Отчёт по лабораторной работе №5

Дисциплина: Архитектура Компьютера

Вакутайпа Милдред

Содержание

# 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. Подключение внешнего файла

# 3 Выполнение лабораторной работы

**Основы работы с mc**

Открываю Midnight Commander, введя в терминал mc:

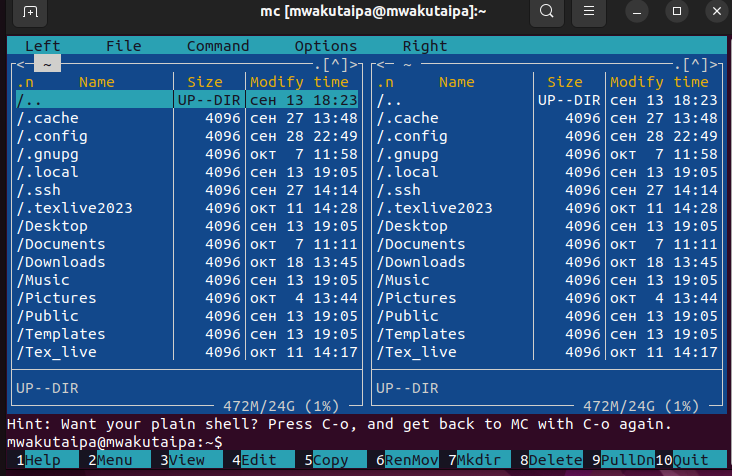


Рис 1

Перехожу в каталог ~/work/arch-pc/, используя файловый менеджер mc:

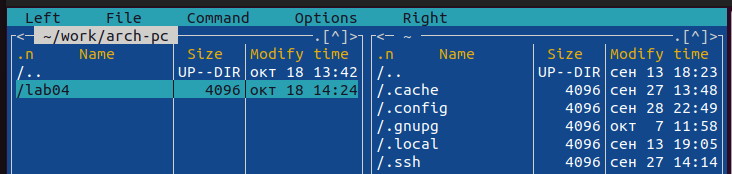


Рис 2

С помощью F7 создаю каталог lab05:

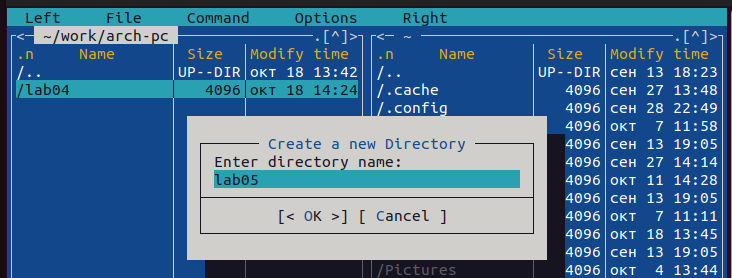


Рис 3

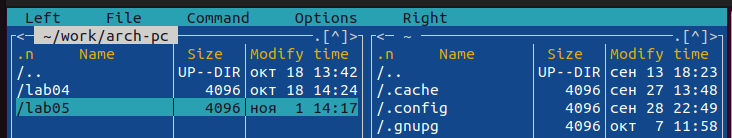


Рис 4

Пользуясь строкой ввода и командой touch создаю файл lab5-1.asm:

Рис 5

Рис 5

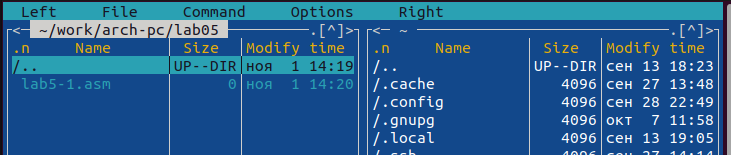


Рис 6

С помощью функциональной клавиши F4 открою файл lab5-1.asm для редактирования в nano:

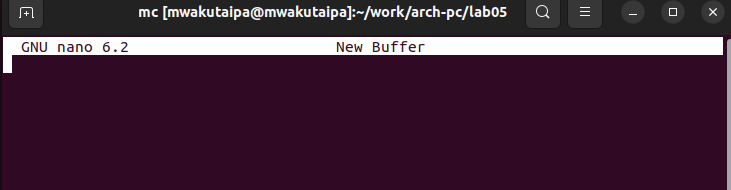


Рис 7

Ввожу в файл код программы для запроса строки у пользователя:

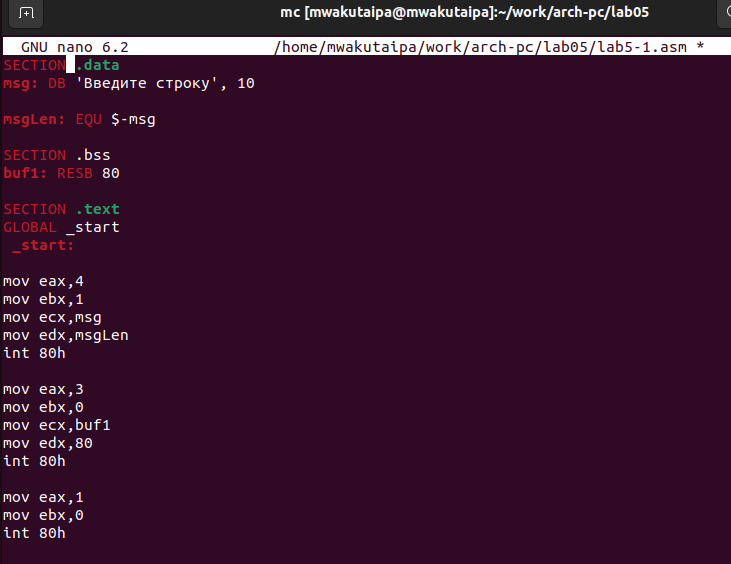


Рис 8

С помощью функциональной клавиши F3 открываю файл для просмотра, чтобы проверить, что файл содержит текст программы:

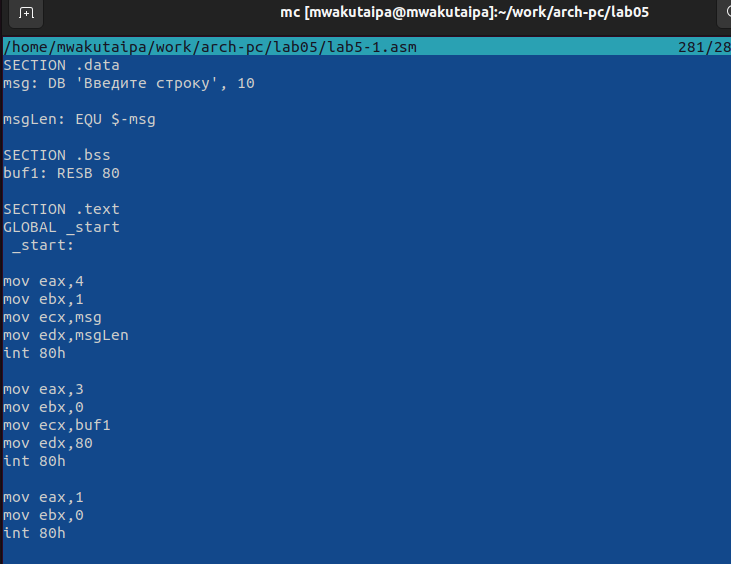


Рис 9

Транслирую текст программы файла в объектный файл командой nasm -f elf lab5-1.asm:

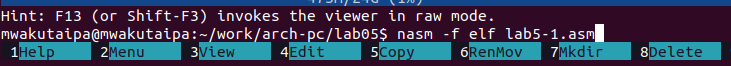


Рис 10

Выполняю компоновку объектного файла с помощью команды ld -m elf\_i386 -o lab5-1 lab5-1.o:

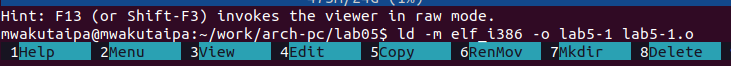


Рис 11

Я запускаю получившийся исполняемый файл:

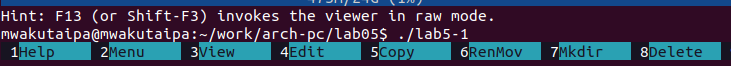


Рис 12

Программа выводит строку “Введите строку:” и ждет ввода с клавиатуры, я ввожу мои ФИО:

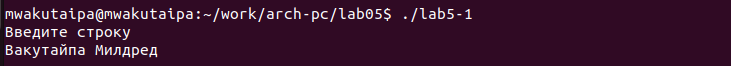


Рис 13

**Подключение внешнего файла in\_out.asm**

Скачиваю файл in\_out.asm со страницы курса в ТУИС:

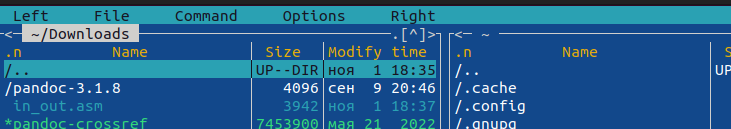


Рис 14

С помощью функциональной клавиши F5 копирую файл in\_out.asm из каталога Downloads в каталог lab05. Потом копирую файл lab5-1.asm в тот же каталог, но с другим именем (lab5-2.asm) :

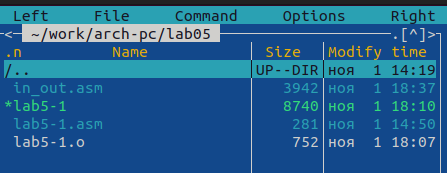


Рис 15

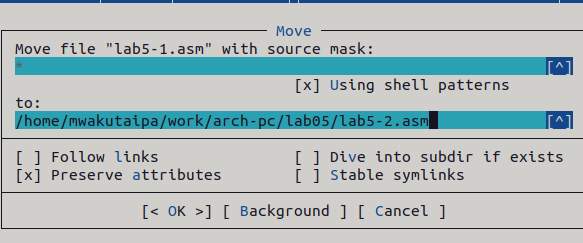


Рис 16

Изменяю содержимое файла lab5-2.asm в редакторе nano, чтобы в программе использовались подпрограммы из внешнего файла in\_out.asm (и также использую подпрограммы sprintLF, sread и quit):

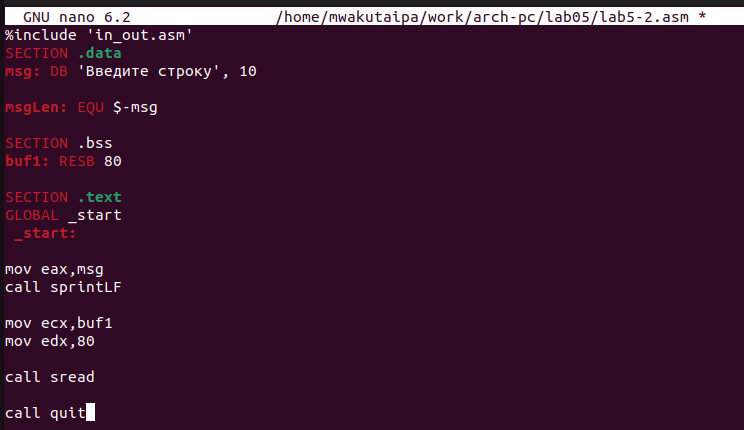


Рис 17

Я транслирую текст программы файла в объектный файл командой nasm -f elf lab5-2.asm и выполняю я компоновку объектного файла с помощью команды ld -m elf\_i386 -o lab5-2 lab5-2.o. Я запускаю получившийся исполняемый файл. Программа выводит строку “Введите строку” и ждет ввода с клавиатуры:

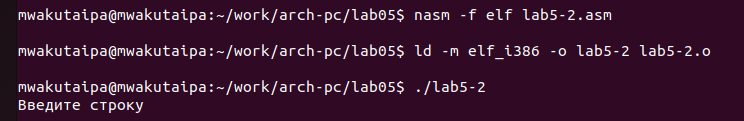


Рис 18

Я ввожу мои ФИО:

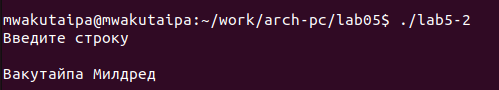


Рис 19

В файле lab5-2.asm заменяю подпрограмму sprintLF на sprint, транслирую и запускаю получившийся исполняемый файл:



Рис 20

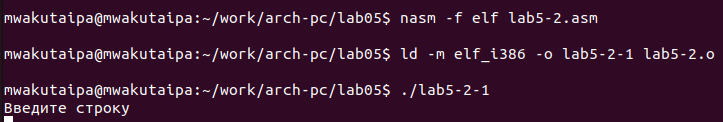


Рис 21

Разница в том, что после строки “Введите строку” нет дополнительной строки:

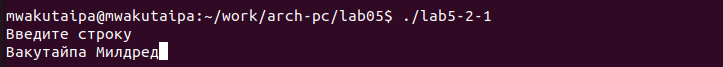


Рис 22

# 4 Выполнение заданий для самостоятельной работы

Создаю копию файла lab5-1.asm с именем lab5-1-0.asm с помощью клавиши F5:

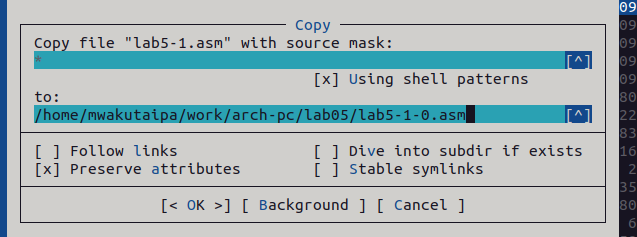


Рис 23

С помощью клавиши F4, открываю созданный файл для редактирования в nano. Изменяю программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку:



Рис 24

Код программы:

``` 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

Я транслирую и запускаю получившийся исполняемый файл:  
  
![Рис 25](image/4.24.png){#fig:001 width=70%}  
  
![Рис 26](image/4.25.png){#fig:001 width=70%}  
  
![Рис 27](image/4.26.png){#fig:001 width=70%}  
  
Программа запрашивает ввод, ввожу мои ФИО, далее программа выводит введенные данные:  
  
![Рис 28](image/4.27.png){#fig:001 width=70%}  
  
Создаю копию файла lab5-2.asm с именем lab5-2-1.asm с помощью функциональной клавиши F5 и открываю созданный файл для редактирования. Изменяю программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку:  
  
![Рис 29](image/4.29.png){#fig:001 width=70%}  
  
Код программы:  
  
```%include 'in\_out.asm'  
SECTION .data  
msg: DB 'Введите строку', 10  
  
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  
mov eax,4  
mov ebx,1  
mov ecx,buf1  
int 80h  
call quit

Я транслирую и запускаю получившийся исполняемый файл:

Рис 30

Рис 30

Рис 31

Рис 31

Рис 32

Рис 32

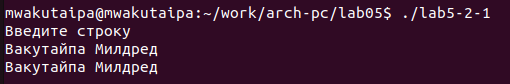


Рис 33

# 5 Выводы

При выполнении данной лабораторной работы я приобрела практические навыки работы в Midnight Commander, а также освоила инструкции языка ассемблера mov и int.

# Список литературы

[Архитектура ЭВМ](https://esystem.rudn.ru/pluginfile.php/2089085/mod_resource/content/0/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%965.%20%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D1%8B%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D1%8B%20%D1%81%20Midnight%20Commander%20%28%29.%20%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D1%8B%20%D0%BD%D0%B0%20%D1%8F%D0%B7%D1%8B%D0%BA%D0%B5%20%D0%B0%D1%81%D1%81%D0%B5%D0%BC%D0%B1%D0%BB%D0%B5%D1%80%D0%B0%20NASM.%20%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%BD%D1%8B%D0%B5%20%D0%B2%D1%8B%D0%B7%D0%BE%D0%B2%D1%8B%20%D0%B2%20%D0%9E%D0%A1%20GNU%20Linux.pdf)