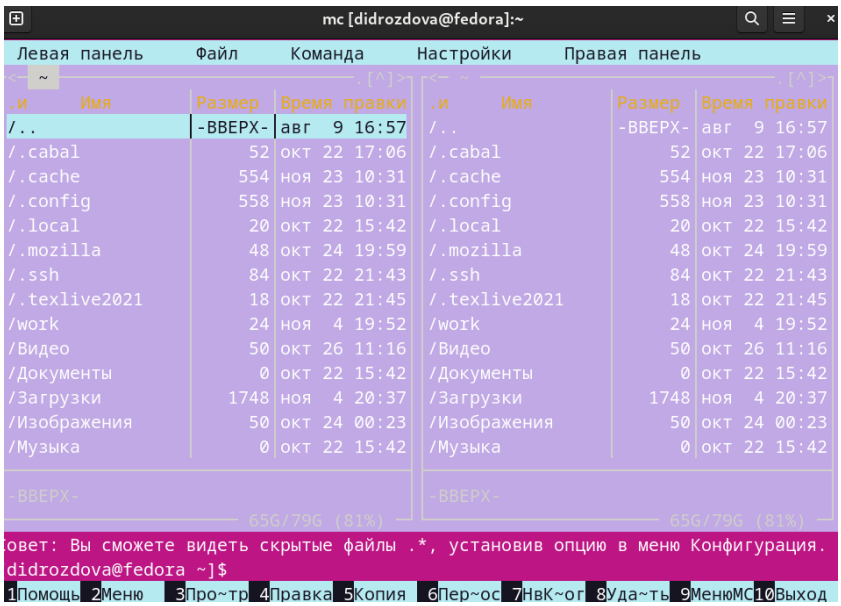
1 Цель работы

Целью данной лабораторной работы является освоение языка ассемблера mov и int и приобретение практических навыков работы в Midnight Commander.

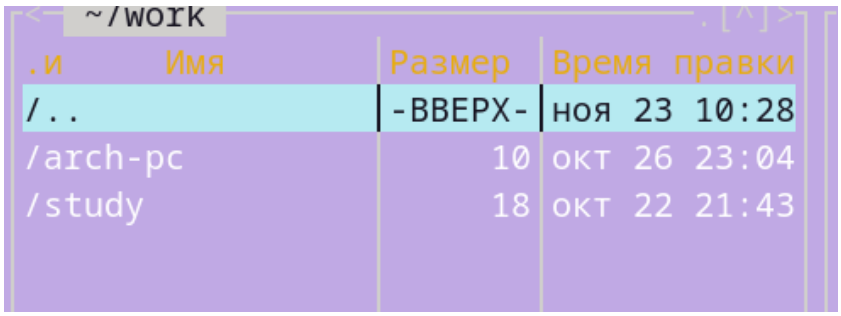
2 Выполнение лабораторной работы

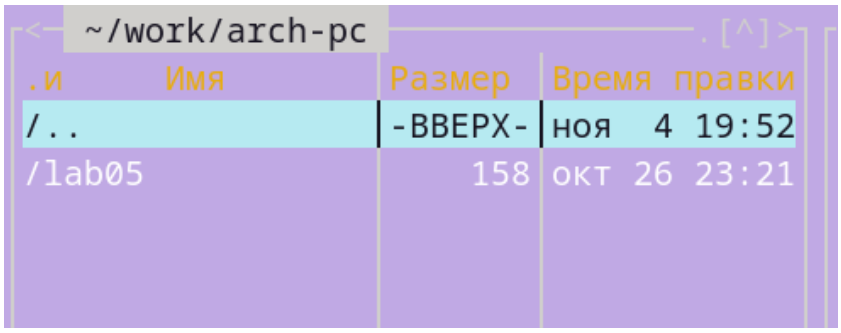
1. Порядок выполнения лабораторной работы

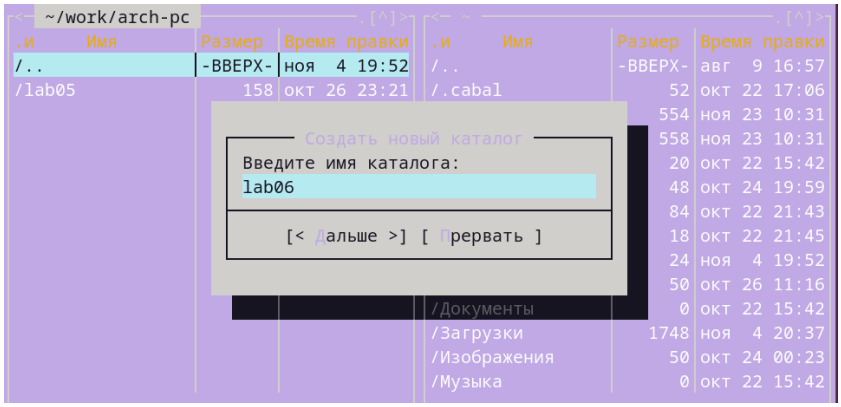
   - С помощью команды mc открываем Midnight Commander и переходим в каталог *~/work/arch-pc* созданный при выполнении лабораторной работы №4:



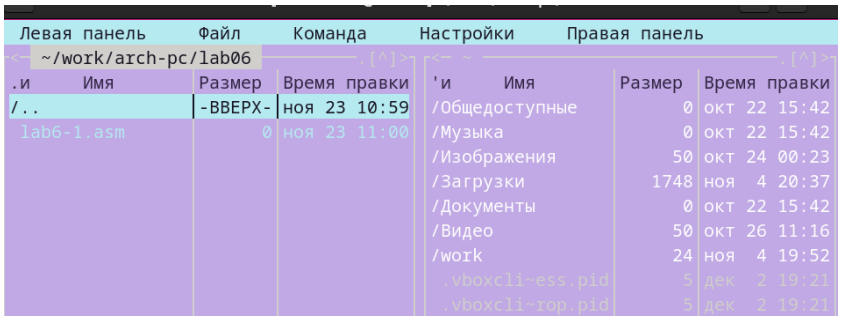
   - Создаем папку lab06 с помощью командной клавиши F7 и переходим в нее после создания:



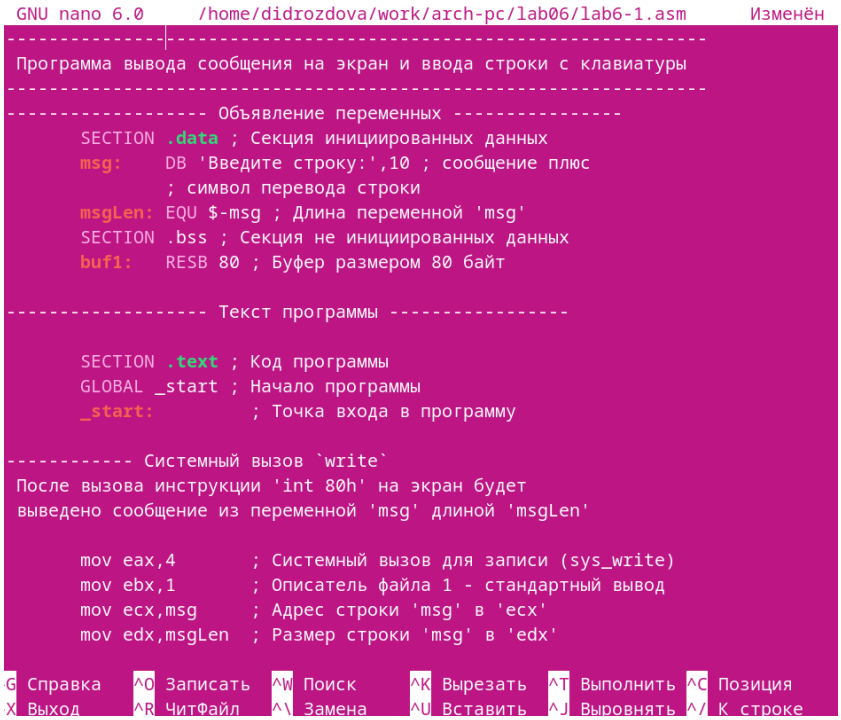




- Пользуясь строкой ввода и командой touch, создаем файл lab6-1.asm:



- Открываем только что созданный файл для редактирования во встроенном редакторе *mcedit* и вводим текст программы, представленный в листинге документа, приложенного к данной лабораторной работе:



- Сохраняем файл lab6-1.asm, закрываем его, а после открываем, чтобы убедиться, что текст программы сохранился.
- Далее оттранслируем текст программы lab6-1.asm в объектный файл, сделаем компоновку объектного файла и запустим получившийся исполняемый файл.

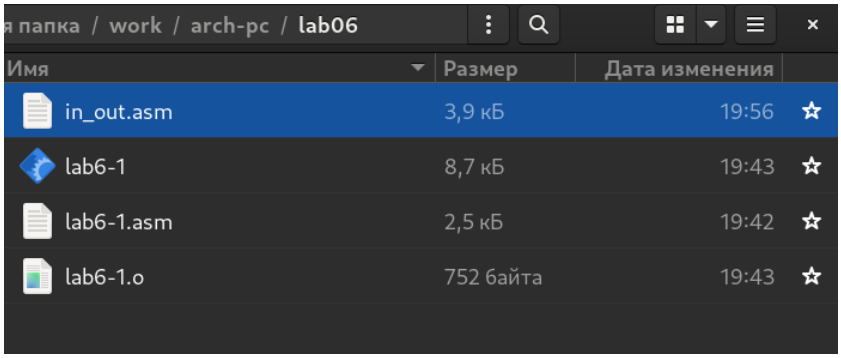
```
[didrozdova@fedora lab06]$ nasm -f elf lab6-1.asm
[didrozdova@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[didrozdova@fedora lab06]$ ./lab6-1
Введите строку:
Дроздова Дарья Игоревна
[didrozdova@fedora lab06]$
```

Программа выводит строку: ‘Введите строку:’, и ожидает ввода с клавиатуры. На запрос вводим собственные ФИО.

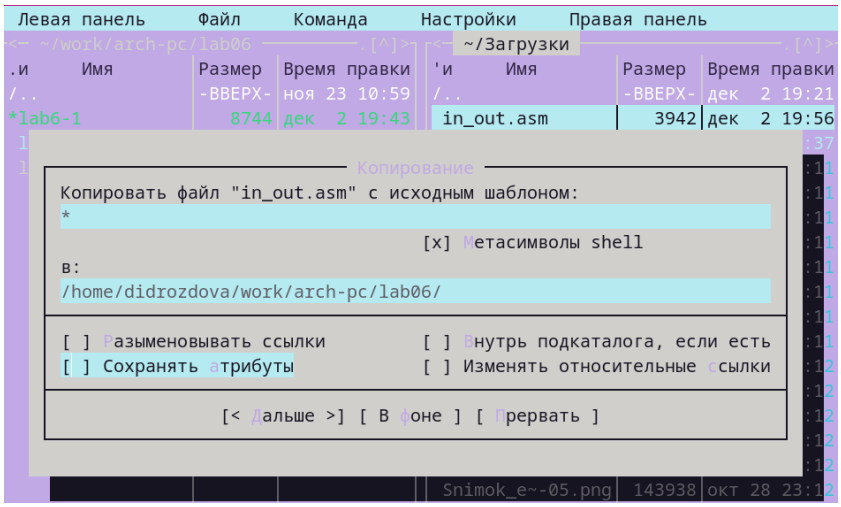
2. Подключение внешнего файла in_out.asm

- Скачиваем файл in_out.asm со страницы курса в ТУИС.
- Сохраняем подключаемый файл in_out.asm в одном каталоге с файлом программы, в которой он используется.

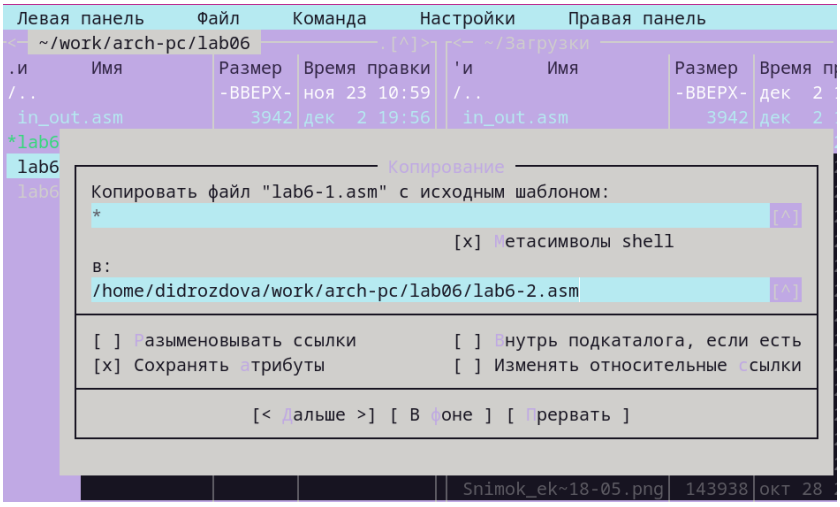
- В одной из панелей mc открываем каталог с файлом lab6-1.asm, в другой - каталог со скаченным файлом in_out.asm:



Копируем файл in_out.asm в каталог с файлом lab6-1.asm с помощью функциональной клавиши F5:



- С помощью функциональной клавиши F6 создаем копию файла lab6-1.asm с именем lab6-2.asm:



- Исправляем текст программы в файле lab6-2.asm с использование подпрограмм из внешнего файла in_out.asm в соответствии с листингом. Создаем исполняемый файл и проверяем его работу:

```
GNU nano 6.0        /home/didrozdova/work/arch-pc/lab06/lab6-2.asm        Изм
;--------------------------------------------------------------------
; 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
```

```
didrozdova@fedora lab06]$ nasm -f elf lab6-2.asm
didrozdova@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
didrozdova@fedora lab06]$ ./lab6-2
ведите строку:
роздова Дарья Игоревна
didrozdova@fedora lab06]$
```

- В файле lab6-2.asm заменяем подпрограмму sprintLF на sprint, создаем исполняемый файл и проверяем его работу:

```
 GNU nano 6.0        /home/didrozdova/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 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 sprint ; 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
```

```
[didrozdova@fedora lab06]$ nasm -f elf lab6-2.asm
[didrozdova@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[didrozdova@fedora lab06]$ ./lab6-2
Введите строку: Дроздова Дарья Игоревна
```

Заметим разницу между командами sprintLF и sprint: в первом случае ввод осуществляется с новой строки, а во втором - сразу после выводимого текста.

3. Задание для самостоятельной работы

    - Делаем копию файла lab6-1.asm, вносим изменения в программу, без использования команд из файла in_out.asm, чтобы программа работала по следующщему алгоритму:

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

```
;------------ 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

        mov eax, 4
        mov ebx, 1
        mov ecx, buf1
        mov edx, 80
        int 80h

^G Справка    ^O Записать   ^W Поиск      ^K Вырезать   ^T Выполнить  ^C Позиция
^X Выход      ^R ЧитФайл    ^\ Замена     ^U Вставить   ^J Выровнять  ^/ К строке
```

Создаем исполняемый файл и проверяем его работу:

```
[didrozdova@fedora lab06]$ nasm -f elf lab6-1_copy.asm
[didrozdova@fedora lab06]$ ld -m elf_i386 -o lab6-1_copy lab6-1_copy.o
[didrozdova@fedora lab06]$ ./lab6-1_copy
Введите строку:
Дроздова Дарья Игоревна
Дроздова Дарья Игоревна
[didrozdova@fedora lab06]$
```

```
[didrozdova@fedora lab06]$ ./lab6-1_copy
Введите строку:
Дроздова
Дроздова
[didrozdova@fedora lab06]$
```

- Создаем копию файла lab6-2.asm и исправляем текст программы с использование 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.

```
;-------------------------------------------------------------------
; 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 sprint ; 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
        mov eax, buf1
        call sprint
        call quit ; вызов подпрограммы завершения
```

Создаем исполняемый файл и проверяем его работу:

```
[didrozdova@fedora lab06]$ nasm -f elf lab6-2_copy.asm
[didrozdova@fedora lab06]$ ld -m elf_i386 -o lab6-2_copy lab6-2_copy.o
[didrozdova@fedora lab06]$ ./lab6-2_copy
Введите строку: Дроздова
Дроздова
[didrozdova@fedora lab06]$
```

3 Выводы

В ходе выполнения практических заданий “Лабораторной работы №5” я приобрела практические навыки работы в Midnight Commander и освоила инструкции языка ассемблера mov и int.