Лабораторная работа №5

НКАбд-02-23

Выборнов Дмитрий Валерьевич

1 Цель работы

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

2 Задание

- 1. Основы работы с Midnight Commander.
- 2. Подключение внешнего файла в NASM.
- 3. Задания для самостоятельной работы.

3 Теоретическое введение

3.1 Midnight Commander

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.

3.2 **Ассемблер NASM**

Программа на языке ассемблера NASM, как правило, состоит из трёх секций: секция кода программы (SECTION .text), секция инициированных (известных во время компиляции) данных (SECTION .data) и секция неинициализированных данных (тех, под которые во время компиляции только отводится память, а значение присваивается в ходе выполнения программы) (SECTION .bss).

4 Выполнение лабораторной работы

4.1 Основы работы с Midnight Commander.

Открываю тс и перехожу в нужный каталог.

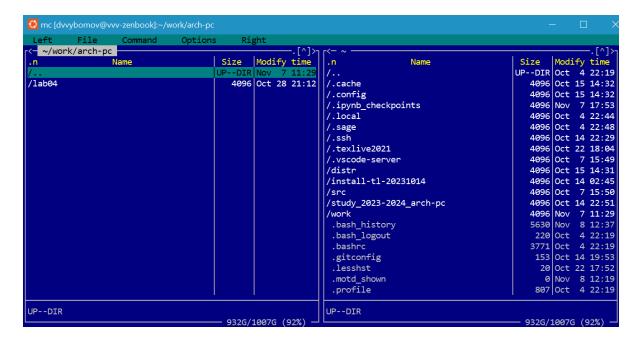


Рис. 4.1: Первый шаг.

Создаю новый каталог и перехожу в него.

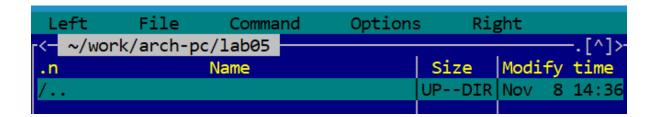


Рис. 4.2: Второй шаг.

Создаю файл lab5-1.asm.

Left	File	Commai	nd	Options	Rig	ght	
_<- ~/	work/arch-pc	/lab05					—.[^]> _Ղ
.n		Name			Size	Modif	y time
/					UPDIR	Nov 8	8 14:36
lab5	-1.asm				0	Nov 8	8 14:37

Рис. 4.3: Третий шаг.

Открываю созданный мной файл и ввожу необходимый текст программы.

```
GNU nano 6.2 /home/dvvybornov/work/arch-pc/lab05/lab5-1.asm
      .data
    DB 'Введите строку:', 10
   Len: EQU $-msg
         .bss
       RESB 80
     N .text
     _start
     eax,4
mov
     ebx,1
mov
mov
     ecx,msg
    edx,msgLen
mov
int
     80h
     eax, 3
mov
     ebx, 0
mov
     ecx, buf1
mov
     edx, 80
mov
int
     80h
     eax,1
mov
     ebx,0
mov
int
     80h
```

Рис. 4.4: Четвёртый шаг.

Проверяю, что изменения сохранены.

```
/home/dvvybornov/work/arch-pc/lab05/lab5-1.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,1
mov ebx,0
     80h
int
```

Рис. 4.5: Пятый шаг.

Превращаю текст программы lab5-1.asm в объектный файл, выполняю компоновку объектного файла и запускаю получившийся исполняемый файл.

dvvybornov@vvv-zenbook:~/work/arch-pc/lab05\$ nasm -f elf lab5-1.asm

Рис. 4.6: Первая часть шестого шага.

```
dvvybornov@vvv-zenbook:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o dvvybornov@vvv-zenbook:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Vybornov Dmitry
dvvybornov@vvv-zenbook:~/work/arch-pc/lab05$
```

Рис. 4.7: Вторая часть шестого шага.

4.2 Подключение внешнего файла в NASM.

Скачайте файл in_out.asm со страницы курса и перемещаю его в нужный каталог.

```
~/work/arch-pc/lab05 -
                                       Size
                                             Modify time
                 Name
                                      UP--DIR Nov
                                                   8 14:36
                                         3942 Nov 11 15:34
in_out.asm:Zone.Identifier
                                          192 Nov 11 15:34
*lab5-1
                                         8744 Nov 8 15:07
                                          335 Nov 8 14:50
                                          752 Nov 8 15:06
lab5-1.o
in_out.asm:Zone.Identifier
                                        932G/1007G (92%)
```

Рис. 4.8: Седьмой шаг.

Создаю копию файла lab5-1.asm.

	ions I	Right	
~/work/arch-pc/lab05	1 .	1	−.[^]>ղ
.n Name	Size		
/··			3 14:36
in_out.asm		42 Nov 11	
in_out.asm:Zone.Identifier		92 Nov 11	
*lab5-1		44 Nov 8	
lab5-1.asm			3 14:50
lab5-1.o	_	_	15:06
lab5-2.asm	33	35 Nov 8	14:50
lab5-2.asm			(()
	9320	G/1007G ((92%)

Рис. 4.9: Восьмой шаг.

Вношу необходимые изменения в текст программы.

```
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите строку: ', 0h
 SECTION .bss
 buf1: RESB 80
  CTION .text
 LOBAL _start
_start:
mov eax, msg
 call sprintLF
mov ecx, buf1
mov edx, 80
call sread
call quit
```

Рис. 4.10: Девятый шаг.

Создаю исполняемый файл и проверяю его работу.

```
dvvybornov@vvv-zenbook:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Dmitry Vybornov
dvvybornov@vvv-zenbook:~/work/arch-pc/lab05$
```

Рис. 4.11: Десятый шаг.

Создаю ещё одну копию файла lab5-2.asm.

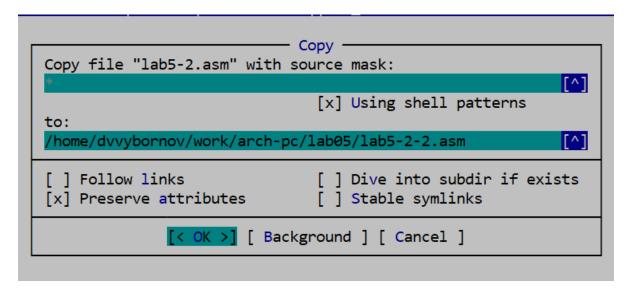


Рис. 4.12: Одиннадцатый шаг.

Проверяю работу программы, заменив sprintLF на sprint.

```
dvvybornov@vvv-zenbook:~/work/arch-pc/lab05$ ./lab5-2-2
Введите строку: Dmitry Vybornov
dvvybornov@vvv-zenbook:~/work/arch-pc/lab05$ _
```

Рис. 4.13: Двенадцатый шаг.

Разница между этими двумя вариантами программы в том, что, в отличии от sprint, при использовании sprintLF ввод имени пользователя начинается с новой строки.

4.3 Задания для самостоятельной работы

1. Создаю ещё одну копию файла lab5-1.asm.

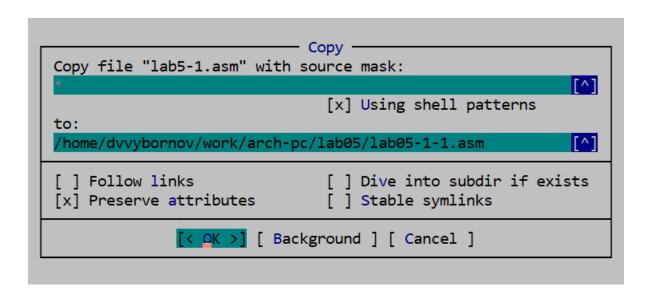


Рис. 4.14: Первый шаг первого задания.

Изменяю программу так, чтобы она выводила имя пользователя.

```
CTION .data
    DB 'Введите строку:', 10
        EQU $-msg
          .bss
            80
      .text
      start
    eax, 4
mov
    ebx, 1
mov
    ecx, msg
mov
mov edx, msgLen
int
     80h
    eax, 3
mov
mov ebx, 0
    ecx, buf1
mov
    edx, 80
mov
int
     80h
    eax, 4
mov
    ebx, 1
mov
     ecx, buf1
mov
     edx, buf1
mov
int
     80h
    eax, 1
mov
     ebx, 0
mov
int
     80h_
```

Рис. 4.15: Второй шаг первого задания.

2. Получаю исполняемый файл и проверяю его работу.

```
dvvybornov@vvv-zenbook:~/work/arch-pc/lab05$ nasm -f elf lab5-1-1.asm dvvybornov@vvv-zenbook:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1-1 lab5-1-1.o dvvybornov@vvv-zenbook:~/work/arch-pc/lab05$ ./lab5-1-1
Введите строку:
Dmitry Vybornov
Dmitry Vybornov
dvvybornov@vvv-zenbook:~/work/arch-pc/lab05$ _
```

Рис. 4.16: Второе задание.

3. Создаю копию файла lab5-2.asm и изменяю текст программы так, чтобы она выводила имя пользователя.

```
%include 'in_out.asm'
  CTION .data
 msg: DB 'Введите строку: ', 0h
  CTION .bss
 buf1: RESB
             80
  CTION .text
  OBAL _start
 _start:
mov eax, msg
call sprintLF
mov ecx, buf1
mov edx, 80
call sread
mov eax, 4
mov ebx, 1
mov ecx, buf1
int 80h
call quit
```

Рис. 4.17: Третье задание.

4. Создаю исполняемый файл и проверяю его работу.

```
dvvybornov@vvv-zenbook:~/work/arch-pc/lab05$ nasm -f elf lab5-2-3.asm
dvvybornov@vvv-zenbook:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2-3 lab5-2-3.o
dvvybornov@vvv-zenbook:~/work/arch-pc/lab05$ ./lab5-2-3
Введите строку:
Dmitry Vybornov
Dmitry Vybornov
dvvybornov@vvv-zenbook:~/work/arch-pc/lab05$
```

Рис. 4.18: Четвёртое задание.

5 Выводы

Выполнив эту лабораторную работу, я приобрёл практические навыки работы в Midnight Commander и освоил инструкции языка ассемблера mov и int.