

Отчёта по лабораторной работе №5

Архитектура компьютера

Еремина Оксана Андреевна НКАбд-02-23

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	15
	Список литературы	16

Список таблиц

Список иллюстраций

4.1	Создание папок в Midnight Commander	8
4.2	Создание файла в Midnight Commander	9
4.3	Написание кода программы	9
4.4	Компиляция файла и передача на обработку компоновщику и исполнение файла	10
4.5	Исполнение файла	10
4.6	Скачивание и перемещение файла	11
4.7	Скачивание и перемещение файла	11
4.8	Содержимое файла lab5-2.asm	12
4.9	Исполнение файла lab5-2.asm	12
4.10	Содержимое копии файла lab5-1.asm	13
4.11	Исполнение файла	13
4.12	Содержимое копии файла lab5-2.asm	14
4.13	Исполнение файла	14

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 тс
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.

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

1. Основы работы с тс

Открываю терминал, ввожу команду тс и создаю папку lab05 (рис1)

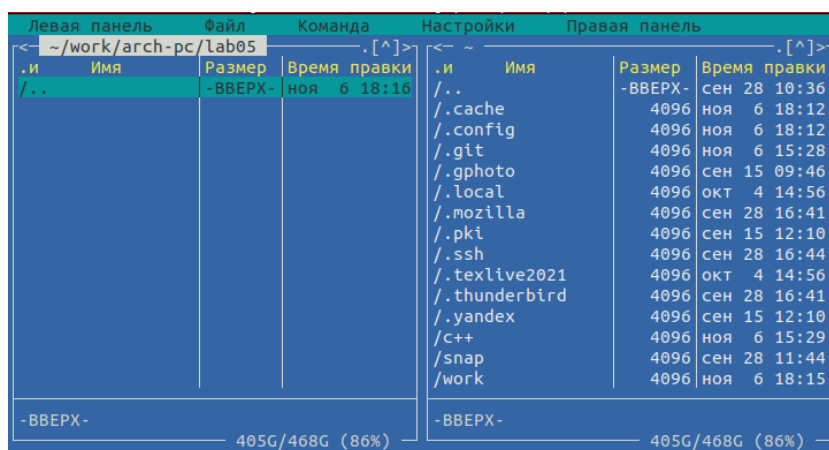


Рис. 4.1: Создание папок в Midnight Commander

С помощью команды `touch` создаю файл `lab5-1.asm` (рис2)

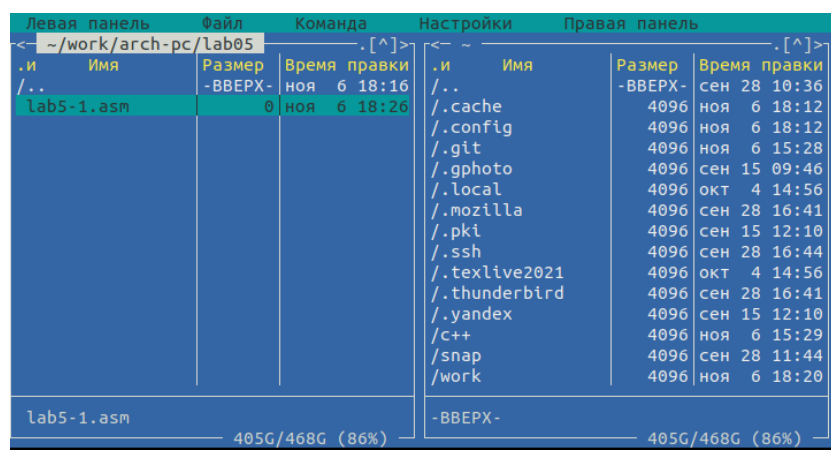


Рис. 4.2: Создание файла в Midnight Commander

2. Структура программы на языке ассемблера NASM

С помощью клавиши F4 открываю созданный файл. Ввожу в файл код программы для запроса строки у пользователя. (рис3)

```

1 SECTION .data
2 msg: DB 'Введите строку:',10
3 msgLen: EQU $-msg
4 SECTION .bss
5 buf1: RESB 80
6 SECTION .text
7 GLOBAL _start
8 _start:
9 mov eax,4
10 mov ebx,1
11 mov ecx,msg
12 mov edx,msgLen
13 int 80h
14 mov eax, 3
15 mov ebx, 0
16 mov ecx, buf1
17 mov edx, 80
18 int 80h
19 mov eax,1
20 mov ebx,0
21 int 80h

```

Рис. 4.3: Написание кода программы

Транслирую текст программы файла в объектный файл командой `nasm -f elf`

lab5-1.asm. Выполняю компоновку объектного файла с помощью команды `ld -m elf_i386 -o lab5-1 lab5-1.o`. Далее запускаю файл командой `$./lab5-1` (рис4)

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

Рис. 4.4: Компиляция файла и передача на обработку компоновщику и исполнение файла

Программа выводит строку “Введите строку:”, я ввожу свои ФИО, на этом программа заканчивает свою работу (рис5)

```
oaeremina@oaeremina:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Еремина Оксана Андреевна
```

Рис. 4.5: Исполнение файла

3. Подключение внешнего файла

Скачиваю файл `in_out.asm` со страницы курса в ТУИС, он сохранился в каталоге Загрузки. С помощью клавиши F5 копирую файл из каталога загрузки в каталог lab05 (рис6)

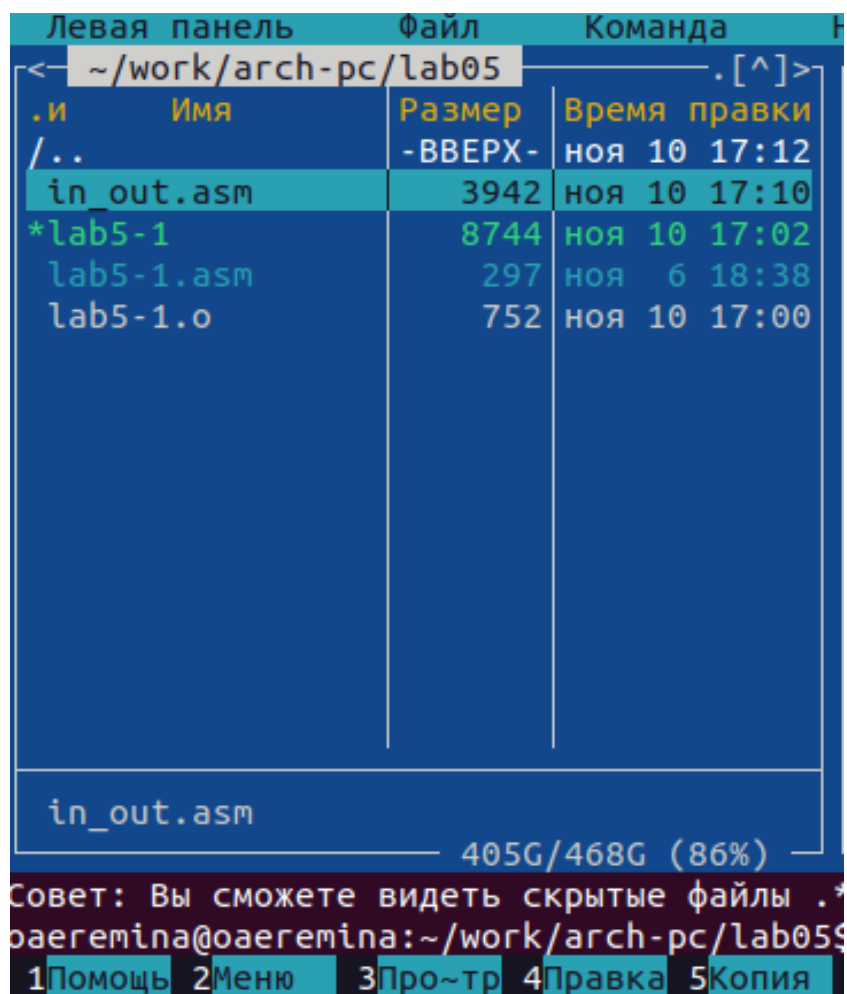


Рис. 4.6: Скачивание и перемещение файла

С помощью клавиши F5 копирую файл lab5-1 в тотже каталог, с названием lab5-2 и проверяю это (рис7)

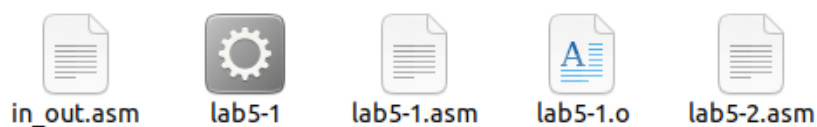


Рис. 4.7: Скачивание и перемещение файла

Изменяю содержимое файла lab5-2.asm в текстовом редакторе, чтобы в про-

грамме использовались подпрограммы из внешнего файла in_out.asm. (рис8)

```
1 %include 'in_out.asm' ; подключение внешнего файла
2 SECTION .data ; Секция инициированных данных
3 msg: DB 'Введите строку: ',0h ; сообщение
4 SECTION .bss ; Секция не инициированных данных
5 buf1: RESB 80 ; Буфер размером 80 байт
6 SECTION .text ; Код программы
7 GLOBAL _start ; Начало программы
8 _start: ; Точка входа в программу
9 mov eax, msg ; запись адреса выводимого сообщения в `EAX`
10 call sprint ; вызов подпрограммы печати сообщения
11 mov ecx, buf1 ; запись адреса переменной в `EAX`
12 mov edx, 80 ; запись длины вводимого сообщения в `EBX`
13 call sread ; вызов подпрограммы ввода сообщения
14 call quit ; вызов подпрограммы завершения
```

Рис. 4.8: Содержимое файла lab5-2.asm

Транслирую текст программы файла в объектный файл командой `nasm -f elf lab5-2.asm`. Выполняю компоновку объектного файла с помощью команды `ld -m elf_i386 -o lab5-2 lab5-2.o`. Далее запускаю файл командой `$./lab5-2` (рис9)

```
oeremina@oeremina:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
oeremina@oeremina:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
oeremina@oeremina:~/work/arch-pc/lab05$ ./lab5-2
Введите строку: Еремина Оксана Андреевна
```

Рис. 4.9: Исполнение файла lab5-2.asm

Разница между первым исполняемым файлом и вторым в том, что запуск первого запрашивает ввод с новой строки, в то время как при запуске второго, запрашивается ввод без переноса на новую строку.

4. Выполнение заданий для самостоятельной работы

Создаю копию файла lab5-1.asm, открываю созданный файл и изменяю программу (рис10)

```

1 SECTION .data ; Секция иницированных данных
2 msg: DB 'Введите строку:',10
3 msgLen: EQU $-msg ; Длина переменной 'msg'
4 SECTION .bss ; Секция не иницированных данных
5 buf1: RESB 80 ; Буфер размером 80 байт
6 SECTION .text ; Код программы
7 GLOBAL _start ; Начало программы
8 _start: ; Точка входа в программу
9 mov eax,4 ; Системный вызов для записи (sys_write)
10 mov ebx,1 ; Описатель файла 1 - стандартный вывод
11 mov ecx,msg ; Адрес строки 'msg' в 'ecx'
12 mov edx,msgLen ; Размер строки 'msg' в 'edx'
13 int 80h ; Вызов ядра
14 mov eax, 3 ; Системный вызов для чтения (sys_read)
15 mov ebx, 0 ; Дескриптор файла 0 - стандартный ввод
16 mov ecx, buf1 ; Адрес буфера под вводимую строку
17 mov edx, 80 ; Длина вводимой строки
18 int 80h ; Вызов ядра
19 mov eax,4 ; Системный вызов для записи (sys_write)
20 mov ebx,1 ; Описатель файла '1' - стандартный вывод
21 mov ecx,buf1 ; Адрес строки buf1 в ecx
22 mov edx,80 ; Размер строки buf1
23 int 80h ; Вызов ядра
24 mov eax,1 ; Системный вызов для выхода (sys_exit)
25 mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
26 int 80h ; Вызов ядра

```

Рис. 4.10: Содержимое копии файла lab5-1.asm

Создаю объектный файл lab5-1-1.0, получаю исполняемый файл lab5-1-1, отдав его на обработку компоновщику, запускаю файл. Программа запрашивает ввод, ввожу ФИО, далее программа выводит введенные данные (рис11)

```

oaeremina@oaeremina:~/work/arch-pc/lab05$ nasm -f elf lab5-1-1.asm
oaeremina@oaeremina:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1-1 lab5-1-1.o
oaeremina@oaeremina:~/work/arch-pc/lab05$ ./lab5-1-1
Введите строку:
Еремина Оксана Андреевна
Еремина Оксана Андреевна

```

Рис. 4.11: Исполнение файла

Создаю копию файла lab5-2.asm, открываю созданный файл и изменяю программу (рис12)

```

1 %include 'in_out.asm'
2 SECTION .data ; Секция инициированных данных
3 msg: DB 'Введите строку: ',0h ; сообщение
4 SECTION .bss ; Секция не инициированных данных
5 buf1: RESB 80 ; Буфер размером 80 байт
6 SECTION .text ; Код программы
7 GLOBAL _start ; Начало программы
8 _start: ; Точка входа в программу
9 mov eax, msg ; запись адреса выводимого сообщения в `EAX`
10 call sprint ; вызов подпрограммы печати сообщения
11 mov ecx, buf1 ; запись адреса переменной в `EAX`
12 mov edx, 80 ; запись длины вводимого сообщения в `EBX`
13 call sread ; вызов подпрограммы ввода сообщения
14 mov eax,4 ; Системный вызов для записи (sys_write)
15 mov ebx,1 ; Описатель файла '1' - стандартный вывод
16 mov ecx,buf1 ; Адрес строки buf1 в ecx
17 int 80h ; Вызов ядра
18 call quit ; вызов подпрограммы завершения

```

Рис. 4.12: Содержимое копии файла lab5-2.asm

Создаю объектный файл lab5-2-1.0, получаю исполняемый файл lab5-2-1, отдав его на обработку компоновщику, запускаю файл. Программа запрашивает ввод, ввожу ФИО, далее программа выводит введенные данные (рис13)

```

oaeremina@oaeremina:~/work/arch-pc/lab05$ nasm -f elf lab5-2-1.asm
oaeremina@oaeremina:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2-1 lab5-2-1.o
oaeremina@oaeremina:~/work/arch-pc/lab05$ ./lab5-2-1
Введите строку: Еремина Оксана Андреевна
Еремина Оксана Андреевна

```

Рис. 4.13: Исполнение файла

5 Выводы

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

Список литературы

Архитектура ЭВМ