

Отчёта по лабораторной работе №5

Дисциплина: архитектура компьютера

Эзиз Хатамов

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
4.1	открытие МС	8
4.2	Подключения внешнего файла	11
5	Задание для самостоятельной работы	14
5.1	Первая работа	14
5.2	Вторая работа	15
6	Выводы	17
	Список литературы	18

Список иллюстраций

4.1	Открытие МС	8
4.2	Создания папки в каталоге	9
4.3	Создания файла	9
4.4	Редактирования файла	10
4.5	оттранслирования, компановка, ввеления ФИО	11
4.6	Копирования файла в нужный каталог	11
4.7	Редактирования файла	12
4.8	Проверка файла	12
4.9	Изменения sprintLF на sprint	13
4.10	Разница между srintLF и sprint	13
5.1	Исправления текста lab5-1-1	15
5.2	Создания исполняемого файла и проверка файла lab5-1-1	15
5.3	Исправления текста lab5-2-2	16
5.4	Создания исполняемого файла и проверка файла lab5-2-2	16

Список таблиц

1 Цель работы

Целью данной лабораторной работы является приобретение практических навыков работы в Midnight Commander, освоение инструкций языка ассемблера `mov` и `int`.

2 Задание

1. Основы работы с Midnight Commander
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. Следующие комбинации клавиш облегчают работу с Midnight Commander:

- Tab используется для переключения между панелями;
- ↑ и ↓ используется для навигации, Enter для входа в каталог или открытия файла (если в файле расширений mc.ext заданы правила связи определённых расширений файлов с инструментами их запуска или обработки);
- Ctrl + u (или через меню Команда > Переставить панели) меняет местами содержимое правой и левой панелей;
- Ctrl + o (или через меню Команда > Отключить панели) скрывает или возвращает панели Midnight Commander, за которыми доступен для работы командный интерпретатор оболочки и выводимая туда информация.
- Ctrl + x + d (или через меню Команда > Сравнить каталоги) позволяет сравнить содержимое каталогов, отображаемых на левой и правой панелях.

Дополнительную информацию о Midnight Commander можно получить по команде `man mc`.

4 Выполнение лабораторной работы

4.1 открытие MC

Для начала я открыл Midnight Commander (рис. 4.1).

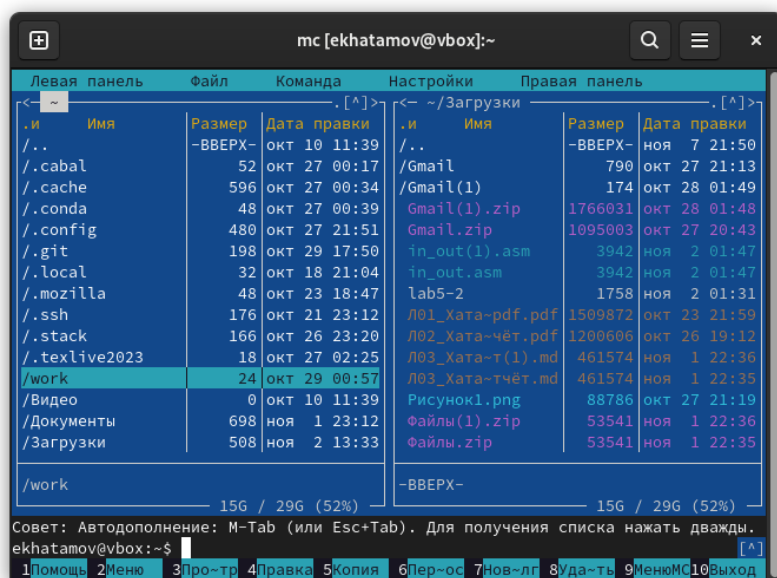


Рис. 4.1: Открывание MC

##Создание папки Потом перешел на каталог созданная в 4 лабораторной работе work/arch-рс и там создал новую папку с помощью F7

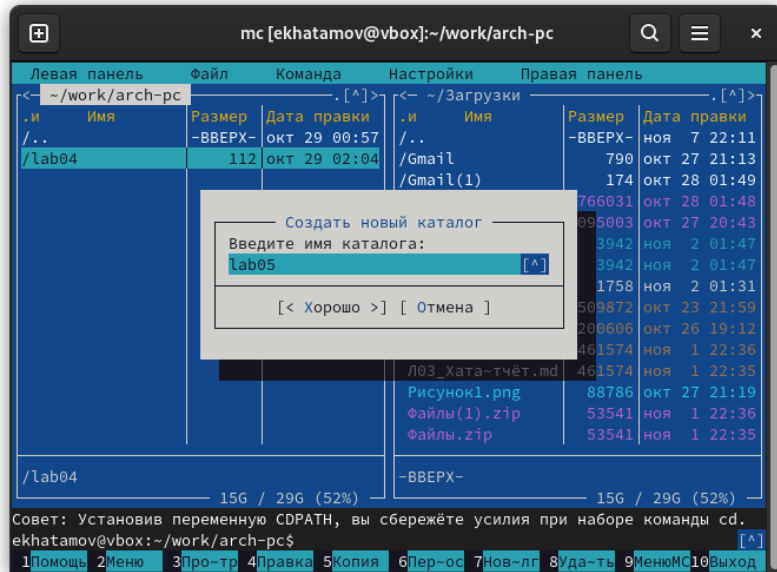


Рис. 4.2: Создания папки в каталоге

##Создание файла

После я создал файл lab5-1.asm с помощью команды touch в созданной папке

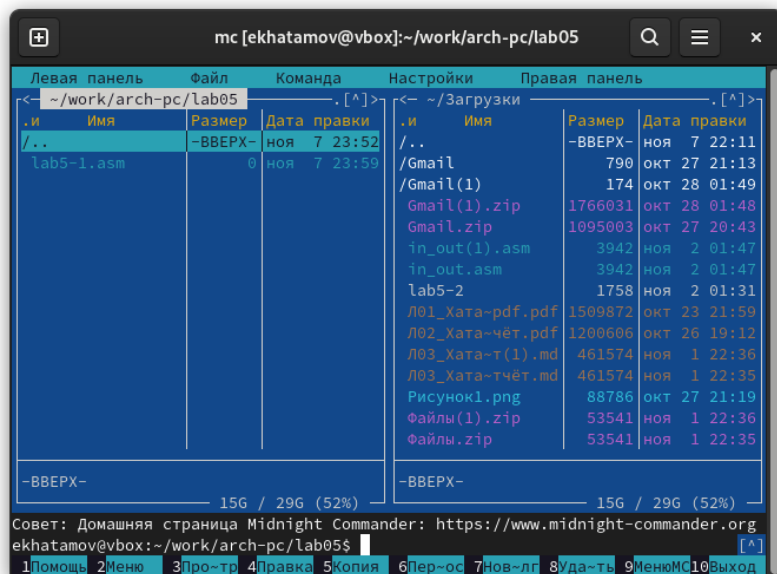
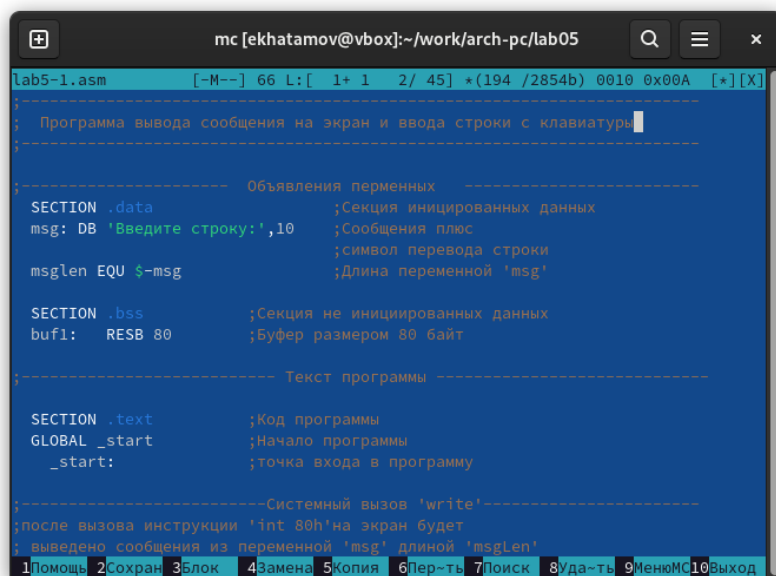


Рис. 4.3: Создания файла

##Редактирование файла С помощью клавиши F4 я открыл файл и внес редак-

тирования как было указанау Потом сохранил и закрыл файл



```
lab5-1.asm [-M--] 66 L: [ 1+ 1 2/ 45] *(194 /2854b) 0010 0x00A [*] [X]

; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
;----- Объявления переменных -----
SECTION .data ;Секция иницированных данных
msg: DB 'Введите строку:',10 ;Сообщения плюс
;символ перевода строки
msglen EQU $-msg ;Длина переменной 'msg'

SECTION .bss ;Секция не иницированных данных
buf1: RESB 80 ;Буфер размером 80 байт

;----- Текст программы -----

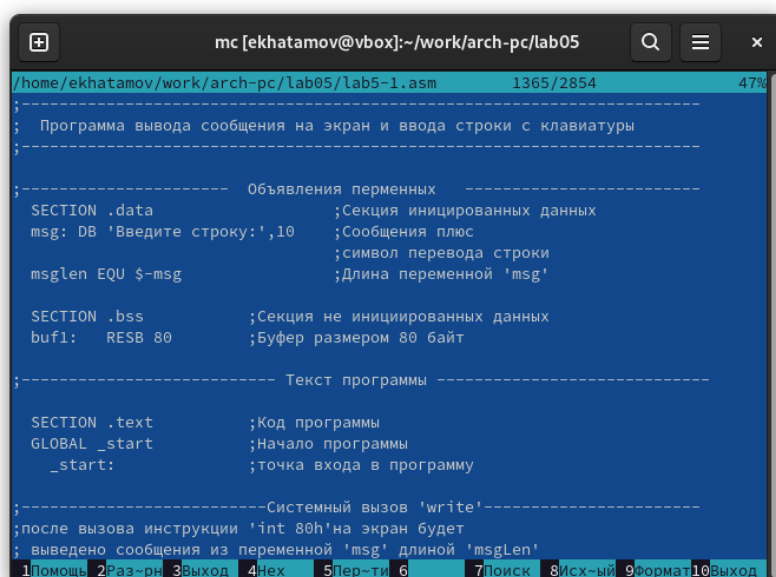
SECTION .text ;Код программы
GLOBAL _start ;Начало программы
_start: ;точка входа в программу

;-----Системный вызов 'write'-----
;после вызова инструкции 'int 80h'на экран будет
; выведено сообщения из переменной 'msg' длиной 'msglen'

1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюMC10Выход
```

Рис. 4.4: Редактирования файла

С помощью клавиши F3 проверил что файл содержит текст которую я написал



```
/home/ekhatamov/work/arch-pc/lab05/lab5-1.asm 1365/2854 47%

; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
;----- Объявления переменных -----
SECTION .data ;Секция иницированных данных
msg: DB 'Введите строку:',10 ;Сообщения плюс
;символ перевода строки
msglen EQU $-msg ;Длина переменной 'msg'

SECTION .bss ;Секция не иницированных 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 сообщения из переменной 'msg' длиной 'msglen'

1Помощь 2Раз-рн 3Выход 4Чех 5Пер-ти 6 7Поиск 8Исх-ый 9Формат10Выход
```

##Оттранслирования

текста в объектный файл. Компановка файла Оттранслировал текст программы lab5-1.asm в объектный файл.Потом выполнил компоновку объектного файла и запустил получившийся исполняемый файл. На строку 'Введите строку:' написал

свое ФИО

```
ekhatamov@vbox:~/work/arch-pc/lab05$ mc
ekhatamov@vbox:~/work/arch-pc/lab05$ rm -r lab5-1
ekhatamov@vbox:~/work/arch-pc/lab05$ rm -r lab5-1.o
ekhatamov@vbox:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
ekhatamov@vbox:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
ekhatamov@vbox:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Эзиз Хатамов
```

Рис. 4.5: оттранслирования, компоновка, ввеления ФИО

4.2 Подключения внешнего файла

Для начала скачал файл `in_out.asm` со страницы курса в ТУИС. Когда скачалб этот файл лежал в каталоге “Загрузки”. Поэтому я в другой панели открыл каталог “Загрузки” и с помощью клавиши F5 я скопировал файл в тот каталог в котором лежал `lab5-1.asm`

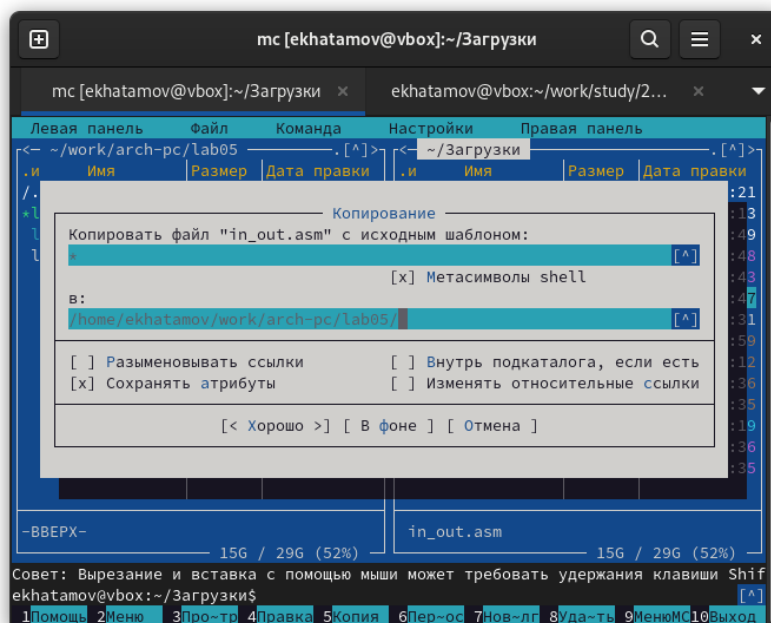
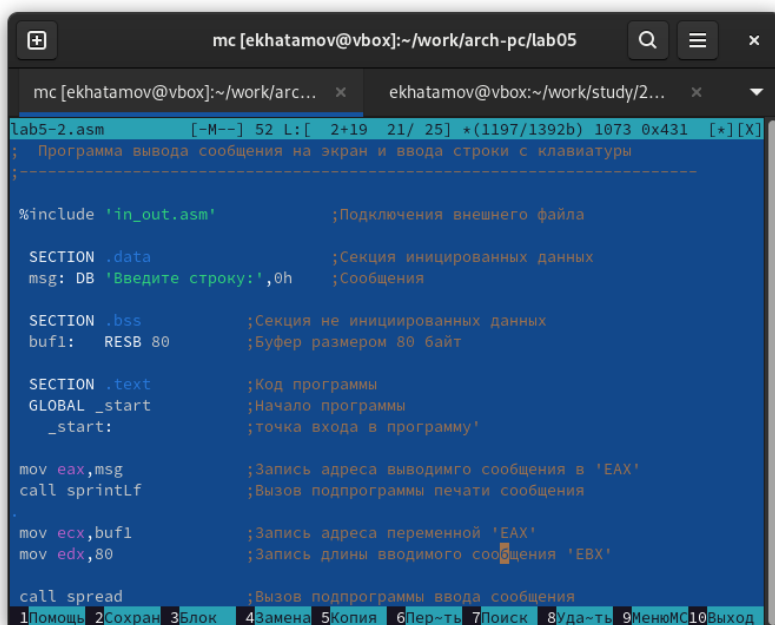


Рис. 4.6: Копирования файла в нужный каталог

Потом исправил текст программы в файле с использованием подпрограммы внешнего файла `in_out.asm`.



```
lab5-2.asm [-M--] 52 L: [ 2+19 21/ 25] *(1197/1392b) 1073 0x431 [*] [X]
; Программа вывода сообщения на экран и ввода строки с клавиатуры
-----
%include 'in_out.asm' ;Подключения внешнего файла

SECTION .data ;Секция иницированных данных
msg: DB 'Введите строку:',0h ;Сообщения

SECTION .bss ;Секция не иницированных данных
buf1: RESB 80 ;Буфер размером 80 байт

SECTION .text ;Код программы
GLOBAL _start ;Начало программы
_start: ;точка входа в программу

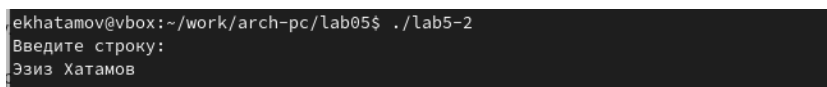
mov eax,msg ;Запись адреса выводимого сообщения в 'EAX'
call sprintf ;Вызов подпрограммы печати сообщения

mov ecx,buf1 ;Запись адреса переменной 'EAX'
mov edx,80 ;Запись длины вводимого сообщения 'EBX'

call spread ;Вызов подпрограммы ввода сообщения
1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюMC10Выход
```

Рис. 4.7: Редактирования файла

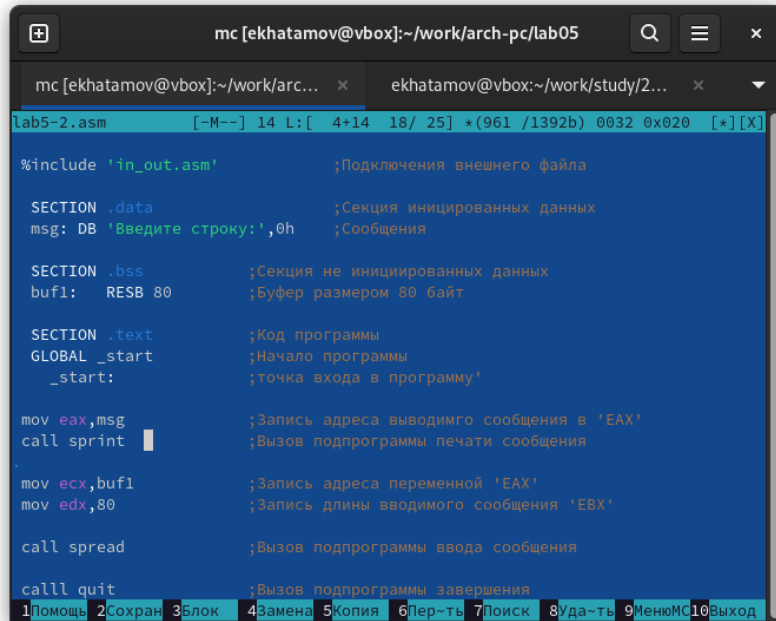
После этого я создал исполняемый файл и проверил его работу



```
ekhatamov@vbox:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Эзиз Хатамов
```

Рис. 4.8: Проверка файла

Потом сменил `sprintf` на `sprint`



```
mc [ekhatamov@vbox]:~/work/arch-pc/lab05
lab5-2.asm [-M--] 14 L: [ 4+14 18/ 25] *(961 /1392b) 0032 0x020 [*] [X]

%include 'in_out.asm'           ;Подключения внешнего файла

SECTION .data                   ;Секция иницированных данных
msg: DB 'Введите строку:',0h    ;Сообщения

SECTION .bss                    ;Секция не иницированных 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

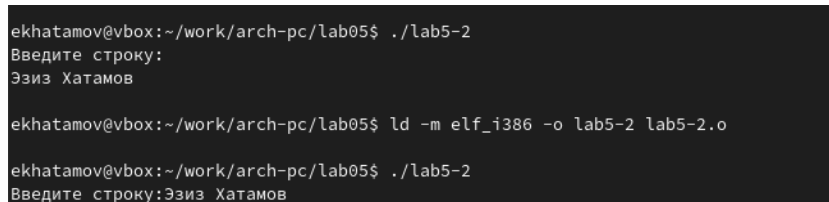
mov ecx,buf1                    ;Запись адреса переменной 'EAX'
mov edx,80                      ;Запись длины вводимого сообщения 'EBX'

call spread                     ;Вызов подпрограммы ввода сообщения

call quit                       ;Вызов подпрограммы завершения
1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюМс10Выход
```

Рис. 4.9: Изменения sprintLF на sprint

Разница между первым исполняемым файлом и вторым в том, что запуск первого запрашивает ввод с новой строки, а программа, которая выполняется при запуске второго, запрашивает ввод без переноса на новую строку, потому что в этом заключается различие между подпрограммами sprintLF и sprint.



```
ekhatamov@vbox:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Эзиз Хатамов

ekhatamov@vbox:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o

ekhatamov@vbox:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:Эзиз Хатамов
```

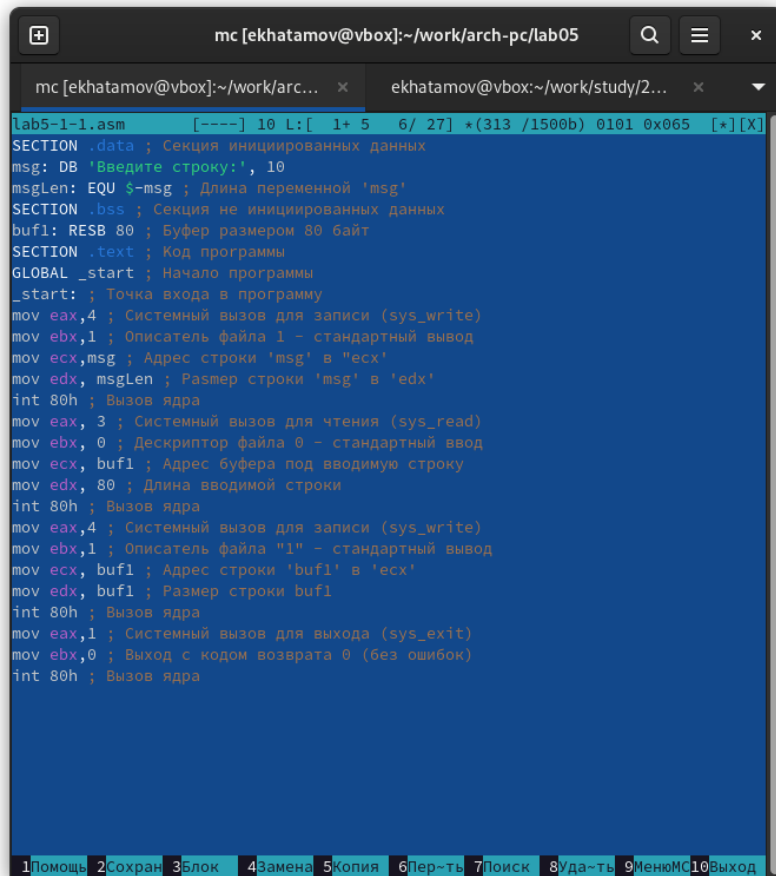
Рис. 4.10: Разница между sprintLF и sprint

5 Задание для самостоятельной работы

5.1 Первая работа

Для начало я создал копию файла lab5-1.asm и исправил текст программыб чтобы она работола по алгоритму:

- Вывести приглашения типа “Введите строку”
- Ввести строку с клавиатуры
- Вывести введённую строку на экран



```
lab5-1-1.asm [----] 10 L: [ 1+ 5 6/ 27] *(313 /1500b) 0101 0x065 [*][X]
SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку:', 10
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не иницированных данных
buf1: RESB 80 ; Буфер размером 80 байт
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
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
mov eax, 3 ; Системный вызов для чтения (sys_read)
mov ebx, 0 ;Descriptor файла 0 - стандартный ввод
mov ecx, buf1 ; Адрес буфера под вводимую строку
mov edx, 80 ; Длина вводимой строки
int 80h ; Вызов ядра
mov eax,4 ; Системный вызов для записи (sys_write)
mov ebx,1 ; Описатель файла "1" - стандартный вывод
mov ecx, buf1 ; Адрес строки 'buf1' в 'ecx'
mov edx, buf1 ; Размер строки buf1
int 80h ; Вызов ядра
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра
```

Рис. 5.1: Исправления текста lab5-1-1

Потом создал исполняемый файл и проверил его работу



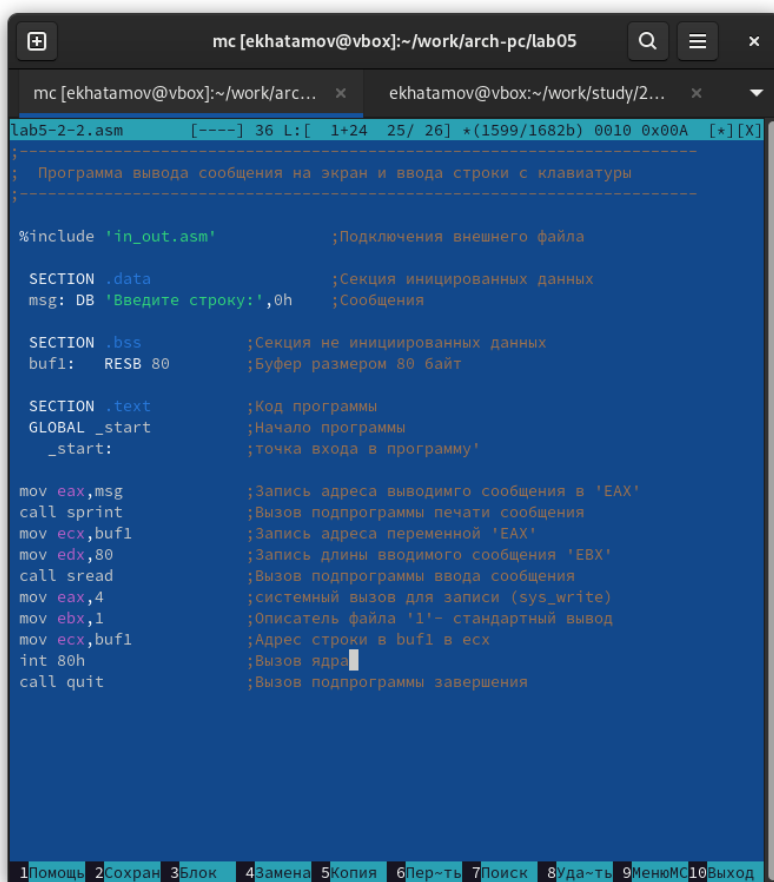
```
ekhatamov@vbox:~/work/arch-pc/lab05$ nasm -f elf lab5-1-1.asm
ekhatamov@vbox:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1-1 lab5-1-1.o
ekhatamov@vbox:~/work/arch-pc/lab05$ ./lab5-1-1
Введите строку:
Эзиз Хатамов
Эзиз Хатамов
ekhatamov@vbox:~/work/arch-pc/lab05$
```

Рис. 5.2: Создания исполняемого файла и проверка файла lab5-1-1

5.2 Вторая работа

Для начала я создал копия файла lab5-2.asm. Потом исправил текст программы с использованием подпрограммы из внешнего файла in_out.asm, Так чтобы

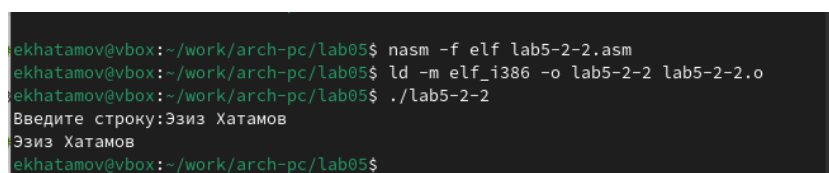
она работала по следующему алгоритму: • вывести приглашение типа “Введите строку.”; • ввести строку с клавиатуры; • вывести введенную строку на экран



```
mc [ekhatamov@vbox]:~/work/arch-pc/lab05
lab5-2-2.asm [----] 36 L: [ 1+24 25/ 26] *(1599/1682b) 0010 0x00A [*] [X]
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
%include 'in_out.asm' ;Подключения внешнего файла
SECTION .data ;Секция иницированных данных
msg: DB 'Введите строку:',0h ;Сообщения
SECTION .bss ;Секция не иницированных 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 выводимго сообщения в 'EAX'
call sprint ;Вызов подпрограммы печати сообщения
mov ecx,buf1 ;Запись адреса переменной 'EAX'
mov edx,80 ;Запись длины вводимого сообщения 'EBX'
call sread ;Вызов подпрограммы ввода сообщения
mov eax,4 ;системный вызов для записи (sys_write)
mov ebx,1 ;Описатель файла '1'-- стандартный вывод
mov ecx,buf1 ;Адрес строки в buf1 в ecx
int 80h ;Вызов ядра
call quit ;Вызов подпрограммы завершения
1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюМС10Выход
```

Рис. 5.3: Исправления текста lab5-2-2

Потом создал исполняемый файл и проверил его работу



```
ekhatamov@vbox:~/work/arch-pc/lab05$ nasm -f elf lab5-2-2.asm
ekhatamov@vbox:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2-2 lab5-2-2.o
ekhatamov@vbox:~/work/arch-pc/lab05$ ./lab5-2-2
Введите строку:Эзиз Хатамов
Эзиз Хатамов
ekhatamov@vbox:~/work/arch-pc/lab05$
```

Рис. 5.4: Создания исполняемого файла и проверка файла lab5-2-2

6 Выводы

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

Список литературы

1. Лабораторная работа №6