

Отчёт по лабораторной работе №5

Дисциплина: Архитектура компьютера

Айлина Сафиуллина

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
4	Выводы	16

Список иллюстраций

3.1	Midnight Commander	6
3.2	Midnight Commander	7
3.3	создание файла lab5-1.asm	7
3.4	команда touch	7
3.5	листинг 5.1	8
3.6	оттранслирование, компоновка и запуск файла	8
3.7	фрагмент файла in_out.asm	9
3.8	создание копии файла	10
3.9	листинг 5.2	10
3.10	листинг 5.2	11
3.11	замена sprintLF на sprint	11
3.12	результат замены sprintLF на sprint	12
3.13	создание копии файла lab5-1.asm	12
3.14	результат создания копии файла	13
3.15	проверка работы исполняемого файла	13
3.16	создание копии файла lab5-2.asm	14
3.17	результат создания копии файла	14
3.18	проверка работы исполняемого файла	15

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. Структура программы на языке ассемблера NASM
3. Подключение внешнего файла
4. Выполнение заданий для самостоятельной работы

3 Выполнение лабораторной работы

1. Открываем Midnight Commander

assafullina@dk3n33:~\$ mc (рис. 3.1).

с помощью клавиш стрелка вверх, стрелка вниз и Enter переходим в каталог ~/work/arch-pc

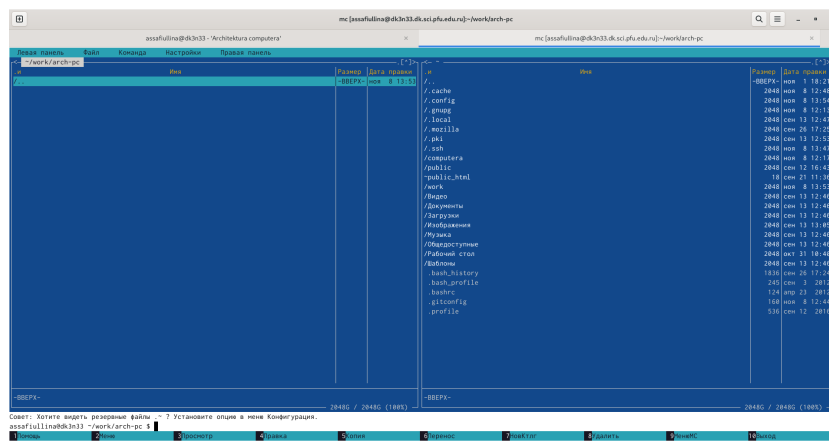


Рис. 3.1: Midnight Commander

с помощью функциональной клавиши F7 создадим папку lab05 и перейдем в созданный каталог (рис. 3.2)

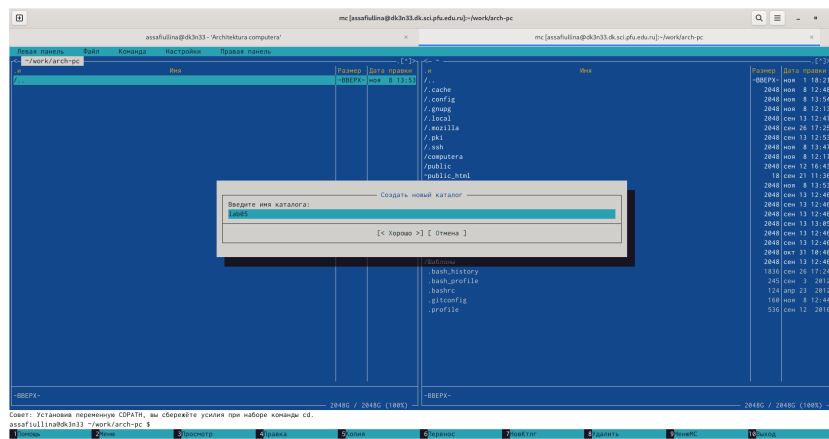


Рис. 3.2: Midnight Commander

пользуясь строкой ввода и командой touch(рис. 3.4) создаем файл lab5-1.asm (рис. 3.3)

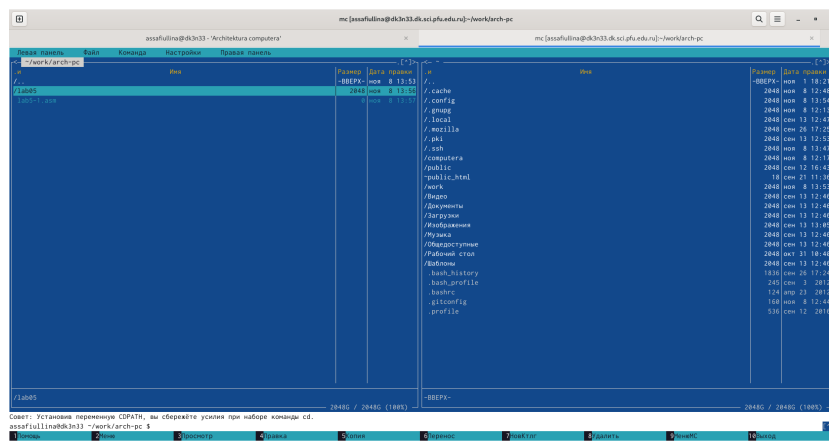


Рис. 3.3: создание файла lab5-1.asm

```
assafiullina@dk3n33 ~ $ mc ~/work/arch-pc
assafiullina@dk3n33 ~/work/arch-pc $ touch lab5-1.asm
```

Рис. 3.4: команда touch

с помощью функциональной клавиши F4 открываем файл lab5-1.asm для редактирования во встроенном редакторе Midnight Commander и вводим текст программы из листинга 5.1 (рис. 3.5)

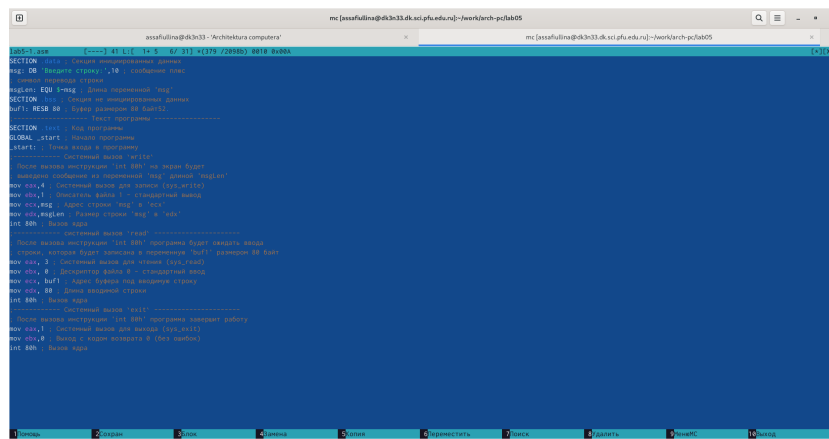


Рис. 3.5: листинг 5.1

с помощью функциональной клавиши F3 открываем файл lab5-1.asm для просмотра и проверяем, что файл содержит текст программы.

2. Структура программы на языке ассемблера NASM

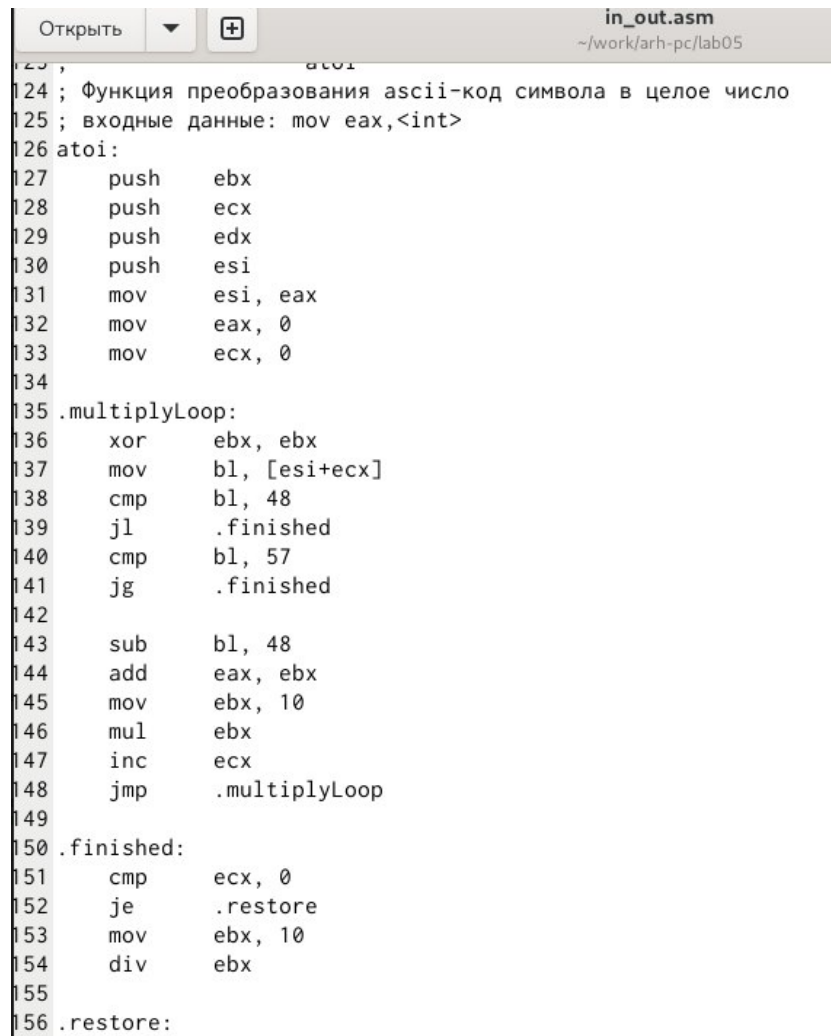
оттранслируем текст программы lab5-1.asm в объектный файл. Выполняем компоновку объектного файла и запускаем получившийся в исполняемый файл. Программа выводит строку “Введите строку:” и ожидает ввода с клавиатуры. (рис. 3.6)

```
assafiullina@dk3n33 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.asm
assafiullina@dk3n33 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.o
assafiullina@dk3n33 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку:
Сафиуллина Айлина Саяровна
assafiullina@dk3n33 ~/work/arch-pc/lab05 $
assafiullina@dk3n33 ~/work/arch-pc/lab05 $
```

Рис. 3.6: оттранслирование, компоновка и запуск файла

3. Подключение внешнего файла in_out.asm

скачиваем файл in_out.asm со страницы курса в ТУИС (рис. 3.7)



```

Открыть  + in_out.asm
~/work/arh-pc/lab05
123 ;
124 ; Функция преобразования ascii-код символа в целое число
125 ; входные данные: mov eax,<int>
126 atoi:
127     push    ebx
128     push    ecx
129     push    edx
130     push    esi
131     mov     esi, eax
132     mov     eax, 0
133     mov     ecx, 0
134
135 .multiplyLoop:
136     xor     ebx, ebx
137     mov     bl, [esi+ecx]
138     cmp     bl, 48
139     jl      .finished
140     cmp     bl, 57
141     jg      .finished
142
143     sub     bl, 48
144     add     eax, ebx
145     mov     ebx, 10
146     mul     ebx
147     inc     ecx
148     jmp     .multiplyLoop
149
150 .finished:
151     cmp     ecx, 0
152     je      .restore
153     mov     ebx, 10
154     div     ebx
155
156 .restore:

```

Рис. 3.7: фрагмент файла in_out.asm

в одной из панелей тс открываем каталог с файлом lab5-1.asm. в другой панели каталог со скаченным файлом in_out.asm, копируем файл in_out.asm в каталог с файлом lab5-1.asm с помощью функциональной клавиши F5
далее создаем копию файла lab5-1.asm с именем lab5-2.asm (рис. 3.8)

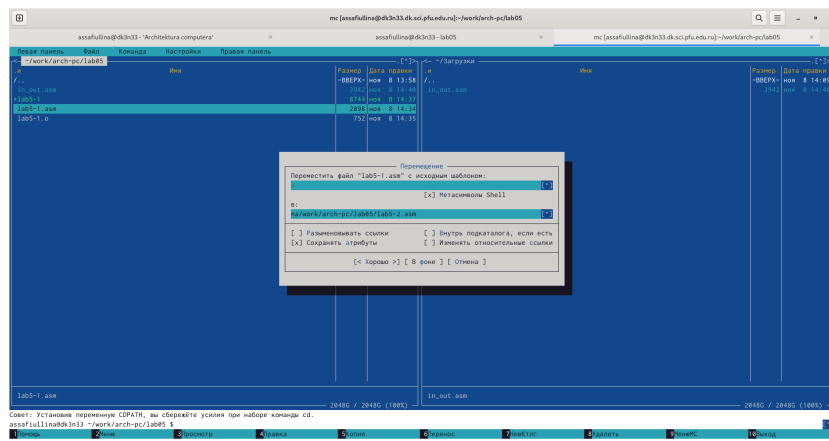


Рис. 3.8: создание копии файла

исправляем текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in_out.asm в соответствии с листингом 5.2. (рис. 3.9)

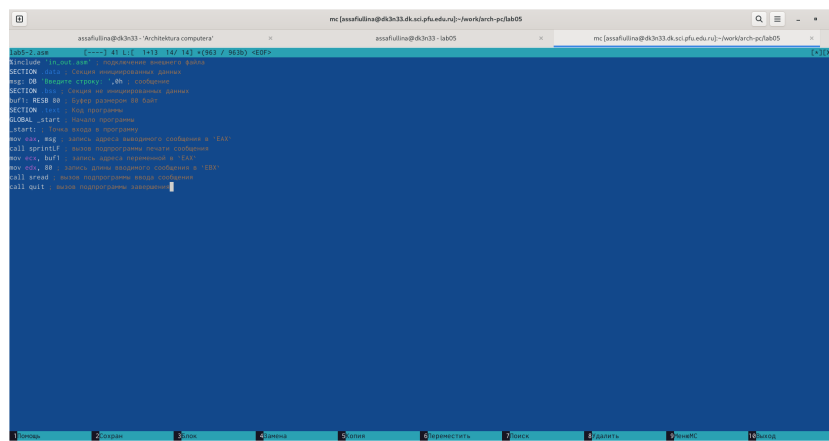


Рис. 3.9: листинг 5.2

создаем исполняемый файл и проверяем его работу. (рис. 3.10)


```

assafullina@dk3n33: ~$ cd /work/arch-pc/lab05
assafullina@dk3n33: /work/arch-pc/lab05$ ls
lab5-1.asm
lab5-2.asm
lab5-1.o
lab5-2.o
assafullina@dk3n33: /work/arch-pc/lab05$ gcc -f elf lab5-2.asm -o lab5-2.o
assafullina@dk3n33: /work/arch-pc/lab05$ ./lab5-2
Введите строку:
\
assafullina@dk3n33: /work/arch-pc/lab05$ gcc -f elf lab5-2.asm -o lab5-2.o
assafullina@dk3n33: /work/arch-pc/lab05$ ./lab5-2
Введите строку:
Сабвушка Алина Сапогова
assafullina@dk3n33: /work/arch-pc/lab05$ gcc -f elf lab5-2.asm -o lab5-2.o
assafullina@dk3n33: /work/arch-pc/lab05$ ./lab5-2
Введите строку:
Сабвушка Алина Сапогова
assafullina@dk3n33: /work/arch-pc/lab05$

```

Рис. 3.12: результат замены sprintf на printf

4. Задание для самостоятельной работы.

создадим копию файла lab5-1.asm. Внесем изменения в программу так, чтобы она работала по следующему алгоритму:

- вывести приглашение типа “Введите строку:”;
- ввести строку с клавиатуры;
- вывести введенную строку на экран.

(рис. 3.13)

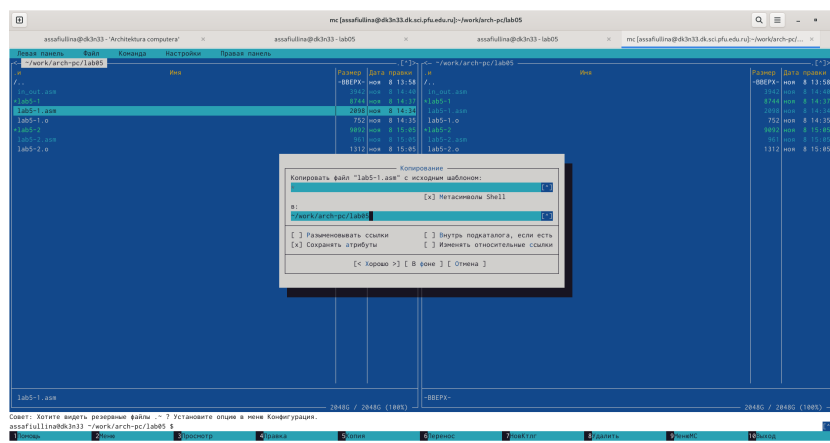


Рис. 3.13: создание копии файла lab5-1.asm

(рис. 3.14)

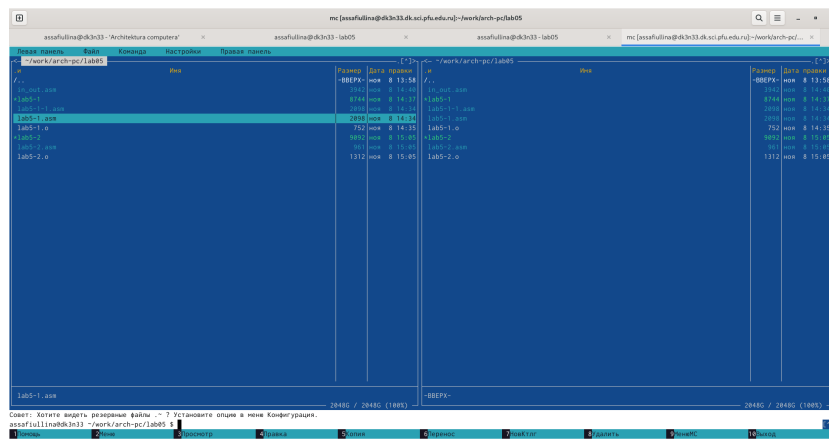


Рис. 3.14: результат создания копии файла

получим исполняемый файл и проверим его работу. на приглашение ввести строку введем свое ФИО
(рис. 3.15)

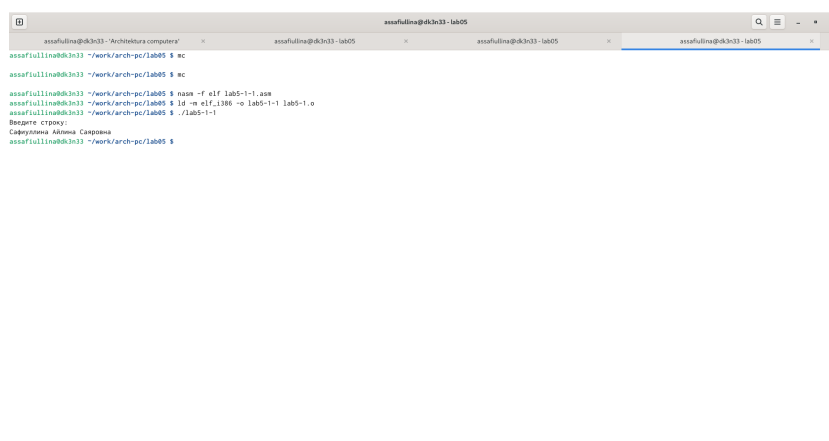


Рис. 3.15: проверка работы исполняемого файла

создадим копию файла lab5-2.asm. Внесем изменения в программу так, чтобы она работала по следующему алгоритму:

- вывести приглашение типа “Введите строку:”;
- ввести строку с клавиатуры;
- вывести введенную строку на экран.

(рис. 3.16)

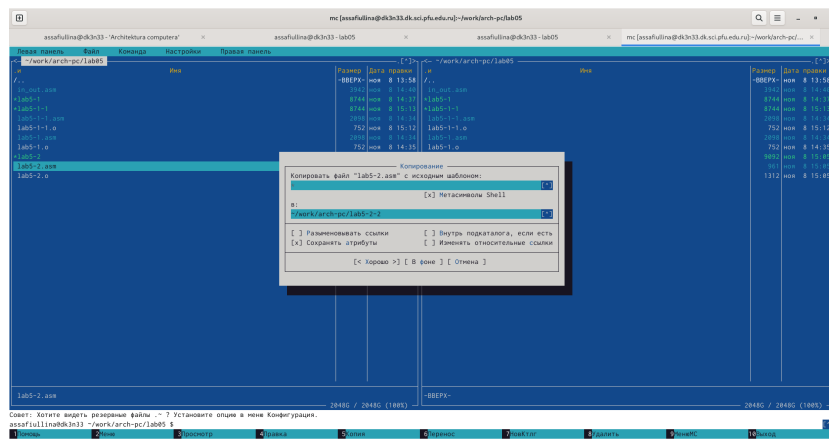


Рис. 3.16: создание копии файла lab5-2.asm

(рис. 3.17)

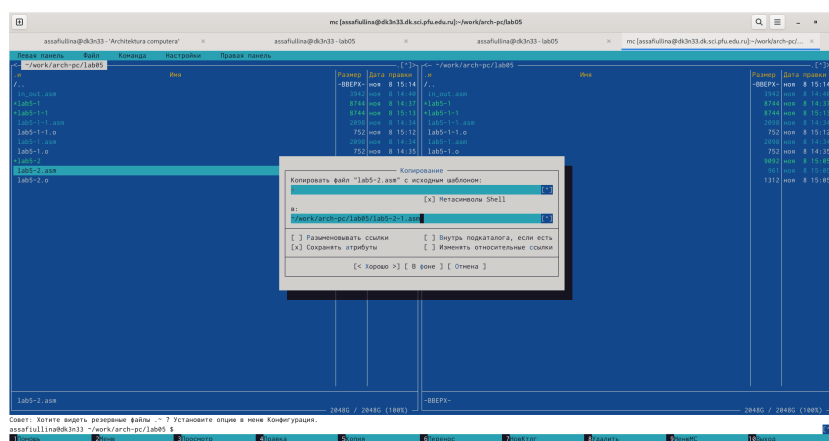


Рис. 3.17: результат создания копии файла

получим исполняемый файл и проверим его работу. на приглашение ввести строку введем свое ФИО

(рис. 3.18)

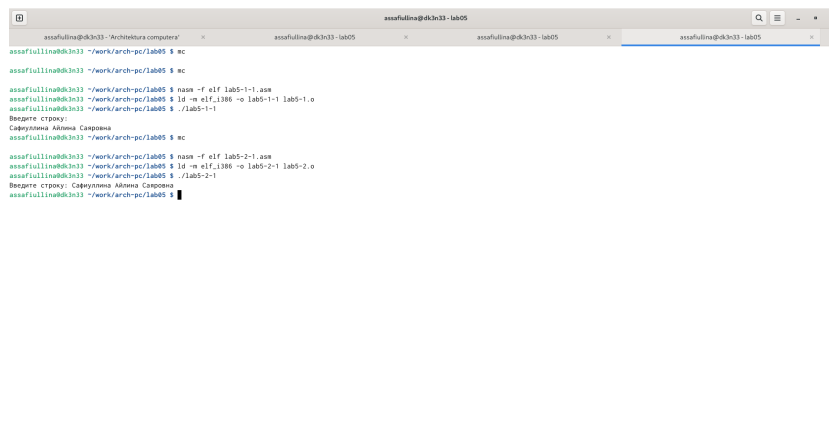


Рис. 3.18: проверка работы исполняемого файла

4 Выводы

При выполнении данной лабораторной работы мы приобрели навыки практической работы в Midnight Commander, освоили инструкции языка ассемблера `mov` и `int`, а также выполнили самостоятельную работу.