# РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук Кафедра прикладной информатики и теории вероятностей

# ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5

дисциплина: Архитектура вычислительных систем

Студент: Панченко Денис Дмитриевич

Группа: НБИбд-03-22

МОСКВА

2022 г.

## Цель работы:

Приобрести практические навыки работы в Midnight Commander. Освоить инструкции языка ассемблера mov и int.

### Ход работы:

1) Открываем Midnight Commander. (Рис. 1-2)

#### ddpanchenko@nbibd-03-22:-\$ mc

Рис. 1. Вызов Midnight Commander

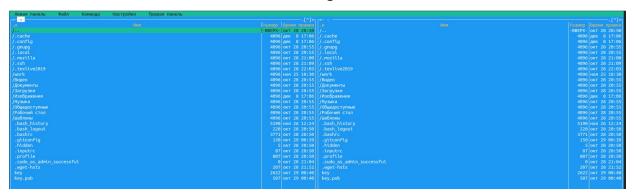


Рис. 2. Midnight Commander

2) Переходим в каталог ~/work/archpc. (Рис. 3)

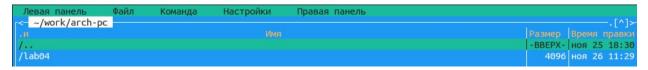


Рис. 3. Каталог

3) Создаем папку lab05. (Рис. 4)

И переходим в созданный каталог. (Рис. 5)

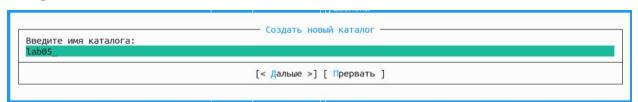


Рис. 4. Создание каталога

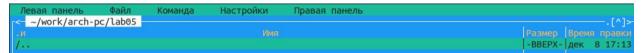


Рис. 5. Каталог

4) Создаем файл lab5-1.asm. (Рис. 6)

ddpanchenko@nbibd-03-22:~/work/arch-pc/lab05\$ touch lab5-1.asm\_

Рис. 6. Создание файла

5) Открываем файл lab6-1.asm для редактирования. (Рис. 7)

```
GNU nano 4.8 /home/ddpanchenko/work/arch-pc/lab05/lab5-1.asm
```

Рис. 7. Открытие файла

6) Вводим текст программы, сохраняем изменения и закрываем файл. (Рис. 8)

```
GNU nano 4.8
                                                                                                           /home/ddpanchenk
: Программа вывода сообщения на экран и ввода строки с клавиатуры
 ----- Объявление переменных
        .data ; Секция инициированных данных
    DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки
           I Ş-msg ; Длина переменной 'msg'
     ON .bss ; Секция не инициированных данных
CTION .text ; Код программы
 LOBAL _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 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
байт4
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 <u>в</u>водимой строки
int 80h ; Вызов ядра
;------ Системный вызов `exit` ------;;После вызова инструкции 'int 80h' программа завершит работу
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра
```

Рис. 8. Программа

7) Открываем файл lab5-1.asm для просмотра. Убеждаемся, что файл содержит текст программы. (Рис. 9)

Рис. 9. Программа

8) Оттранслируем текст программы lab5-1.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл. (Рис. 10)

```
ddpanchenko@nbibd-03-22:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
ddpanchenko@nbibd-03-22:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
ddpanchenko@nbibd-03-22:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Панченко Денис Дмитриевич
```

Рис. 10. Запуск файла

### 5.3.1. Подключение внешнего файла in out.asm

9) Скачаем файл in out.asm со страницы курса в ТУИС. (Рис. 11)

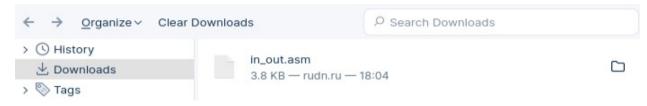


Рис. 11. Загрузка файла

10) Скопируем файл in\_out.asm в каталог с файлом lab5-1.asm. (Рис. 12)

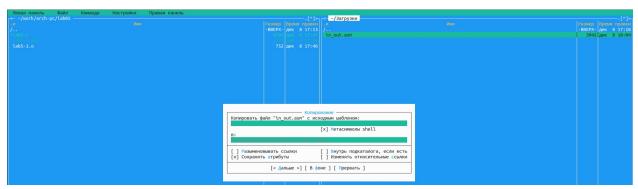


Рис. 12. Копирование файла

11) Создаем копию файла lab5-1.asm с именем lab5-2.asm. (Рис. 13)

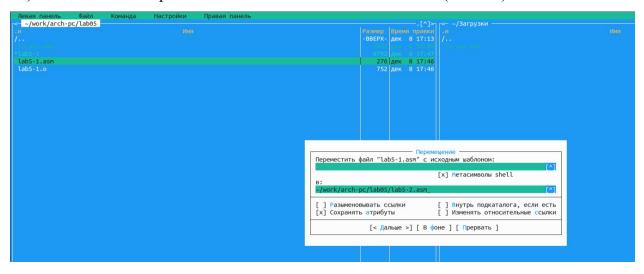


Рис. 13. Копия файла

12) Исправим текст программы в файле lab5-2.asm. (Рис. 14)

```
GNU nano 4.8
%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 ecx, buf1
mov edx, 80
call sread
call quit
```

Рис. 14. Текст программы

13) Создаем исполняемый файл и проверяем его работу. (Рис. 15)

```
ddpanchenko@nbibd-03-22:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
ddpanchenko@nbibd-03-22:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
ddpanchenko@nbibd-03-22:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Панченко Денис Дмитриевич
```

Рис. 15. Запуск файла

14) Заменяем подпрограмму sprintLF на sprint в файле lab5-2.asm. (Рис. 16)

```
GNU nano 4.8
%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 sprint
mov ecx, buf1
mov edx, 80
call sread
call quit
```

Рис. 16. Изменения

15) Создаем исполняемый файл и проверяем его работу. (Рис. 17)

```
ddpanchenko@nbibd-03-22:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm ddpanchenko@nbibd-03-22:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o ddpanchenko@nbibd-03-22:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:Панченко Денис Дмитриевич
```

Рис. 17. Запуск файла

Разница в том, что подпрограмма sprintLF переводит сообщение на следующую строку, а sprint данного действия не делает.

#### Задание для самостоятельной работы.

1) Создаем копию файла lab5-1.asm. (Рис. 18)

Вносим изменения в программу. (Рис. 19)

```
Копировать файл "lab5-1.asm" с исходным шаблоном:

[x] Метасимволы shell

в:
-/work/arch-pc/lab05/lab5-1s.asm

[] Разыменовывать ссылки
[x] Сохранять атрибуты

[ 3 Внутрь подкаталога, если есть
[x] Сохранять атрибуты

[ 4 Дальше >] [ В фоне ] [ Прервать ]
```

Рис. 18. Создание копии

```
GNU nano 4.8
 Программа вывода сообщения на экран и ввода строки с клавиатуры
              ----- Объявление переменных ---
        .data ; Секция инициированных данных
   : DB 'Введите строку:',10 ; сообщение плюс
 символ перевода строки
           Ј$-msg ; Длина переменной 'msg'
        .bss ; Секция не инициированных данных
       ESB 80 ; Буфер размером 80 байт
   ----- Текст программы ------
       .text ; Код программы
        start ; Начало программы
       ; Точка входа в программу
;------ 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
```

Рис. 19. Изменения

2) Получаем исполняемый файл и проверяем его работу. (Рис. 20)

```
ddpanchenko@nbibd-03-22:~/work/arch-pc/lab05$ nasm -f elf lab5-1s.asm ddpanchenko@nbibd-03-22:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1s lab5-1s.o ddpanchenko@nbibd-03-22:~/work/arch-pc/lab05$ ./lab5-1s Введите строку:
```

Рис. 20. Запуск файла

3) Создаем копию файла lab5-2.asm. (Рис. 21)

Вносим изменения в программу. (Рис. 22)

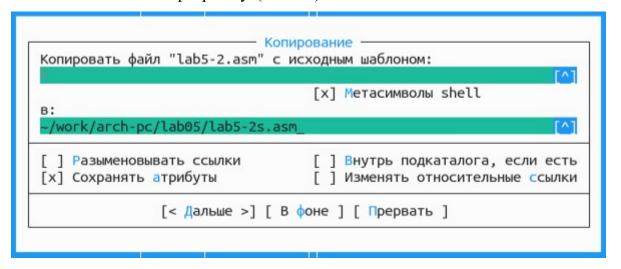


Рис. 21. Создание копии

```
GNU nano 4.8
 Программа вывода сообщения на экран и ввода строки с клавиатуры
%include 'in_out.asm' ; подключение внешнего файла
        .data ; Секция инициированных данных
        'Введите строку: ',0h ; сообщение
       .bss ; Секция не инициированных данных
           80 ; Буфер размером 80 байт
        .text ; Код программы
       start ; Начало программы
     t: ; 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 `
mov edx, 80 ; запись длины вводимого сообщения в `EBX`
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения_
```

Рис. 22. Изменения

4) Получаем исполняемый файл и проверяем его работу. (Рис. 23)

```
ddpanchenko@nbibd-03-22:~/work/arch-pc/lab05$ nasm -f elf lab5-2s.asm
ddpanchenko@nbibd-03-22:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2s lab5-2s.o
ddpanchenko@nbibd-03-22:~/work/arch-pc/lab05$ ./lab5-2s
Введите строку:
Панченко
```

Рис. 23. Запуск файла

# Вывод:

В ходе выполнения данной лабороторной работы я приобрел практические навыки работы в Midnight Commander и освоил инструкции языка ассемблера mov и int.