

Отчет по лабораторной работе №5
Номер студенческого билета:
10322536339

Габдуллина Александра Булатовна

Содержание

1 Цель работы	5
2 Задание	6
3 Выполнение лабораторной работы	7
4 Выводы	16

Список иллюстраций

3.1	7
3.2	7
3.3	Окно Midnight Commander. Создание каталога	8
3.4	Создание файла	8
3.5	8
3.6	Окно Midnight Commander. Редактор nano	9
3.7	9
3.8	9
3.9	Компиляция и линковка lab5-1.asm	10
3.10	Запускаем файл	10
3.11	Окно Midnight Commander. Скопирование файл in_out.asm в каталоге с файлом lab5-1.asm	10
3.12	Создание копии файла	11
3.13	Редактирование файла	11
3.14	Компиляция, линковка lab5-2.asm. Запуск файла	12
3.15	Замена подпрограммы sprintLF на sprint.	12
3.16	Создаём копию	13
3.17	Вносим изменения	13
3.18	Компиляция и линковка. Запуск файла.	14
3.19	Создание копии	14
3.20	Редактирование текста программы	14
3.21	Компиляция и линковка. Запуск файла	15

Список таблиц

1 Цель работы

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

2 Задание

В качестве ответа на лабораторную работу в ТУИС необходимо загрузить :

1.Ссылка на репозиторий (<https://github.com/>) 2.Следует загрузить отдельными файлами (не общим архивом): 3.отчёт в markdown; 4.отчёт в docx (сделанный из markdown); 5.отчёт в pdf (сделанный из markdown); 6.архив с исходными материалами markdown (текстовые файлы, скриншоты и т. д.) 7.Файл(ы) с исходным текстом программ, написанных в ходе выполнения работы (в формате .asm).

3 Выполнение лабораторной работы

Открываем Midnight Commander

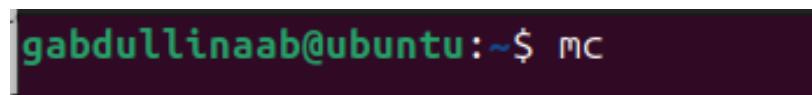


Рисунок 3.1: .

Переходим в каталог `~/work/arch-pc` созданный при выполнении лабораторной работы №4

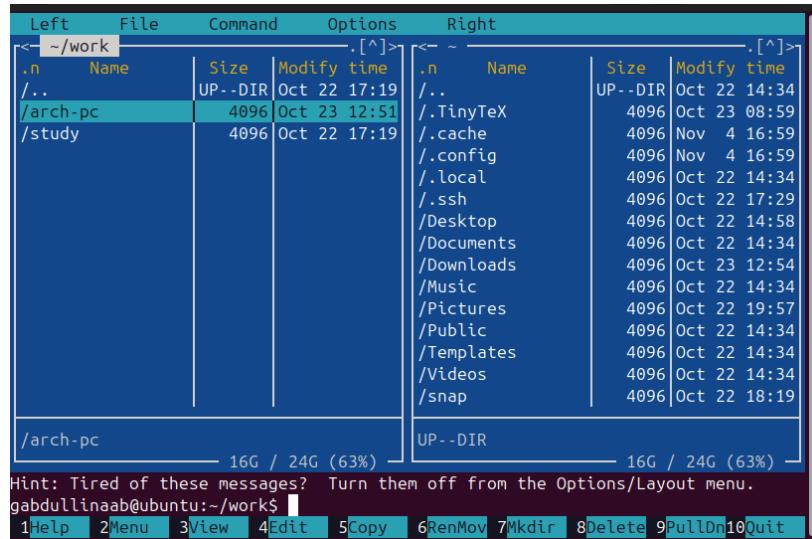


Рисунок 3.2: .

Создаём папку `lab05` и переходим в созданный каталог

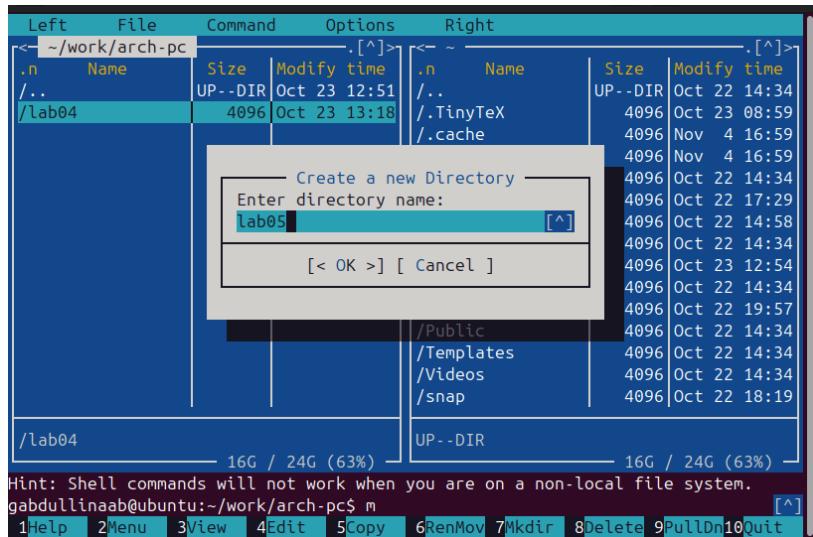


Рисунок 3.3: Окно Midnight Commander. Создание каталога

Создаём файл lab5-1.asm

```
gabdullinaab@ubuntu:~/work/arch-pc/lab05$ touch lab5-1.asm
```

Рисунок 3.4: Создание файла

Открываем файл lab5-1.asm.

```
gabdullinaab@ubuntu:~/work/arch-pc/lab05$ nano lab5-1.asm
```

Рисунок 3.5: .

Вводим текст программы из листинга 5.1 , сохраняем изменения и закрываем файл.

```

GNU nano 7.2      /home/gabdullinaab/work/arch_pc/lab05/lab5-1.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 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` -

```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

Рисунок 3.6: Окно Midnight Commander. Редактор nano

Открываем файл lab5-1.asm для просмотра. Убедждаемся, что файл содержит текст программы.

```

/home/gabdullinaab/work/arch_pc/lab05/lab5-1.asm          1522/2432   62%
;-----+
; 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 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` -

```

1Help 2UnWrap 3Quit 4Hex 5Goto 6 7Search 8Raw 9Format 10Quit

Рисунок 3.7: .

Переходим в нужный нам каталог.

```

gabdullinaab@ubuntu:~$ cd ~/work/arch_pc/lab05
gabdullinaab@ubuntu:~/work/arch_pc/lab05$
```

Рисунок 3.8: .

Оттранслируем текст программы lab5-1.asm в объектный файл. Выполняем компоновку объектного файла и запускаем получившийся исполняемый файл.

```
gabdullinaab@ubuntu:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
gabdullinaab@ubuntu:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
gabdullinaab@ubuntu:~/work/arch-pc/lab05$ ./lab5-1
```

Рисунок 3.9: Компиляция и линковка lab5-1.asm

Программа выводит строку **Введите строку:** и ожидает ввода с клавиатуры. На запрос ввожу своё ФИО.

```
gabdullinaab@ubuntu:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
GabdullinaAB
```

Рисунок 3.10: Запускаем файл

Копирую файл in_out.asm в каталог с файлом lab5-1.asm .

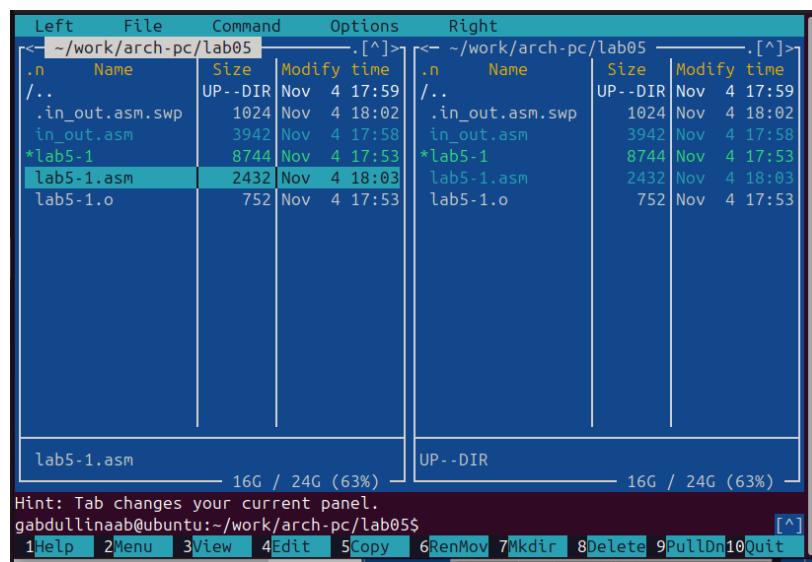


Рисунок 3.11: Окно Midnight Commander. Скопирование файла in_out.asm в каталоге с файлом lab5-1.asm

Создаю копию файла lab5-1.asm с именем lab5-2.asm.

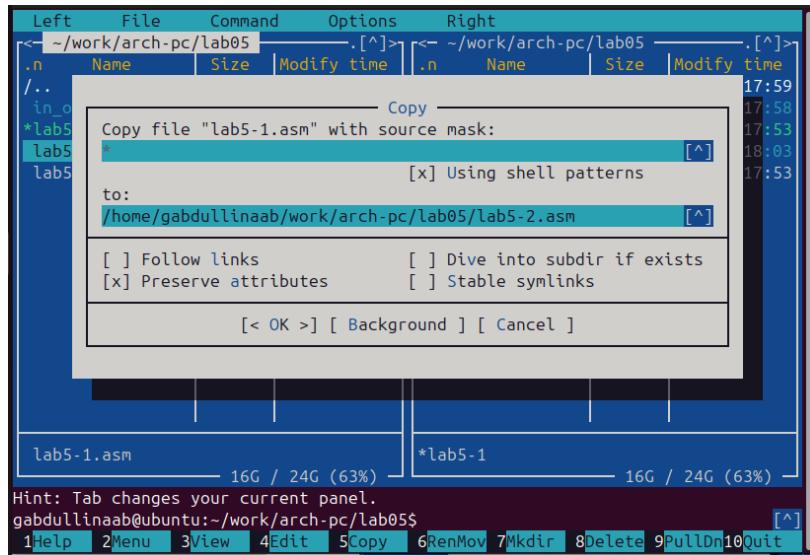


Рисунок 3.12: Создание копии файла

Исправляю текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in_out.asm (использую подпрограммы sprintLF, sread и quit) в соответствии с листингом 5.2.

```
GNU nano 7.2      /home/gabdullinaab/work/arch-pc/lab05/lab5-2.asm
-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не инициализированных 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 `EBX`
    mov edx, 80 ; запись длины вводимого сообщения в `EBX`
    call sread ; вызов подпрограммы печати сообщения с переводом строки
    call quit ; завершение программы

[Wrote 18 lines]
^C Help      ^O Write Out ^W Where Is ^K Cut      ^T Execute ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste    ^J Justify ^/ Go To Line
```

Рисунок 3.13: Редактирование файла

Компиляция и линковка lab5-2.asm. Запускаем получившийся исполняемый файл.

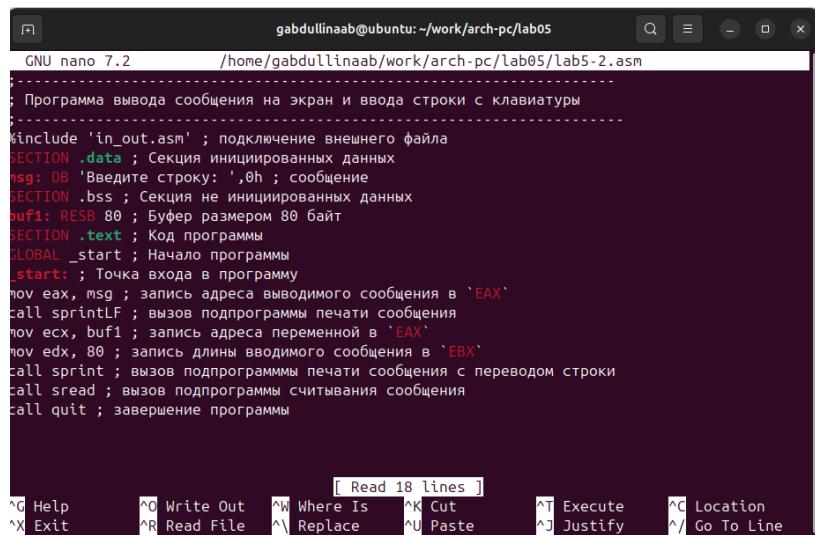
```

gabdullinaab@ubuntu:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
gabdullinaab@ubuntu:~/work/arch-pc/lab05$ nasm -f elf in_out.asm
gabdullinaab@ubuntu:~/work/arch-pc/lab05$ ld -m elf_i386 lab5-2.o in_out.o -o lab5-2
gabdullinaab@ubuntu:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:

```

Рисунок 3.14: Компиляция, линковка lab5-2.asm. Запуск файла

В файле lab5-2.asm заменяем подпрограмму sprintLF на sprint.



```

GNU nano 7.2          /home/gabdullinaab/work/arch-pc/lab05/lab5-2.asm
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;
;include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не инициализированных 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 sprint ; вызов подпрограммы печати сообщения с переводом строки
    call sread ; вызов подпрограммы считывания сообщения
    call quit ; завершение программы

[ Read 18 lines ]
^G Help      ^O Write Out   ^W Where Is   ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File   ^\ Replace    ^U Paste     ^J Justify   ^/ Go To Line

```

Рисунок 3.15: Замена подпрограммы sprintLF на sprint.

Разница в том, что sprintLF добавляет автоматический перевод строки после вывода сообщения, а sprint выводит текст без перевода строки, и ввод происходит на той же строке.

#Задание для самостоятельной работы

Создаём копию файла lab5-1.asm и называем этот файл lab5-1.2.asm

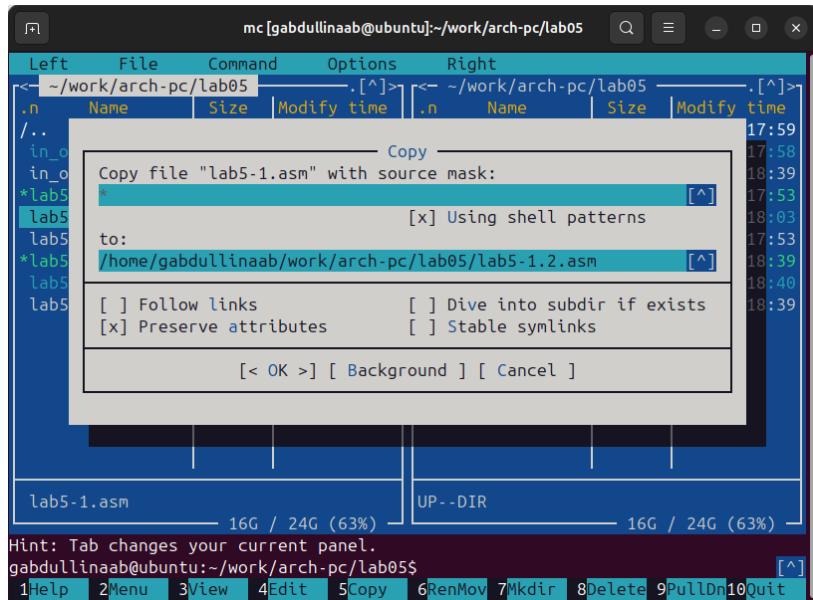


Рисунок 3.16: Создаём копию

Вносим изменения в программу (без использования внешнего файла `in_out.asm`), так чтобы она работала по данному в задании алгоритму.

```
gabdullinaab@ubuntu:~/work/arch-pc/lab05
GNU nano 7.2 /home/gabdullinaab/work/arch-pc/lab05/lab5-1.2.asm *
section .data
    msg db 'Введите строку: ',0
    len_msg equ $-msg

section .bss
    input resb 100      ; буфер для ввода строки
    len_input equ $-input

section .text
    global _start

_start:
    ; Вывод приглашения "Введите строку:"
    mov eax, 4          ; системный вызов write
    mov ebx, 1          ; файловый дескриптор stdout
    mov ecx, msg        ; указатель на сообщение
    mov edx, len_msg    ; длина сообщения
    int 0x80

    ; Ввод строки с клавиатуры
    mov eax, 3          ; системный вызов read
    mov ebx, 0           ; файловый дескриптор stdin
    mov ecx, input       ; буфер для ввода
    mov edx, len_input   ; количество байт для чтения
    int 0x80
```

Рисунок 3.17: Вносим изменения

Компиляция и линковка `lab5-1.2.asm`. Запускаем получившийся исполняемый файл. Проверяем его работу. На приглашение ввести строку ввожу своё ФИО

```

gabdullinaab@ubuntu:~/work/arch-pc/lab05$ nasm -f elf lab5-1.2.asm
gabdullinaab@ubuntu:~/work/arch-pc/lab05$ ld -m elf_i386 lab5-1.2.o in_out.o -o lab5-1.2
Введите строку: GabdullinaAB
GabdullinaAB

```

Рисунок 3.18: Компиляция и линковка. Запуск файла.

Создаю копию файла lab5-2.asm и называю lab5-2.2.asm .

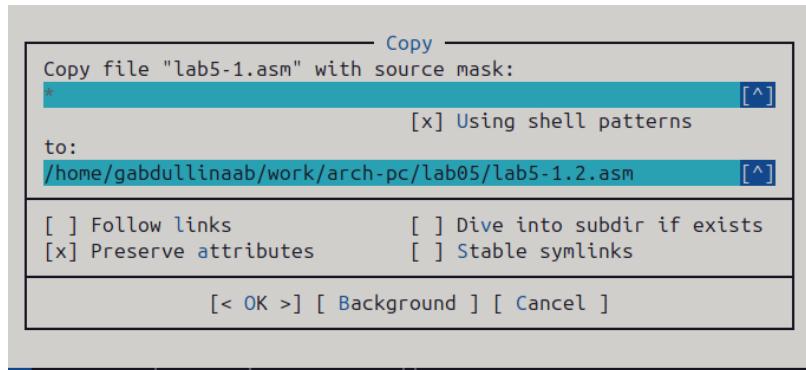


Рисунок 3.19: Создание копии

Исправляю текст программы с использование подпрограмм из внешнего файла in_out.asm, так чтобы она работала по данному в задании алгоритму.

```

GNU nano 7.2                               /home/gabdullinaab/work/arch-pc/lab05/report
#include 'in_out.asm'

SECTION .data
    msg db 'Введите строку: ',0h

SECTION .bss
    input resb 100      ; буфер для ввода строки

SECTION .text
    global _start

_start:
    ; Вывод приглашения "Введите строку:"
    mov eax, msg
    call sprint      ; вывод без перевода строки

    ; Ввод строки с клавиатуры
    mov ecx, input   ; буфер для ввода
    mov edx, 100      ; максимальная длина
    call sread       ; вызов подпрограммы ввода

    ; Вывод введенной строки
    mov eax, input
    call sprintLF   ; вывод с переводом строки

    ; Завершение программы
    call quit

```

Рисунок 3.20: Редактирование текста программы

Компиляция и линковка lab5-2.2.asm. Запускаем получившийся исполняемый файл. Проверяем его работу. На приглашение ввести строку ввожу своё ФИО.

```
gabdullinaab@ubuntu:~/work/arch-pc/lab05$ nasm -f elf lab5-2.2.asm
gabdullinaab@ubuntu:~/work/arch-pc/lab05$ ld -m elf_i386 lab5-2.2.o in_out.o -o lab5-2.2
gabdullinaab@ubuntu:~/work/arch-pc/lab05$ ./lab5-2.2
Введите строку: GabdullinaAB
```

Рисунок 3.21: Компиляция и линковка. Запуск файла

4 Выводы

В ходе лабораторной работы я приобрела практические навыки работы в Midnight Commander. Также я освоение инструкции языка ассемблера mov и int.