

Отчет по лабораторной работе №5

Архитектура вычислительных систем

Лукина Ирина Олеговна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	9

Список иллюстраций

2.1	Открытие МС	6
2.2	Каталог 4 работы	6
2.3	Создание файла	6
2.4	Редактирование в nano	7
2.5	Введение данных	7
2.6	Копирование файла	8
2.7	Редактирование lab05-2.asm	8

Список таблиц

1 Цель работы

Целью работы является приобретение практических навыков работы в Midnight Commander и освоение инструкций языка ассемблера mow и int.

2 Выполнение лабораторной работы

1. Введем в терминале команду `mc` и откроем Midnight Commander:

```
iolukina@dk8n67 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ mc
```

Рис. 2.1: Открытие MC

2. Переходим в каталог, созданный при выполнении работы №4 и создаем папку `lab05`:

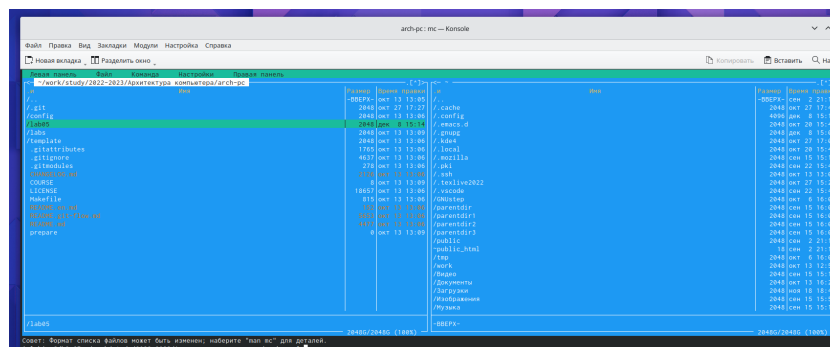


Рис. 2.2: Каталог 4 работы

3. С помощью команды `touch` создаем файл `lab05.asm`:

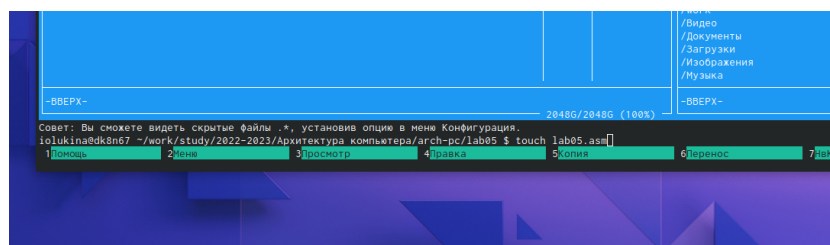
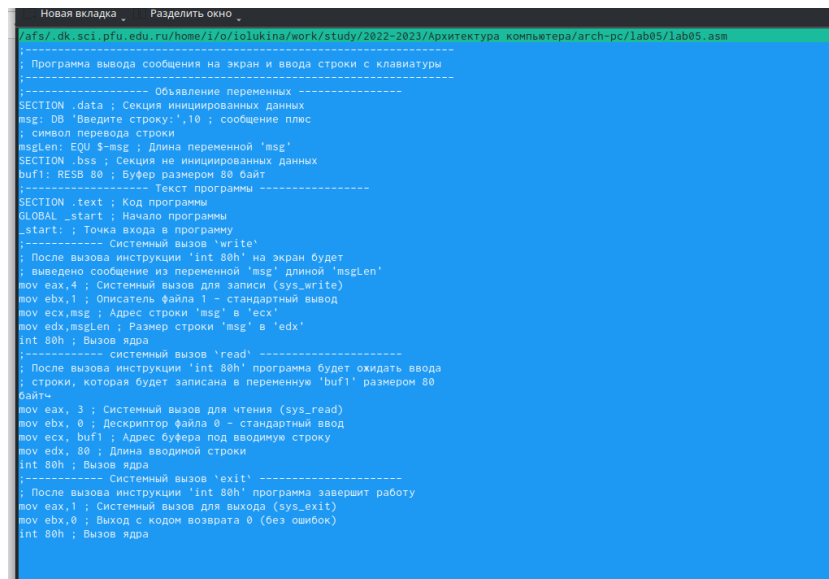


Рис. 2.3: Создание файла

4. Клавишей F4 откроем созданный файл и отредактируем его, введя текст программы из листинга 6.1, сохраняем изменения и закрываем:



```
Новая вкладка Разделить окно
/afs/dfk.sci.pfu.edu.ru/home/i/o/iolukina/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05/lab05.asm
; Программа 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 ebx, 4 ; Системный вызов для записи (sys_write)
mov ecx, 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 ebx, 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 ebx, 1 ; Системный вызов для выхода (sys_exit)
mov ebx, 0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра
```

Рис. 2.4: Редактирование в nano

5. Клавиша F3 откроет файл. Просматриваем и убеждаемся, что он содержит текст программы.
6. Транслируем текст программы lab05.asm в объектный файл. Выполняем его компоновку и запускаем получившийся файл. Терминал ожидает ввода после “Введите строку:”. Вводим свои ФИО:



```
iolukina@dk8n67 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ nasm -f elf lab05.asm
iolukina@dk8n67 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ld -m elf_i386 -o lab05.o
ld: отсутствует входные файлы
iolukina@dk8n67 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ld -m elf_i386 -o lab05 lab05.o
iolukina@dk8n67 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ./lab05
Введите строку:
Lukina Rina Olegovna
iolukina@dk8n67 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $
```

Рис. 2.5: Введение данных

7. Скачиваем файл in_out.asm и копируем его в тот же каталог, что и файл с программой, т.к. для корректной работы он должен лежать в этом каталоге. После этого создаем копию файла lab05.asm и даем ему название lab05-2.asm:

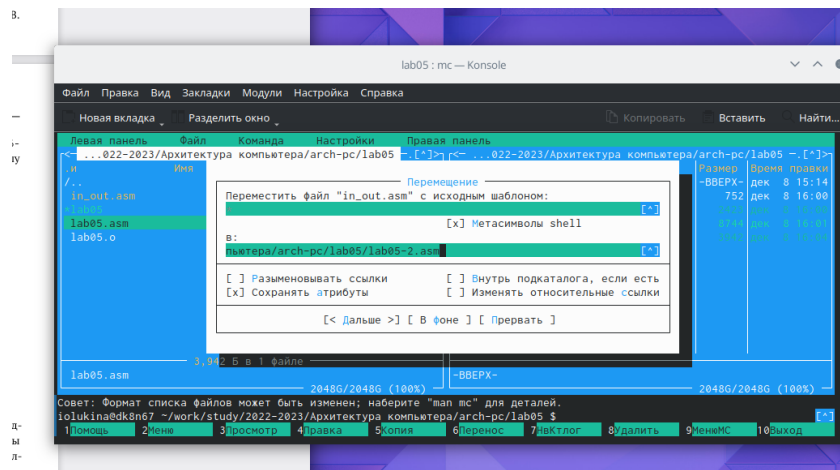


Рис. 2.6: Копирование файла

8. Исправляем в новом файле текст с использованием подпрограмм из внешнего файла в соответствии с листингом 6.2. Создаем исполняемый файл и проверяем его работу.



Рис. 2.7: Редактирование lab05-2.asm

3 Выводы

В ходе выполнения лабораторной работы мы научились работать с Midnight Commander и освоили инструкции языка mow и int