

Отчет по лабораторной работе №5

Дисциплина: архитектура компьютера

Марцев Аркадий

Содержание

Цель работы1

Задание1

Выполнение лабораторной работы1

Основы работы с mc1

Структура программы на языке ассемблера NASM4

Подключение внешнего файла5

Выполнение заданий для самостоятельной работы7

Выводы9

Список литературы10

Цель работы

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

Задание

1. Основы работы с mc
2. Структура программы на языке ассемблера NASM
3. Подключение внешнего файла
4. Выполнение заданий для самостоятельной работы

Выполнение лабораторной работы

Основы работы с mc

Открываю Midnight Commander, введя в терминал mc.

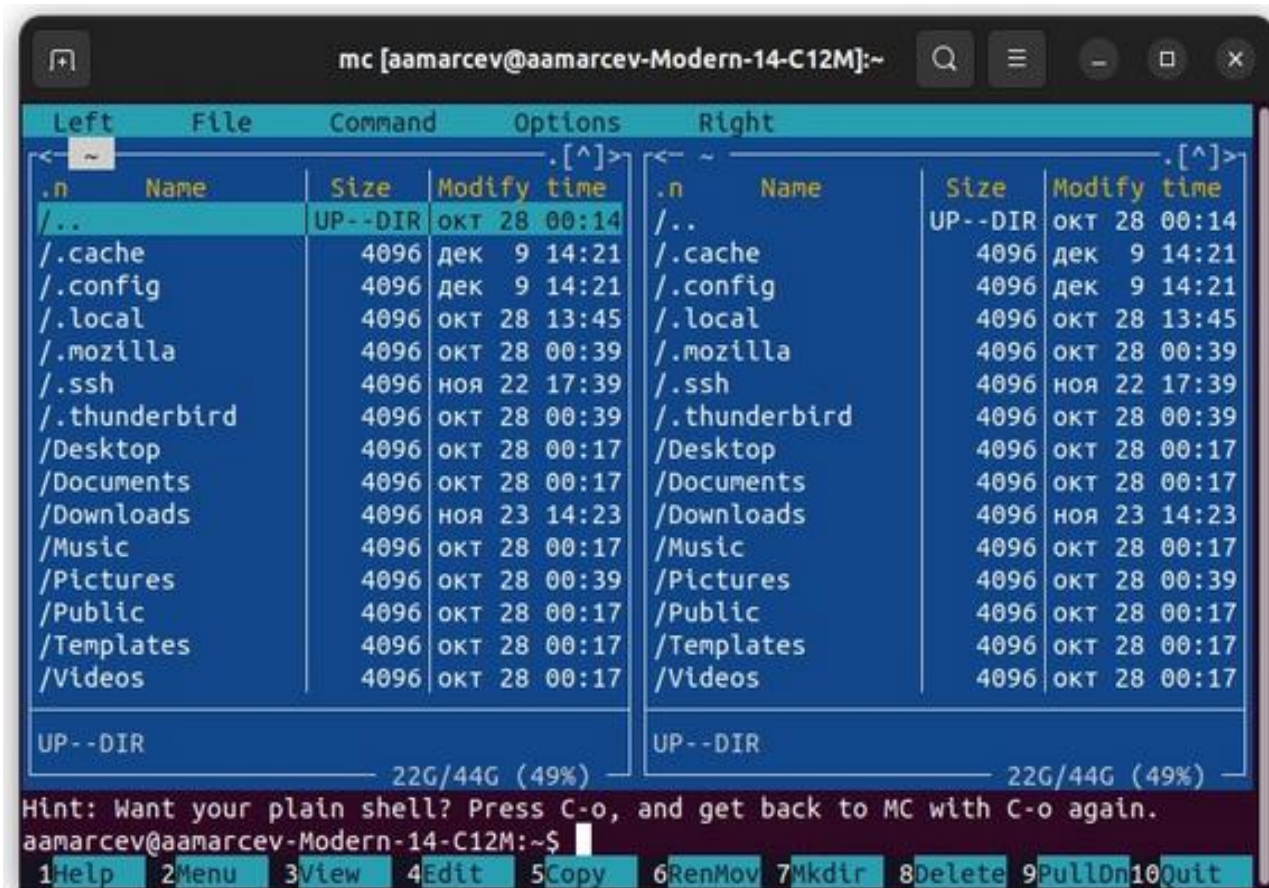


рис 1

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

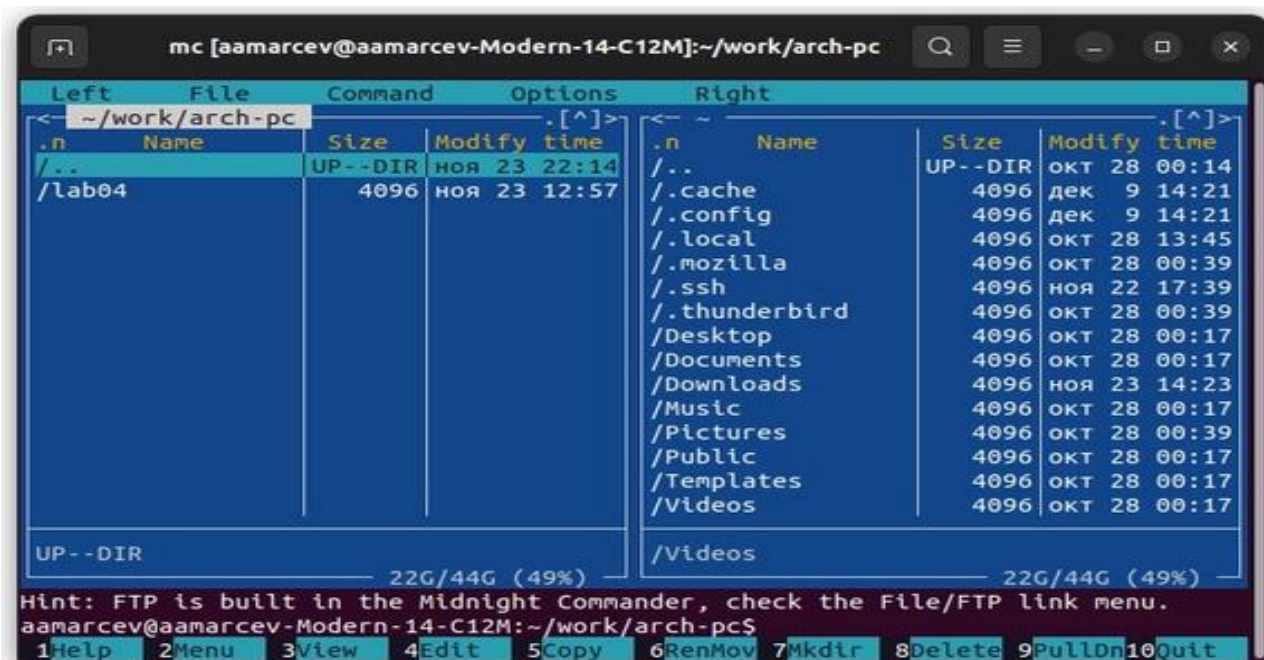
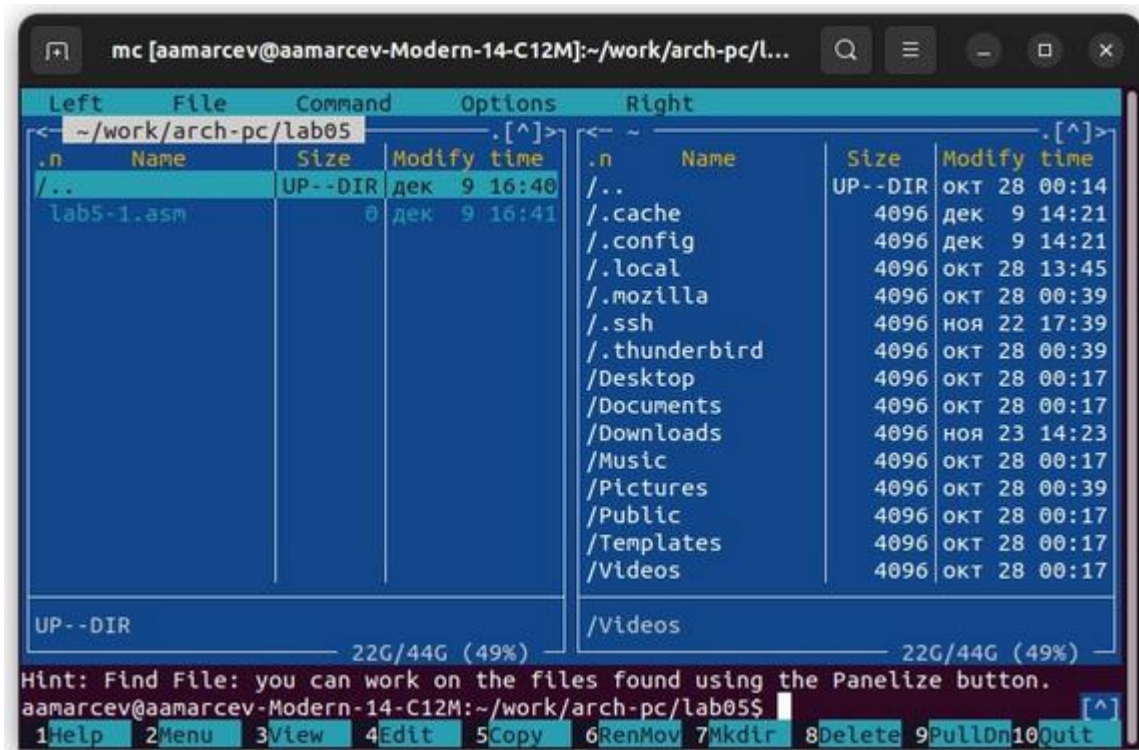


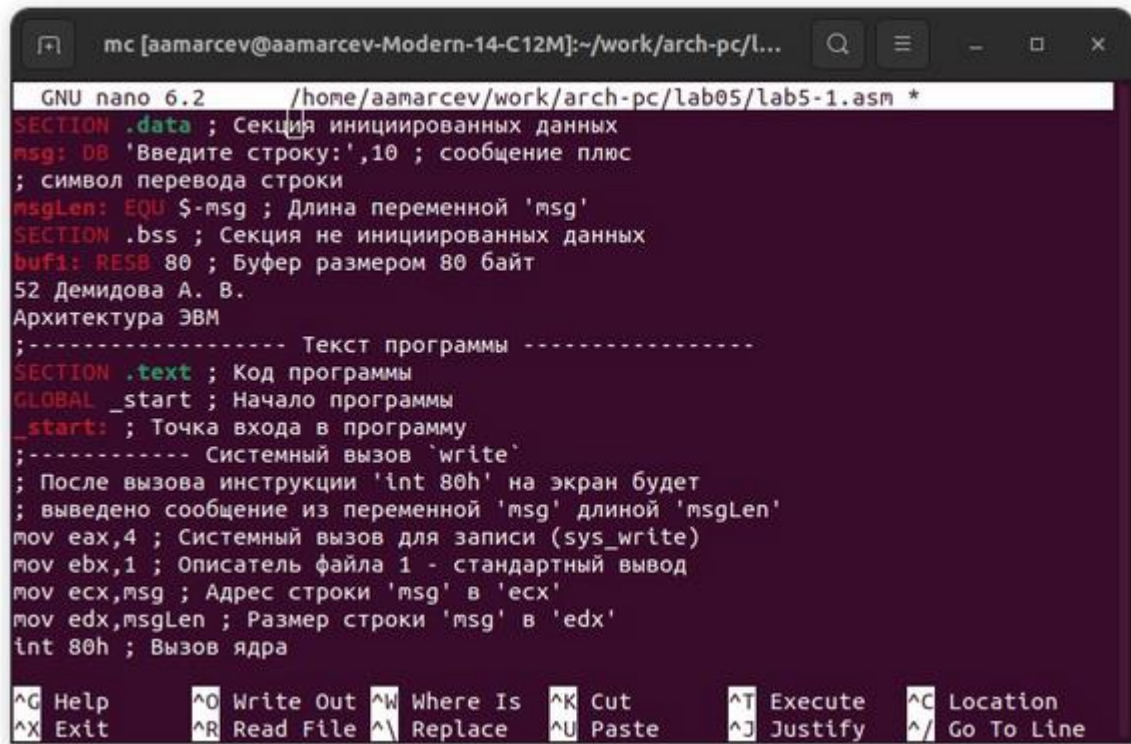
рис 2

С помощью функциональной клавиши F7 создаю каталог. Перехожу в созданный каталог. В строке ввода прописываю команду touch lab5-1.asm, чтобы создать файл, в котором буду работать.



Структура программы на языке ассемблера NASM

С помощью функциональной клавиши F4 открываю созданный файл для редактирования в редакторе nano. Ввожу в файл код программы для запроса строки у пользователя. Далее выхожу из файла, сохраняя изменения. С помощью функциональной клавиши F3 открываю файл для просмотра, чтобы проверить, содержит ли файл текст программы.

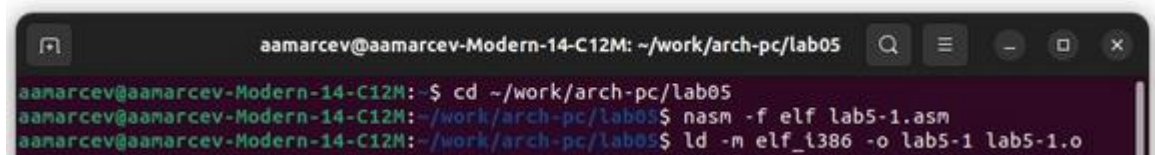


```
GNU nano 6.2 /home/aamarcev/work/arch-pc/lab05/lab5-1.asm *
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не инициализированных данных
buf1: RESB 80 ; Буфер размером 80 байт
52 Демидова А. В.
Архитектура ЭВМ
;----- 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

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

Рис 4

Транслирую текст в объектный файл при помощи команд `nasm -f elf lab5-1.asm` и `ld -m elf_i386 -o lab5-1 lab5-1.o`



```
aamarcev@aamarcev-Modern-14-C12M: ~/work/arch-pc/lab05
aamarcev@aamarcev-Modern-14-C12M:~/work/arch-pc/lab05$ cd ~/work/arch-pc/lab05
aamarcev@aamarcev-Modern-14-C12M:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
aamarcev@aamarcev-Modern-14-C12M:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
```

Рис 5

Запускаю исполняемый файл. Программа выводит строку “Введите строку:” и ждет ввода с клавиатуры, я ввожу свои ФИО, на этом программа заканчивает свою работу.

```
amarcev@amarcev-Modern-14-C12M:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Марцев Аркадий Алексеевич
amarcev@amarcev-Modern-14-C12M:~/work/arch-pc/lab05$
```

Рис 6

Подключение внешнего файла

Скачиваю файл in_out.asm со страницы курса в ТУИС. Он сохранился в каталог “Загрузки”. С помощью функциональной клавиши F5 копирую файл in_out.asm из каталога Загрузки в созданный каталог lab05.

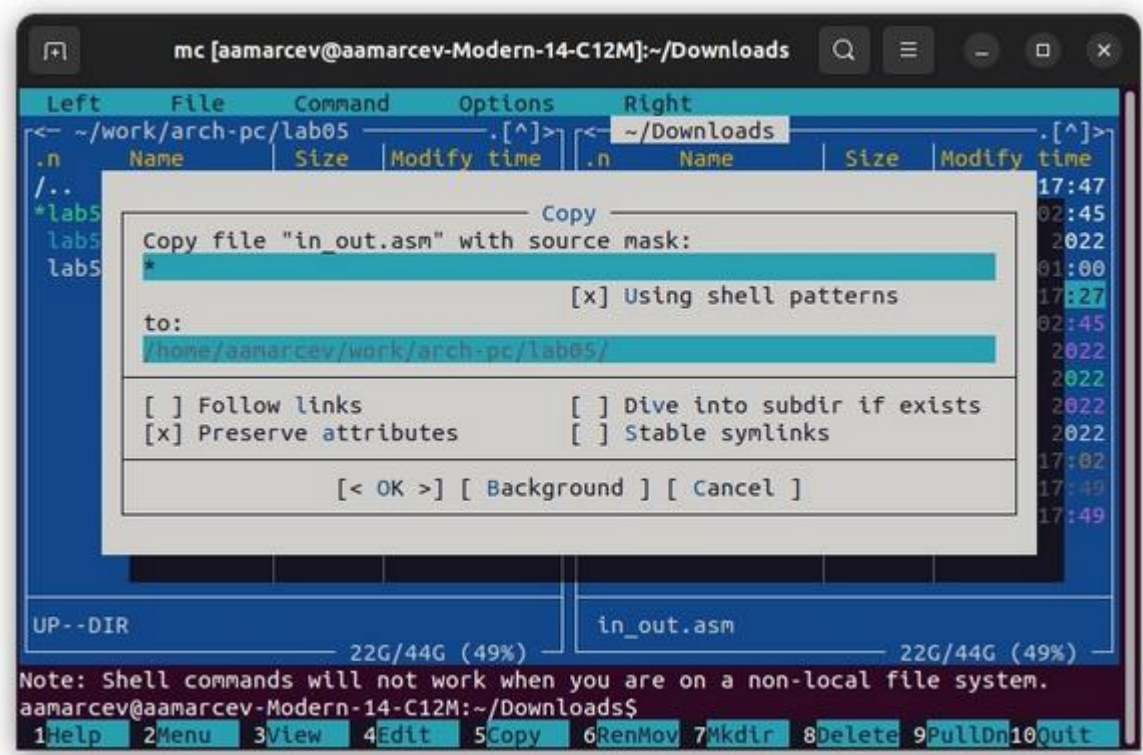
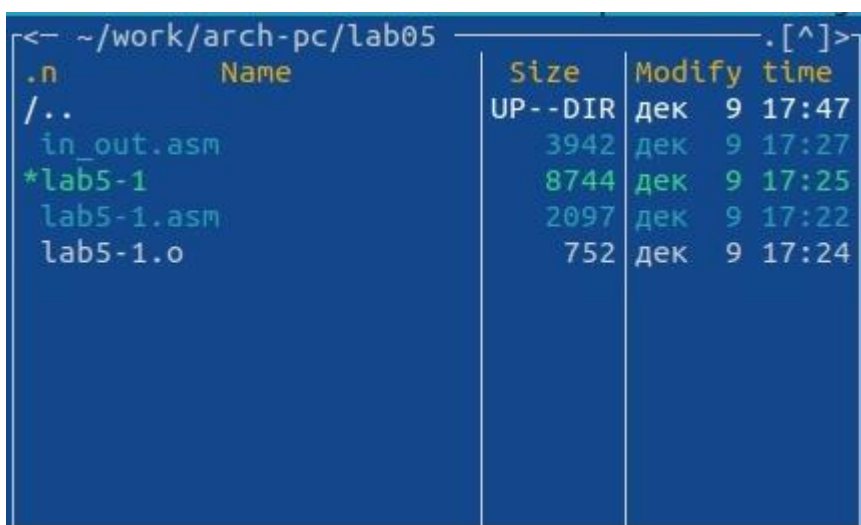


Рис 7

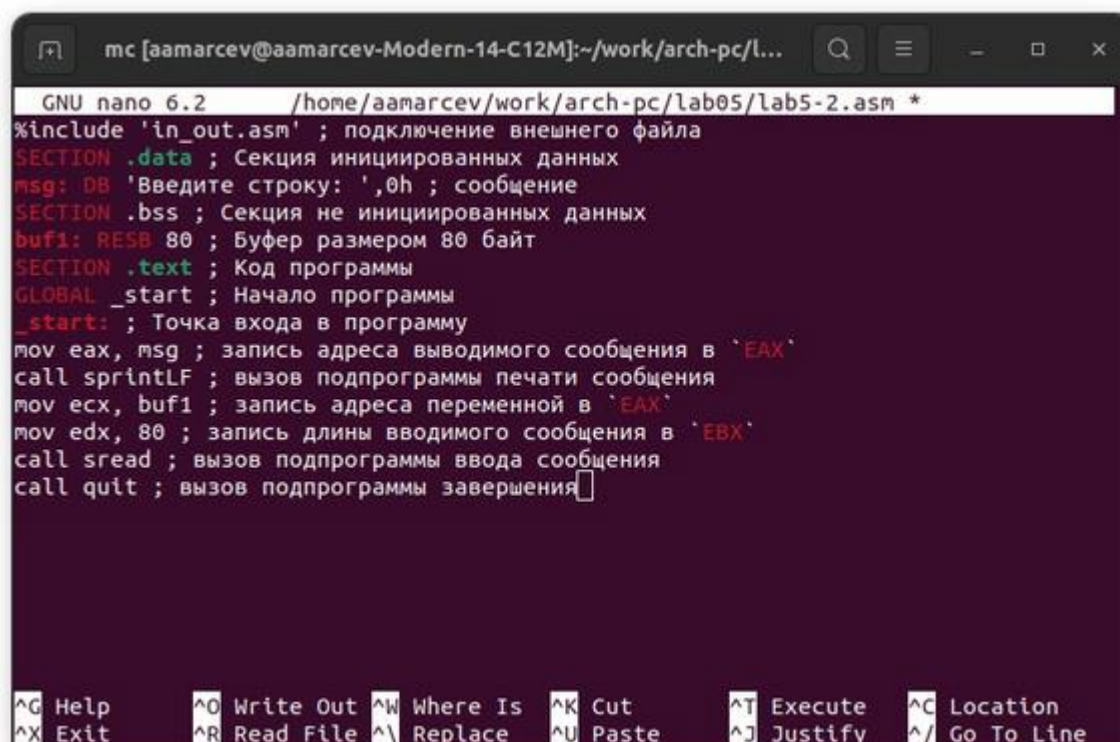
С помощью функциональной клавиши F5 копирую файл lab5-1 в тот же каталог с тем же именем и изменяю его на lab5-2.



ino	Name	Size	Modify	time
UP--DIR	../		дек 9	17:47
3942	in_out.asm		дек 9	17:27
8744	*lab5-1		дек 9	17:25
2097	lab5-1.asm		дек 9	17:22
752	lab5-1.o		дек 9	17:24

Рис 8

Изменяю содержимое файла lab5-2.asm во встроенном редакторе nano (рис. 15), чтобы в программе использовались подпрограммы из внешнего файла in_out.asm.



```
GNU nano 6.2 /home/aamarcev/work/arch-pc/lab05/lab5-2.asm *
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку: ',0h ; сообщение
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 в `EAX`
mov edx, 80 ; запись длины вводимого сообщения в `EBX`
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения
```

Рис 9

Транслирую текст программы файла в объектный файл командой `nasm -f elf lab5-2.asm`. Создался объектный файл `lab5-2.o`. Выполняю компоновку объектного файла с помощью команды `ld -m elf_i386 -o lab5-2 lab5-2.o` Создался исполняемый файл `lab5-2`. Запускаю исполняемый файл.

```

aamarcev@aamarcev-Modern-14-C12M: ~/work/arch-pc/lab05
aamarcev@aamarcev-Modern-14-C12M:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
aamarcev@aamarcev-Modern-14-C12M:~/work/arch-pc/lab05$ ld -m elfi386 -o lab5-2 lab5-2.o
ld: unrecognized emulation mode: elfi386
Supported emulations: elf_x86_64 elf32_x86_64 elf_i386 elf_iamcu elf_l1om elf_k1om i386pep i386pe
aamarcev@aamarcev-Modern-14-C12M:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
aamarcev@aamarcev-Modern-14-C12M:~/work/arch-pc/lab05$

```

Рис 10

Разница между первым исполняемым файлом `lab5-2` использующим `sprintf` и `sprint` в том, что запуск первого запрашивает ввод с новой строки, а программа, которая выполняется при запуске второго, запрашивает ввод без переноса на новую строку.

Выполнение заданий для самостоятельной работы

Создаю копии файлов `lab5-1-1` и `lab5-2-1` которые нужны будут для самостоятельной работы.

Left	File	Command	Options	Right
<-	~/work/arch-pc/lab05		.[^]>	<- ~/work/arch-pc
.n	Name	Size	Modify time	.n Name Size
/..		UP--DIR	дек 9 18:30	/.. UP--DIR
in	out.asm	3942	дек 9 17:27	/lab04 4096
	lab5-1-1.asm	2097	дек 9 17:22	/lab05 4096
	lab5-2-1.asm	962	дек 9 21:25	

Рис 11

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

```
GNU nano 6.2 /home/aamarcev/work/arch-pc/lab05/lab5-1-1.asm
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
```

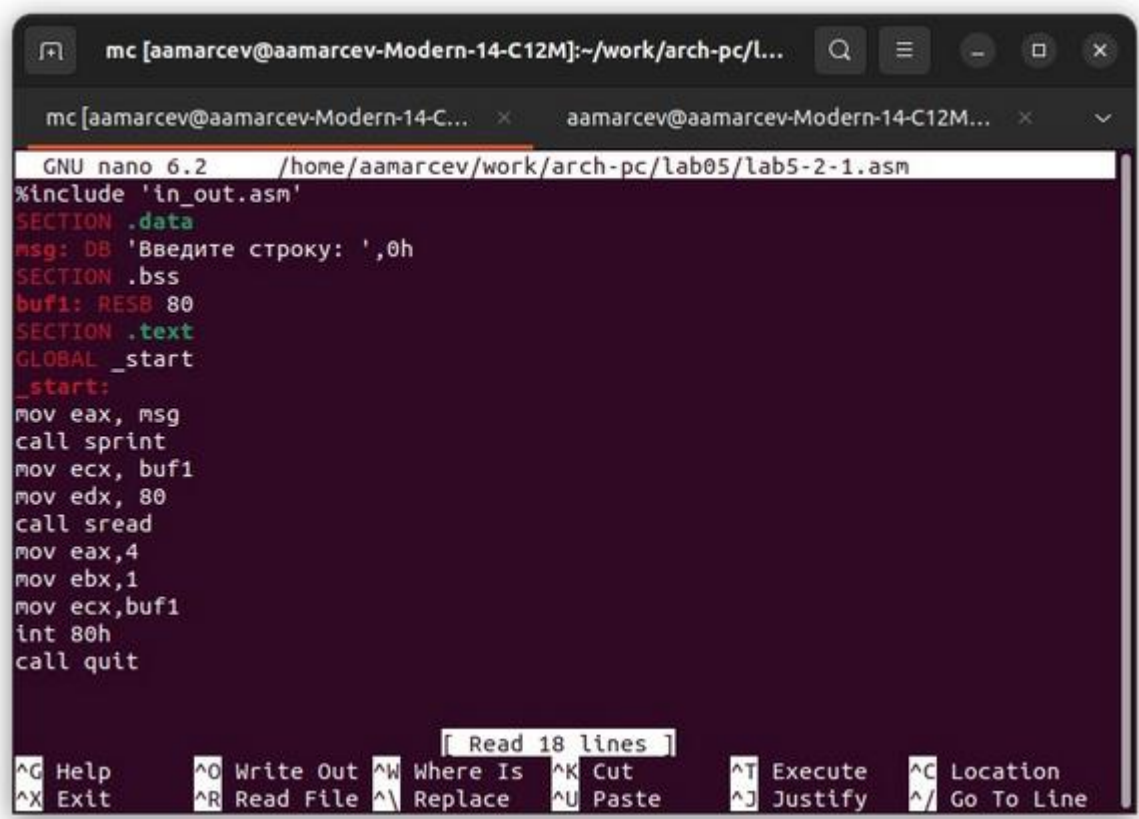
Рис 12

Работа программы созданной для первого пункта самостоятельной работы

```
aamarcev@aamarcev-Modern-14-C12M:~/work/arch-pc/lab05$ ./lab5-1-1
Введите строку
Martsev
Martsev
aamarcev@aamarcev-Modern-14-C12M:~/work/arch-pc/lab05$
```

Рис
13

Редактирую код программы lab5-2-1 для того чтоб он выводил строку введенную пользователем.

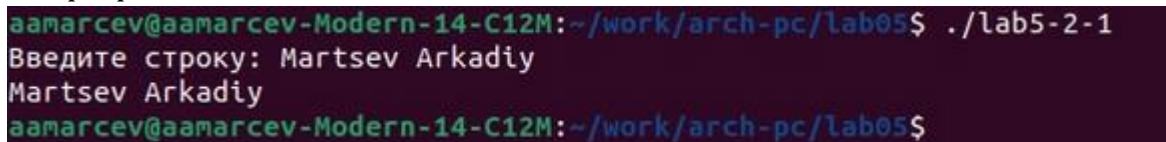


```
mc [aamarcev@aamarcev-Modern-14-C12M]:~/work/arch-pc/L...
mc [aamarcev@aamarcev-Modern-14-C... x aamarcev@aamarcev-Modern-14-C12M... x
GNU nano 6.2 /home/aamarcev/work/arch-pc/lab05/lab5-2-1.asm
#include 'in_out.asm'
SECTION .data
msg: DB 'Введите строку: ',0h
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprint
mov ecx, buf1
mov edx, 80
call sread
mov eax, 4
mov ebx, 1
mov ecx, buf1
int 80h
call quit

[ 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
```

Рис 14

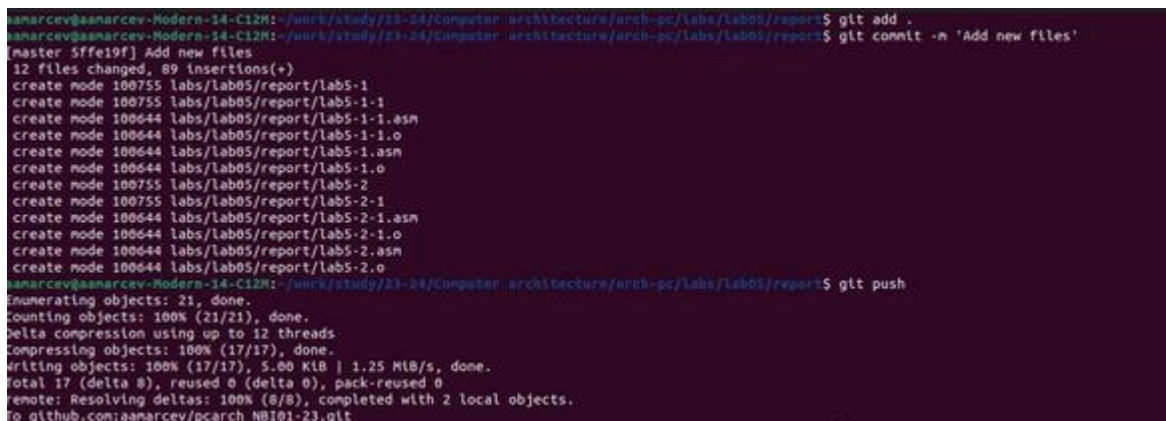
Работа программы.



```
aamarcev@aamarcev-Modern-14-C12M:~/work/arch-pc/lab05$ ./lab5-2-1
Введите строку: Martsev Arkadiy
Martsev Arkadiy
aamarcev@aamarcev-Modern-14-C12M:~/work/arch-pc/lab05$
```

Рис 15

Отправляю сделанные во время выполнения лабораторной работы файлы на GitHub при помощи команд git.



```
aamarcev@aamarcev-Modern-14-C12M:~/work/study/23-24/Computer_architecture/arch-pc/labs/lab05/report$ git add .
aamarcev@aamarcev-Modern-14-C12M:~/work/study/23-24/Computer_architecture/arch-pc/labs/lab05/report$ git commit -m 'Add new files'
[master 5ffe19f] Add new files
12 files changed, 89 insertions(+)
create mode 100755 labs/lab05/report/lab5-1
create mode 100755 labs/lab05/report/lab5-1-1
create mode 100644 labs/lab05/report/lab5-1-1.asm
create mode 100644 labs/lab05/report/lab5-1-1.o
create mode 100644 labs/lab05/report/lab5-1.asm
create mode 100644 labs/lab05/report/lab5-1.o
create mode 100755 labs/lab05/report/lab5-2
create mode 100755 labs/lab05/report/lab5-2-1
create mode 100644 labs/lab05/report/lab5-2-1.asm
create mode 100644 labs/lab05/report/lab5-2-1.o
create mode 100644 labs/lab05/report/lab5-2.asm
create mode 100644 labs/lab05/report/lab5-2.o
aamarcev@aamarcev-Modern-14-C12M:~/work/study/23-24/Computer_architecture/arch-pc/labs/lab05/report$ git push
Enumerating objects: 21, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 12 threads.
Compressing objects: 100% (17/17), done.
Writing objects: 100% (17/17), 5.00 KiB | 1.25 MiB/s, done.
Total 17 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (8/8), completed with 2 local objects.
To github.com:aamarcev/pcarch_NB101-23.git
```

Вы
во
ды
При
вып
олн
ени

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

Список литературы

1. Лабораторная работа №5