

Отчёт по лабораторной работе №5

Дисциплина: архитектура компьютера

Булыгин Николай Александрович

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Midnight Commander	7
3.2	Подключение внешнего файла in_out.asm	9
3.3	Выполнение самостоятельной работы	11
4	Выводы	15

Список иллюстраций

3.1	Midnight Commander	7
3.2	Код программы	8
3.3	Просмотр	8
3.4	Работа программы	9
3.5	in_out.asm	9
3.6	Новый код	10
3.7	Проверка	11
3.8	sprint	11
3.9	task1.asm	13
3.10	task2.asm	14

Список таблиц

1 Цель работы

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

2 Задание

1. Midnight Commander
2. Подключение внешнего файла in_out.asm
3. Выполнение самостоятельной работы

3 Выполнение лабораторной работы

3.1 Midnight Commander

Открываю Midnight Commander, перехожу в каталог `~/work/arch-pc`, создаю там папку `lab05` и в ней создаю файл `lab5-1.asm` (рис. 3.1).

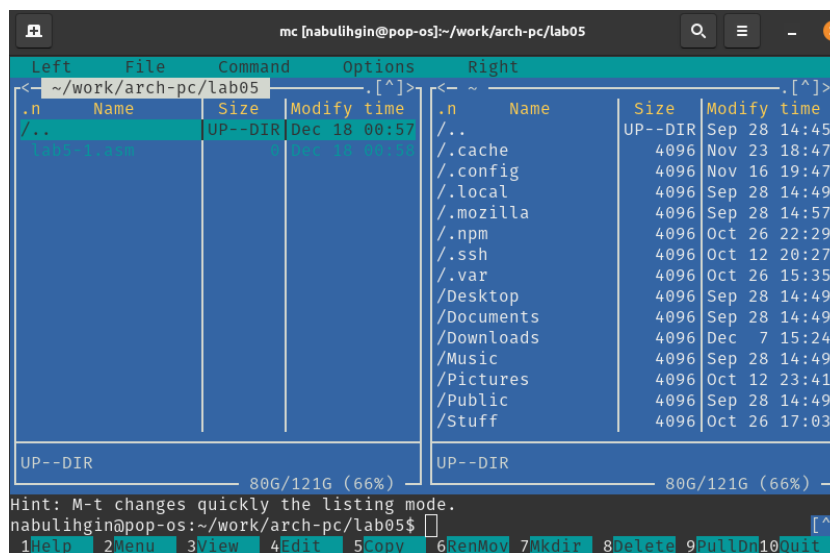
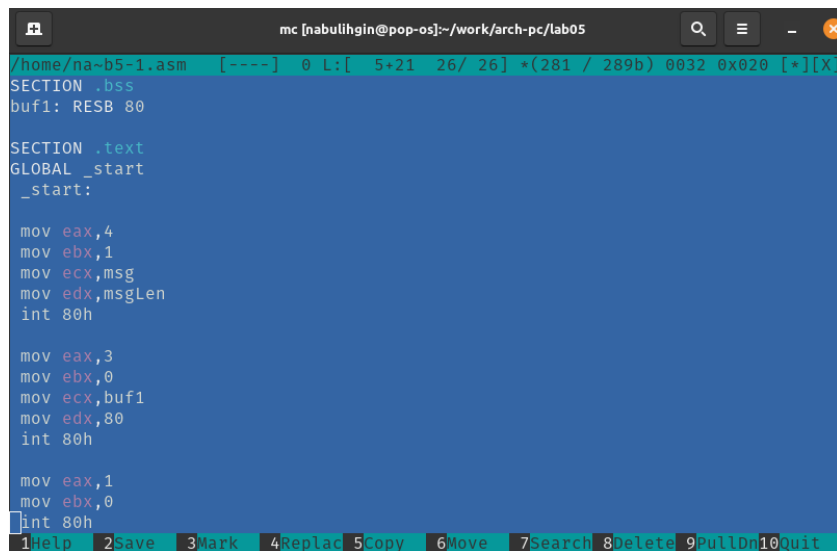


Рис. 3.1: Midnight Commander

Открываю созданный файл во встроенном редакторе `mcedit` и ввожу в него данный код (рис. 3.2).



```
mc [nabulihgin@pop-os]~/work/arch-pc/lab05
/home/na-b5-1.asm [----] 0 L: [ 5+21 26/ 26] *(281 / 289b) 0032 0x020 [*][X]
SECTION .bss
buf1: RESB 80

SECTION .text
GLOBAL _start
_start:

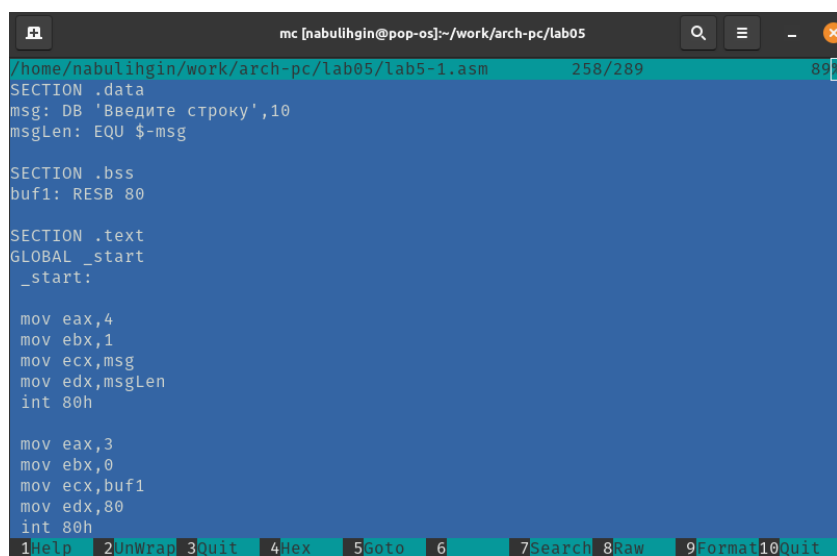
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h

mov eax,3
mov ebx,0
mov ecx,buf1
mov edx,80
int 80h

mov eax,1
mov ebx,0
int 80h
1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn10Quit
```

Рис. 3.2: Код программы

Сохраняю файл, выхожу из редактора и в режиме просмотра убеждаюсь в наличии кода программы (рис. 3.3).



```
mc [nabulihgin@pop-os]~/work/arch-pc/lab05
/home/nabulihgin/work/arch-pc/lab05/lab5-1.asm 258/289 89%
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
1Help 2UnWrap 3Quit 4Hex 5Goto 6 7Search 8Raw 9Format10Quit
```

Рис. 3.3: Просмотр

Транслирую lab5-1.asm в объектный файл, выполняю его компоновку и запускаю получившийся исполняемый файл. На запрос ввода строки ввожу свои инициалы (рис. 3.4).


```
/home/na~b5-2.asm [-M--] 1
%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. 3.6: Новый код

Создаю исполняемый файл и проверяю его работу (рис. 3.7).

```
nabulihgin@pop-os:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
nabulihgin@pop-os:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
nabulihgin@pop-os:~/work/arch-pc/lab05$ ./lab5-2
Введите строку
Булыгин Николай Александрович
nabulihgin@pop-os:~$
```

Рис. 3.7: Проверка

Меняю `sprintLF` на `sprint`, теперь текст выводится на одной строке т.к. `sprintLF` работает так же, как и `sprint`, но добавляет к сообщению символ перевода строки (рис. 3.8).

```
nabulihgin@pop-os:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
nabulihgin@pop-os:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
nabulihgin@pop-os:~/work/arch-pc/lab05$ ./lab5-2
Введите строкуБулыгин Николай Александрович
nabulihgin@pop-os:~$
```

Рис. 3.8: `sprint`

3.3 Выполнение самостоятельной работы

Создаю копию файла `lab5-1.asm` под названием `task1.asm` и меняю код программы, чтобы она возвращала написанную строчку. Для вывода строки буду использовать вызов `sys_write`.

Код программы:

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

; sys_write
mov eax,4
mov ebx,1
int 80h

mov eax,1
mov ebx,0
int 80h

```

Получаю исполняемый файл и проверяю работу написанной программы. Теперь она также возвращает написанную строку (рис. 3.9).

```
nabulihgin@pop-os:~/work/arch-pc/lab05$ nasm -f elf task1.asm
nabulihgin@pop-os:~/work/arch-pc/lab05$ ld -m elf_i386 -o task1 task1.o
nabulihgin@pop-os:~/work/arch-pc/lab05$ ./task1
Введите строку
Булыгин Николай Александрович
Булыгин Николай Александрович
nabulihgin@pop-os:~$
```

Рис. 3.9: task1.asm

Аналогично первому заданию создаю копию файла lab5-2.asm под названием task2.asm и меняю код программы, чтобы она возвращала написанную строку.

Код программы:

```
%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,buf1
```

call sprintLF ; *Вывод строки*

call quit

Получаю исполняемый файл и проверяю работу написанной программы. Первый ввод идет без перевода строки т.к. в ходе лабораторной работы мы поменяли sprintLF на sprint в коде программы, которую скопировали (рис. 3.10).

```
nabulihgin@pop-os:~/work/arch-pc/lab05$ nasm -f elf task2.asm
nabulihgin@pop-os:~/work/arch-pc/lab05$ ld -m elf_i386 -o task2 task2.o
nabulihgin@pop-os:~/work/arch-pc/lab05$ ./task2
Введите строкуБулыгин Николай Александрович
Булыгин Николай Александрович
```

Рис. 3.10: task2.asm

4 Выводы

Я приобрел практические навыки работы в Midnight Commander и освоил инструкции языка ассемблера `mov` и `int`.