Отчёт по лабораторной работе №5

Дисциплина: Архитектура компьтеров

Карпова Анастасия Александровна

Содержание

Список иллюстраций

1 Цель работы

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

2 Задание

1. Основы работы с mc
2. Структура программы на языке ассемблера NASM
3. Подключение внешнего файла
4. Выполнение заданий для самостоятельной работы

3 Теоретическое введение

Midnight Commander (или просто mc) — это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управлению файловой системой, т.е. mc является файловым менеджером. Midnight Commander позволяет сделать работу с файлами более удобной и наглядной. Программа на языке ассемблера NASM, как правило, состоит из трёх секций: секция кода программы (SECTION .text), секция инициированных (известных во время компиляции) данных (SECTION .data) и секция неинициализированных данных (тех, под которые во время компиляции только отводится память, а значение присваивается в ходе выполнения программы) (SECTION .bss). Для объявления инициированных данных в секции .data используются директивы DB, DW, DD, DQ и DT, которые резервируют память и указывают, какие значения должны храниться в этой памяти:

DB (define byte) — определяет переменную размером в 1 байт; DW (define word) — определяет переменную размеров в 2 байта (слово); DD (define double word) — определяет переменную размером в 4 байта (двойное слово); DQ (define quad word) — определяет переменную размером в 8 байт (учетве- рённое слово); DT (define ten bytes) — определяет переменную размером в 10 байт. Директивы используются для объявления простых переменных и для объявления массивов. Для определения строк принято использовать директиву DB в связи с особенностями хранения данных в оперативной памяти. Инструкция языка ассемблера mov предназначена для дублирования данных источника в приёмнике. mov dst,src Здесь операнд dst — приёмник, а src — источник. В качестве операнда могут высту-

пать регистры (register), ячейки памяти (memory) и непосредственные значения (const). Инструкция языка ассемблера intпредназначена для вызова прерывания с указанным номером. int n Здесь n — номер прерывания, принадлежащий диапазону 0–255. При программировании в Linux с использованием вызовов ядра sys_calls n=80h (принято задавать в шестнадцатеричной системе счисления).

4 Выполнение лабораторной работы

Основы работы с mc

Открываю Midhight Commander, введя mc в терминал (рис. 4.1).

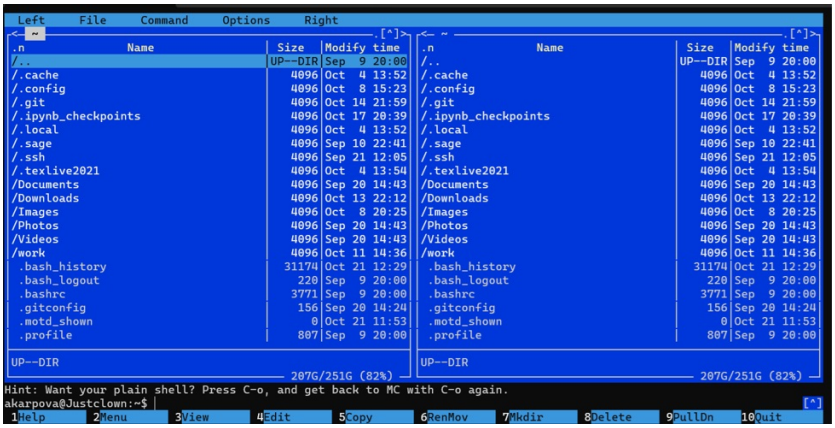
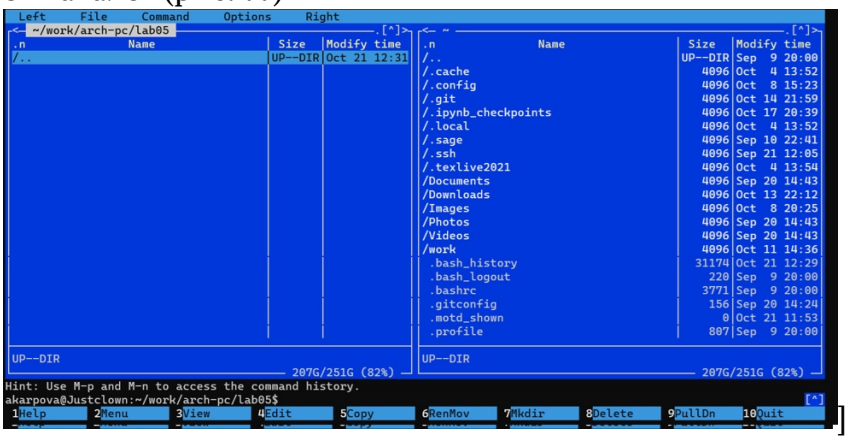
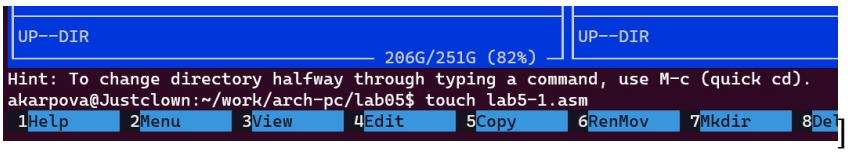


Рис. 4.1: Открытие mc

Перехожу в каталог ~/work/arch-pc и создаю в нем каталог lab05, перехожу в этот каталог (рис. **??**)



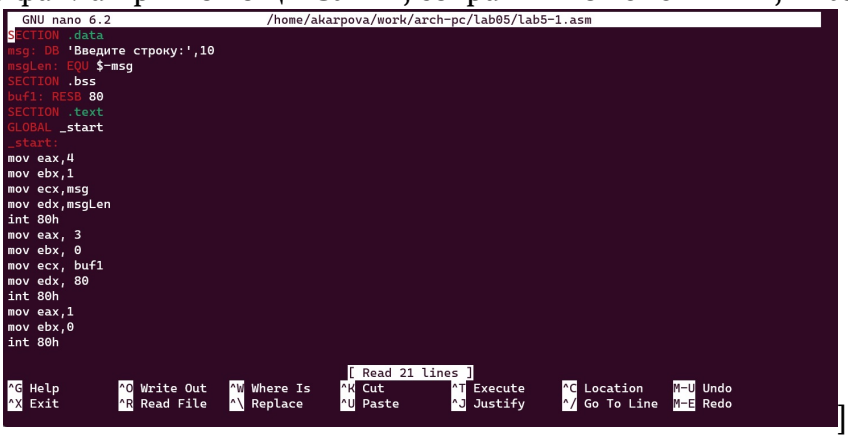
Создаю файл lab5-1.asm при помощи команды touch (рис. **??**)



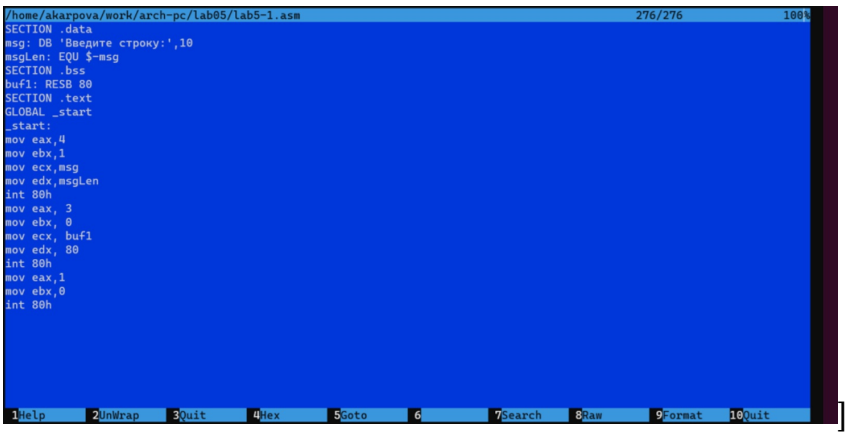
Структура программы на языке ассемблера NASM

При помощи клавиши f4 открываю файл lab5-1.asm для редактирования в nano (рис. **??**)

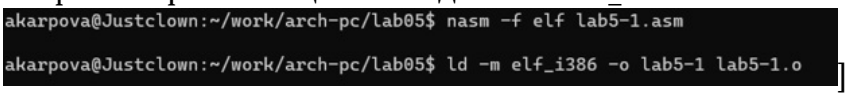
Ввожу в файл код программы для запроса строки у пользователя. Потом выхожу из файла при помощи Ctrl+X, сохранив изменения - Y, Enter. (рис. **??**)



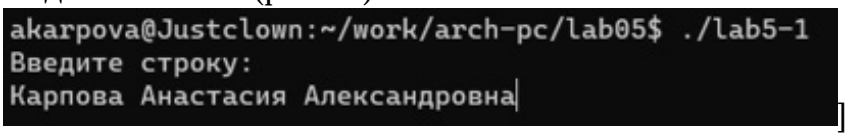
Проверяю файл на содержание кода при помощи f3(для просмотра файла) (рис. **??**)



Транслирую текст программы в объектный файл командой nasm -f elf lab5-1.asm (создался объектный файл lab5-1.o). Затем выполняю компоновку объектного файла при помощи команды ld -m elf_i386 -o lab5-1 lab5-1.o. (рис. **??**)

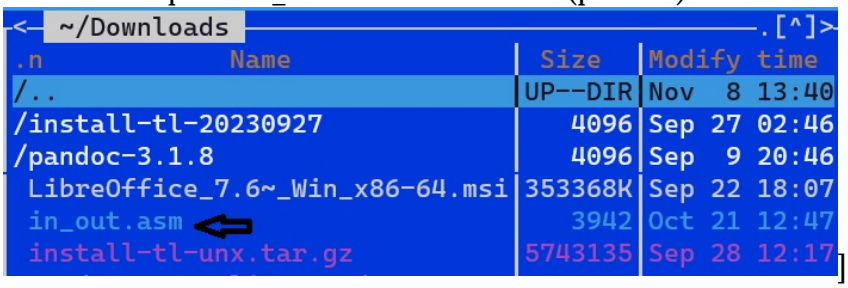


Запускаю исполняемый файл. (Программа выводит строку “Введите строку” - соответственно мы вводим своё ФИО и программа заканчивает свою работу, выведя наше ФИО (рис. **??**)

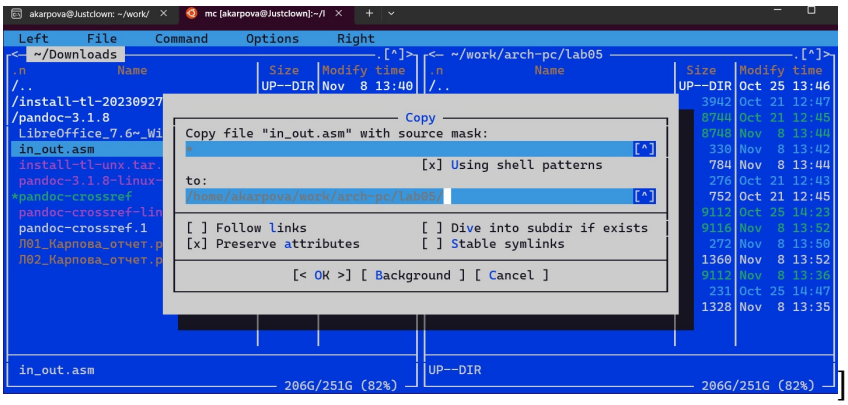


Запуск исполняемого файла

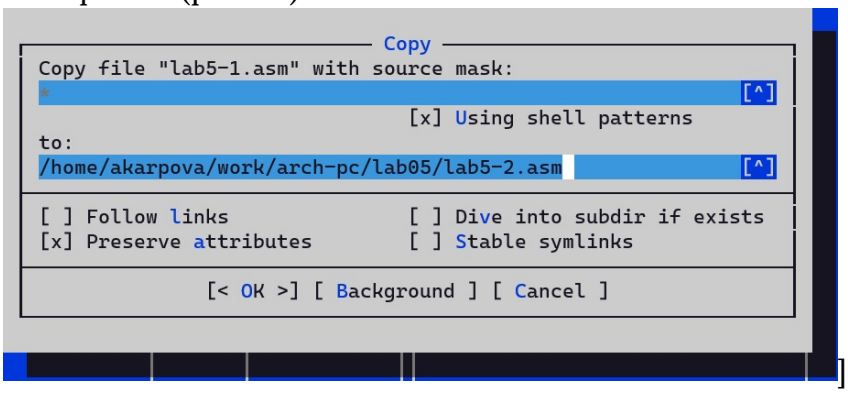
Скачиваю файл in_out.asm из ТУИСа. (рис. **??**)



С помощью функциональной клавиши f5 копирую файл in_out.asm из каталога Downloads в созданный каталог lab05. (рис. **??**)



С помощью функциональной клавиши f5 копирую файл lab5-1 в тот же каталог, но с другим именем, для этого в появившемся окне mc прописываю имя для копии файла. (рис. **??**)



Изменяю содержимое файла lab5-2.asm редакторе nano, чтобы в программе использовались подпрограммы из внешнего файла in_out.asm. (рис. **??**)

Транслирую текст программы файла в объектный файл командой nasm -f elf lab5-2.asm (Создался объектный файл lab5-2.o). Выполняю компоновку объектного файла с помощью команды ld -m elf_i386 -o lab5-2 lab5-2.o (Создался исполняемый файл lab5-2). Запускаю исполняемый файл. (рис. **??**)

```
akarpova@Justclown:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
akarpova@Justclown:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
akarpova@Justclown:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Карпова Анастасия Александровна
```

Открываю файл lab5-2.asm в редакторе в nano функциональной клавишей f4. Изменяю в нем подпрограмму sprintLF на sprint. Сохраняю изменения и открываю файл для просмотра, чтобы проверить сохранение действий. (рис. **??**)

```
 GNU nano 6.2                  /home/akarpova/work/arch-pc/lab05/lab5-2.asm *
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите строку:',0h
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprint
mov ecx, buf1
mov edx, 80
call sread
call quit
```

Снова транслирую файл, выполняю компоновку созданного объектного файла, запускаю новый исполняемый файл. (рис. **??**)

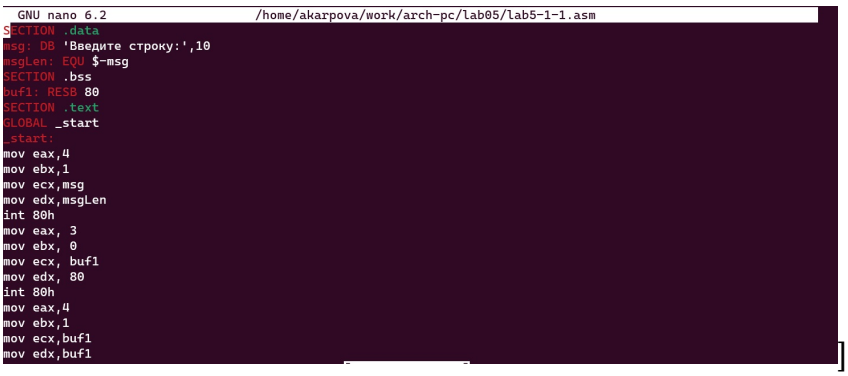
```
akarpova@Justclown:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
akarpova@Justclown:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2-2 lab5-2.o
akarpova@Justclown:~/work/arch-pc/lab05$ ./lab5-2-2
Введите строку:Карпова Анастасия Александровна
```

Выполнение заданий для самостоятельной работы

1. Создаю копию файла lab5-1.asm с именем lab5-1-1.asm с помощью клавиши f5. (рис. **??**)

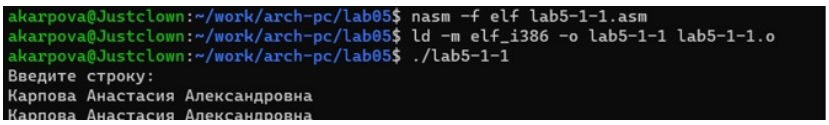
```
                              Copy
 Copy file "lab5-1.asm" with source mask:
 *                                                                [^]
                                  [x] Using shell patterns
 to:
 /home/akarpova/work/arch-pc/lab05/lab5-1-1.asm                   [^]

 [ ] Follow links                 [ ] Dive into subdir if exists
 [x] Preserve attributes          [ ] Stable symlinks

                [< OK >] [ Background ] [ Cancel ]
```

С помощью клавиши f4 открываю созданный файл для редактирования. Изменяю программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку. (рис. **??**)



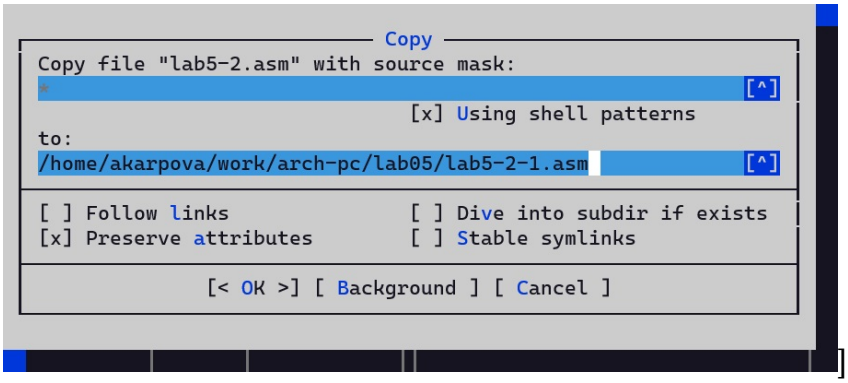
]

2. Создаю объектный файл lab5-1-1.o, отдаю его на обработку компоновщику, получаю исполняемый файл lab5-1-1, запускаю полученный исполняемый файл. Программа запрашивает ввод, ввожу свои ФИО, и программа соответственно выводит введенные мною данные (рис. **??**)



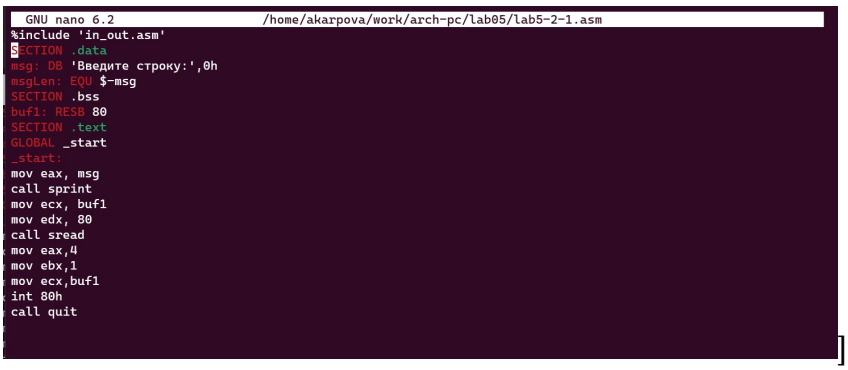
]

3. Создаю копию файла lab5-2.asm с именем lab5-2-1.asm с помощью клавиши f5. (рис. **??**)

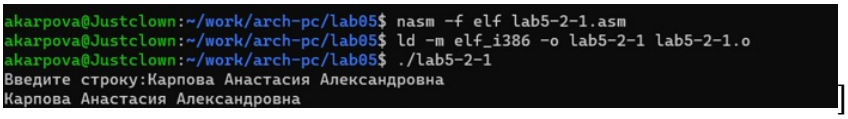


]

С помощью клавиши f4 открываю созданный файл для редактирования. Изменяю программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку. (рис. **??**)



4. Создаю объектный файл lab5-2-1.o, отдаю его на обработку компоновщику, получаю исполняемый файл lab5-2-1, запускаю полученный исполняемый файл. Программа запрашивает ввод без переноса на новую строку, ввожу свои ФИО, далее программа выводит введенные мною данные. (рис. **??**)



5 Выводы

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

6 Список литературы

1. Архитектура ЭВМ