# Лабораторная работа 5

#### Отчёт

Соловьев Богдан Михайлович

#### Содержание

## 1 Цель работы

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

## 2 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. Для активации оболочки Midnight Commander достаточно ввести в командной строке mc и нажать клавишу Enter (рис. 5.1). В Midnight Commander используются функциональные клавиши F1 — F10, к которым привязаны часто выполняемые операции.

Архитектура ЭВМ Функцио- нальные клавиши Выполняемое действие F5 копирование файла или группы отмеченных файлов из каталога, отображаемого в активной панели, в каталог, отображаемый на второй панели F6 перенос файла или группы отмеченных файлов из каталога, отображаемого в активной панели, в каталог, отображаемый на второй панели F7 создание подкаталога в каталоге, отображаемом в активной панели F8 удаление файла (подкаталога) или группы отмеченных файлов F9 вызов основного меню программы F10 выход из программы

Следующие комбинации клавиш облегчают работу с Midnight Commander: • Таb используется для переключениями между панелями; • Ти Гиспользуется 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 + о (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 содер- жимое каталогов, отображаемых на левой и правой панелях.

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). Таким образом, общая структура программы имеет следующий вид:

```
SECTION .data ; Секция содержит переменные, для ... ; которых задано начальное значение
```

## 3 Выполнение лабораторной работы

## 3.1 Лабораторная работа

Открываю Midnight Commabnder (предварительно скачав библиотеку) написав в командной строке mc (рис. 1).

и Имя	Размер	Rnew		[^]>- правки		Размер	Rne		[^]> правки
, m mmm	-BBEPX-			19:33	/	-BBEPX-			19:33
.cache	476	ноя		13:08	/.cache	476			13:08
config.	512			13:08	/.config	512			13:08
/.local	32			23:35	/.local	32			23:35
/.mozilla	48			23:09	/.mozilla	48			23:09
/.ssh	84			23:31	/.ssh	84			23:31
texlive2023	18			23:57	/.texlive2023	18			23:57
.var	6			20:08	/.var	6			20:08
/work	24			15:27	/work	24			15:27
/Видео	0	сен	13	19:33	/Видео	Θ	сен	13	19:33
/Документы	0	сен	13	19:33	/Документы	Θ	сен	13	19:33
/Загрузки	418	окт		12:34	/Загрузки	418	окт		12:34
/Изображения	50	сен	14	20:33	/Изображения	50	сен	14	20:33
/Музыка		сен	13	19:33	/Музыка		сен	13	19:33
/Общедоступные		сен	13	19:33	/Общедоступные		сен	13	19:33
/Рабочий стол	26	окт		15:28	/Рабочий стол	26	окт		15:28
/Шаблоны		сен	13	19:33	/Шаблоны		сен	13	19:33
.bash_history	10572	окт	18	16:45	.bash_history	10572	окт	18	16:45
.bash_logout	18	фев		2023	.bash_logout	18	фев		2023
.bash_profile	141	фев		2023	.bash_profile	141	фев		2023
.bashrc	492	фев		2023	.bashrc	492	фев		2023
.gitconfig	153			22:27	.gitconfig	153	сен	23	22:27
.vboxclient-clipboard.pid				16:13	.vboxclient-clipboard.pid				16:13
.vboxclient-draganddrop.pid				16:13	.vboxclient-draganddrop.pid				16:13
.vboxclient-seamless.pid	5	окт	11	16:13	.vboxclient-seamless.pid	5	окт	11	16:13
-BBEPX-					-вверх-				
	— 3920M ,		_			3920M	/ 19	G (	20%) -
овет: Автодополнение: M-Tab (ил omsolovjev@fedora ~]\$ █	и Esc+Tab)	). Дл	ЯГ	олучен	ия списка нажать дважды.				

Figure 1: Внешний вид тс

Создаю каталог lab05 с помощью встроенных функций mc(рис. 2).

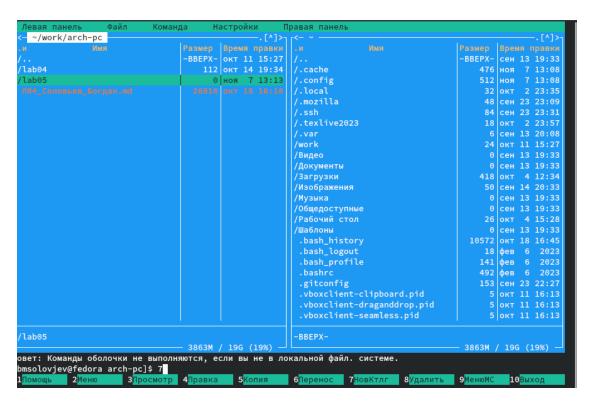


Figure 2: Παnκα lab05

Ввожу в созданных файл код программы, которая будет запрашивать текст (рис. 3).

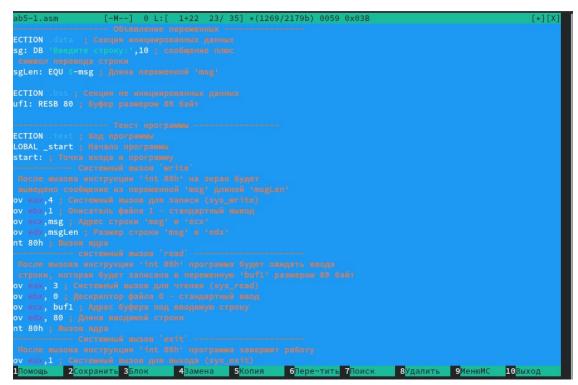


Figure 3: Код программы

Создаю объектный и сполняемый файл с помощью -f elf lab5-1.asm и lab5-1.o (рис. 4).

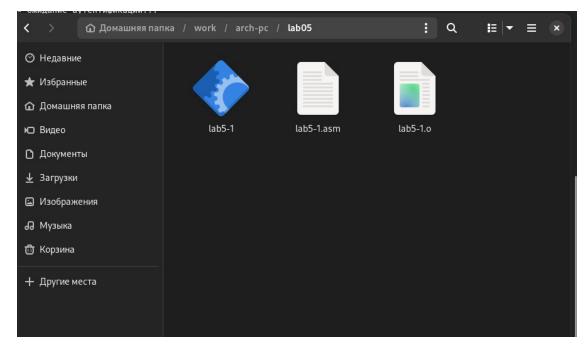


Figure 4: Объектный и исполняемый файл

Запускаю исполняемый файл (рис. 5).

```
bmsolovjev@fedora lab05]$ ./lab5-1
ведите строку:
оловьев Богдан Михайлович
```

Figure 5: Программа запрашивает строку

Скачиваю файл in\_out.asm и переношу его в каталог lab05 (рис. 6).

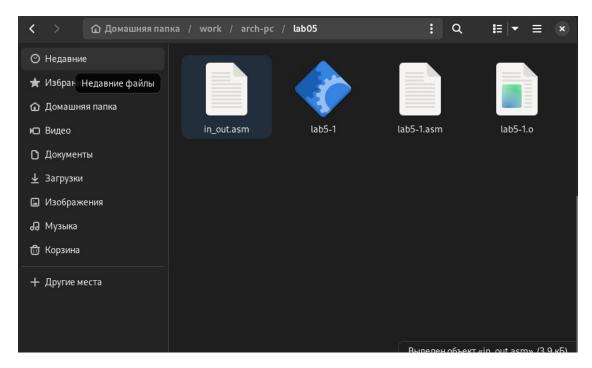


Figure 6: Файл in\_out.asm

Копирую файл lab5-1.asm и изменяю его название (рис. 7).

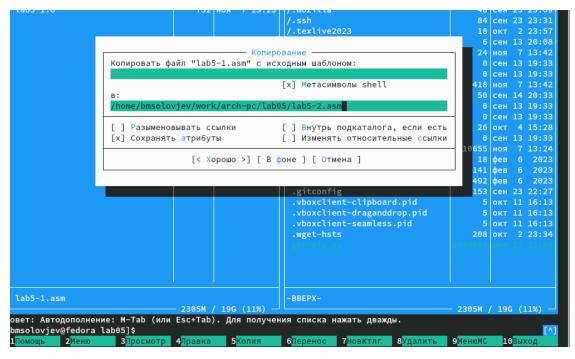


Figure 7: Копирование и переименовывание файла

Изменяю текст программы lab5-2.asm (рис. 8).

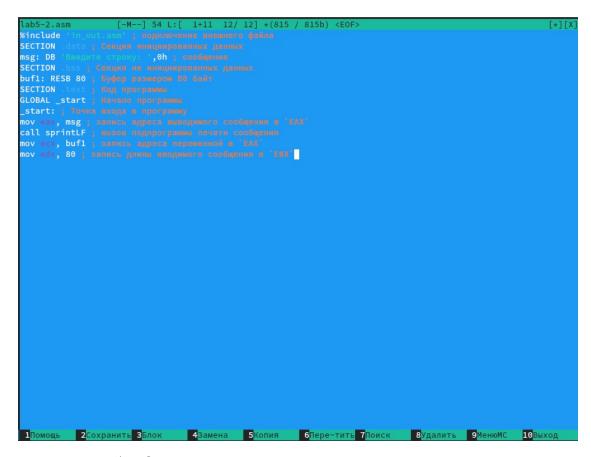


Figure 8: Новый код программы

Создание объектного и исполняемого файла и проверка работы команды (рис. 9).

```
[bmsolovjev@fedora lab05]$ nasm -f elf lab5-2.asm
[bmsolovjev@fedora lab05]$ ld -m elf_i386 -o lab5-2 lab5-2.o
[bmsolovjev@fedora lab05]$ ./lab5-2
Введите строку:
Соловьев Богдан Михайлович
```

Figure 9: Создание и проверка

Изменяю текст программы, прописывая sprint вместо sprintLF(рис. 10).

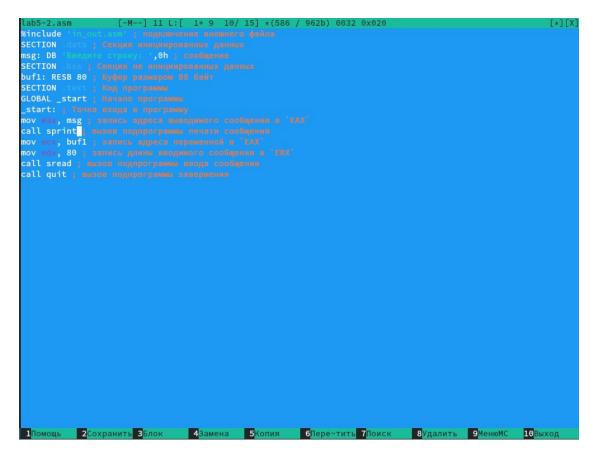


Figure 10: Замена sprintLF на sprint

Повторяю свои действия в третий раз (рис. 11).

```
[bmsolovjev@fedora lab05]$ nasm -f elf lab5-2.asm
[bmsolovjev@fedora lab05]$ ld -m elf_i386 -o lab5-2-2 lab5-2.o
[bmsolovjev@fedora lab05]$ ./lab5-2-2
Введите строку: Соловьев Богдан Михайович
```

Figure 11: Программа больше не переносит строку

#### 3.2 Самостоятельная работа

Создаю файл lab5-1-1.asm и вставляю в него новый код (рис. 12).

```
GNU nano 7.2
                                      /home/bmsolovjev/work/arch-pc/lab05/lab5-1-1.asm
                                                                                                                   Изменён
         .bss ; Секция не инициированных данных
        ESB 80 ; Буфер размером 80 байт
                     -- Текст программы -----
        .text ; Код программы
        _start ; Начало программы
      :: ; 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` ----
; После вызова инструкции 'int 80h' программа будет ожидать ввода
; строки, которая будет записана в переменную 'buf1' размером 80 байт
mov eax, 3 ; Системный вызов для чтения (sys_read)
mov ebx, 0 ; Дескриптор файла 0 - стандартный ввод
mov ecx, bufl ; Адрес буфера под вводимую строку
mov edx, 80 ; Длина вводимой строки
int 80h ; Вызов ядра
          ---- Системный вызов `exit` -----
; После вызова инструкции 'int 80h' программа завершит работу
mov eax,4 ; Системный вызов для выхода (sys_exit)
mov ebx,1 ;
mov ecx, buf1;
mov edx,buf1;
int 80h;
mov eax,1;
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра
                                  ^W Поиск
^\ 3эч
                 ^О Записать
^R ЧитФайл
                                                   ^К Вырезать
^U Вставить
                                                                     ^Т Выполнить
                                                                                      ^С Позиция
                                                                                                       М−U Отмена
   Справка
                    ЧитФайл
                                      Замена
                                                                                         К строке
                                                                                                            Повтор
```

Figure 12: Новый код файла lab5-1-1.asm

Повторяю свои действия в четвёртый раз (рис. 13).

```
[bmsolovjev@fedora lab05]$ nasm -f elf lab5-1-1.asm
[bmsolovjev@fedora lab05]$ ld -m elf_i386 -o lab5-1-1 lab5-1-1.o
[bmsolovjev@fedora lab05]$ ./lab5-1-1
Введите строку:
Соловьев Богдан
Соловьев Богдан
[bmsolovjev@fedora lab05]$
```

Figure 13: Теперь программа пишет введённый текст

Создаю файл lab5-2-1.asm и вставляю в него новый код (рис. 14).

```
GNU nano 7.2
                                     /home/bmsolovjev/work/arch-pc/lab05/lab5-2-1.asm
                                                                                                                   Изменён
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция инициированных данных
         'Введите строку: ',0h ; сообщение
        .bss ; Секция не инициированных данных
         SB 80´; Буфер размером 80 байт
.text ; Код программы
        _start ; Начало программы
        ; 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
mov edx, 80 ; запись длины вводимого сообщения в `EBX`
call sread ; вызов подпрограммы ввода сообщения
mov eax,4;
mov ebx,1;
mov ecx,buf1 ;
int 80h ;
call quit ; вызов подпрограммы завершения
                                                                     ^Т Выполнить
^Ј Выровнять
                                                                                     ^C Позиция
^/ К строке
                 ^0 Записать
                                  ^W Поиск
^\ Замена
 G Справка
                                                    ^К Вырезать
                                                                                                       M-U Отмена
                                                                                                       М-Е Повтор
   Выход
                    ЧитФайл
                                     Замена
                                                       Вставить
                                                                                         К строке
```

Figure 14: Новый код файла lab5-2-1.asm

Повторяю свои действия в пятый раз (рис. 15).

```
[bmsolovjev@fedora lab05]$ nasm -f elf lab5-2-1.asm
[bmsolovjev@fedora lab05]$ ld -m elf_i386 -o lab5-2-1 lab5-2-1.o
[bmsolovjev@fedora lab05]$ ./lab5-2-1
Введите строку: Соловьев Богдан Михайлович
Соловьев Богдан Михайлович
[bmsolovjev@fedora lab05]$ ■
```

Figure 15: Теперь программа пишет введённый текст и не переносит строку

## 4 Выводы

При выполнении лабораторной работы, я приобрёл навыки работы в Midnight Commander, а также узнал принцип работы mov, int, print, printLF на языке программирования NASM.

## Список литературы