

Отчёт по лабораторной работе №8

дисциплина: Архитектура компьютера

Попов Даниил Георгиевич

Содержание

1)Цель работы	5
2)Задание	6
3)Выполнение лабораторной работы	7
3.1)Создаем каталог для программ ЛБ8, и в нем создаем файл	7
3.2)Открываем файл в Midnight Commander и заполняем его согласно листингу 1. Создаем исполняемый файл и проверяем	8
3.3)Изменяем код в файле и проверяем что изменилось	9
3.4)Еще раз меняем код и проверяем что изменилось	9
3.5)Создаем файл и заполняем его согласно листингу 2, проверяем работу файла . . .	10
3.6)Создаем файл и заполняем его согласно листингу 3, проверяем работу файла . . .	10
3.7)Снова открываем файл для редактирования и изменяем его, чтобы вычислялось произведение вводимых значений	11
4)Задания для самостоятельной работы	12
4.1)Создаем файл и открываем его через Midnight Commander	12
4.2)Открываем файл и пишем код	13
4.3)Проверяем	14
5)Выводы	15

Список иллюстраций

1	Создаем	7
2	Заполняем, проверяем	8
3	создаем, проверяем	10
1	проверяем	12
2	код	13
3	проверка	14

Список таблиц

1)Цель работы

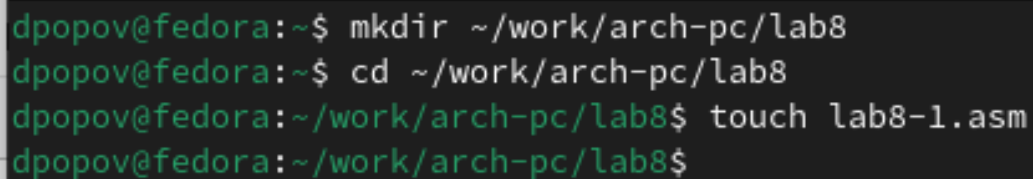
Приобретение навыков написания программ с использованием циклов и обработкой аргументов командной строки.

2)Задание

Написать программу с использование циклов и обработки аргументов командной строки.

3)Выполнение лабораторной работы

3.1)Создаем каталог для программ ЛБ8, и в нем создаем файл

A terminal window with a dark background and green text. It shows four lines of commands and their execution. The first line creates a directory named 'lab8' in the path '~/work/arch-pc/'. The second line changes the current directory to 'lab8'. The third line creates a file named 'lab8-1.asm' using the 'touch' command. The fourth line shows the prompt after the file has been created.

```
dpopov@fedora:~$ mkdir ~/work/arch-pc/lab8
dpopov@fedora:~$ cd ~/work/arch-pc/lab8
dpopov@fedora:~/work/arch-pc/lab8$ touch lab8-1.asm
dpopov@fedora:~/work/arch-pc/lab8$
```

Рис. 1: Создаем

3.2)Открываем файл в Midnight Commander и заполняем его согласно листингу 1. Создаем исполняемый файл и проверяем

```
cdpopov@fedora:~/work/arch-pc/lab8$ mc
cdpopov@fedora:~/work/arch-pc/lab8$ nasm -f elf lab8-1.asm
cdpopov@fedora:~/work/arch-pc/lab8$ ld -m elf_i386 -o lab8-1 lab8-1.o
cdpopov@fedora:~/work/arch-pc/lab8$ ./lab8-1
Введите N: 10
10
9
8
7
6
5
4
3
2
1
cdpopov@fedora:~/work/arch-pc/lab8$
```

Рис. 2: Заполняем, проверяем

3.3)Изменяем код в файле и проверяем что изменилось

```
drogov@fedora:~/work/arch-pc/lab8$ mc

drogov@fedora:~/work/arch-pc/lab8$ nasm -f elf lab8-1.asm
drogov@fedora:~/work/arch-pc/lab8$ cd -m elf_i386 -o lab8-1 lab8-1.o
bash: cd: -m: недопустимый параметр
cd: использование: cd [-L|[-P [-e]] [-@]] [каталог]
drogov@fedora:~/work/arch-pc/lab8$ ld -m elf_i386 -o lab8-1 lab8-1.o
drogov@fedora:~/work/arch-pc/lab8$ ./lab8-1
Введите N: 10
9
7
5
3
1
Ошибка сегментирования (образ памяти сброшен на диск)
drogov@fedora:~/work/arch-pc/lab8$
```

Регистр esx принимает значения 9,7,5,3,1(на вход подается число 10, в цикле label данный регистр уменьшается на 2 командой sub и loop). Число проходов цикла не соответствует числу N, так как уменьшается на 2.

3.4)Еще раз меняем код и проверяем что изменилось

```
drogov@fedora:~/work/arch-pc/lab8$ mc

drogov@fedora:~/work/arch-pc/lab8$ nasm -f elf lab8-1.asm
drogov@fedora:~/work/arch-pc/lab8$ ld -m elf_i386 -o lab8-1 lab8-1.o
drogov@fedora:~/work/arch-pc/lab8$ ./lab8-1
Введите N: 10
9
8
7
6
5
4
3
2
1
0
Ошибка сегментирования (образ памяти сброшен на диск)
```

В данном случае число проходов цикла равна числу N.

3.5) Создаем файл и заполняем его согласно листингу 2, проверяем работу файла

```
Ошибка сегментирования (образ памяти сброшен на диск)
dporov@fedora:~/work/arch-pc/lab8$ touch lab8-2.asm
dporov@fedora:~/work/arch-pc/lab8$ mc

dporov@fedora:~/work/arch-pc/lab8$ nasm -f elf lab8-2.asm
dporov@fedora:~/work/arch-pc/lab8$ ld -m elf_i386 -o lab8-2 lab8-2.o
dporov@fedora:~/work/arch-pc/lab8$ ./lab8-2 1 '2' 3
1
2
3
dporov@fedora:~/work/arch-pc/lab8$
```

Программой было обработано 3 аргумента.

3.6) Создаем файл и заполняем его согласно листингу 3, проверяем работу файла

```
dporov@fedora:~/work/arch-pc/lab8$ touch lab8-3.asm
dporov@fedora:~/work/arch-pc/lab8$ mc

dporov@fedora:~/work/arch-pc/lab8$ nasm -f elf lab8-3.asm
dporov@fedora:~/work/arch-pc/lab8$ ld -m elf_i386 -o lab8-3 lab8-3.o
dporov@fedora:~/work/arch-pc/lab8$ ./lab8-3
Результат: 0
dporov@fedora:~/work/arch-pc/lab8$ ./lab8-3
Результат: 0
dporov@fedora:~/work/arch-pc/lab8$ ./lab8-3 1 56 9
Результат: 66
dporov@fedora:~/work/arch-pc/lab8$
```

Рис. 3: создаем, проверяем

3.7) Снова открываем файл для редактирования и изменяем его, чтобы вычислялось произведение вводимых значений

```
next:
cmp ecx,0
jz _end
pop eax
call atoi
imul esi,eax
loop next
_end:
```

```
dpopov@fedora:~/work/arch-pc/lab8$ mc
```

```
dpopov@fedora:~/work/arch-pc/lab8$ nasm -f elf lab8-3
```

```
dpopov@fedora:~/work/arch-pc/lab8$ ld -m elf_i386 -o
```

```
dpopov@fedora:~/work/arch-pc/lab8$ ./lab8-3 5 4 3
```

```
Результат: 60
```

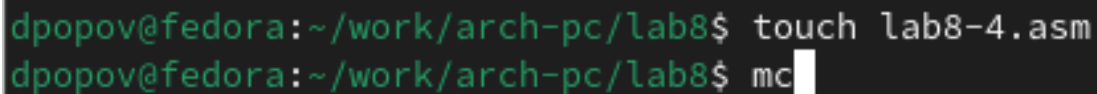
```
dpopov@fedora:~/work/arch-pc/lab8$
```

4)Задания для самостоятельной работы

ВАРИАНТ 10 ##Задача: Напишите программу, которая находит сумму значений функции $f(x)$ для $x = x_1, x_2, \dots, x_n$, т.е. программа должна выводить значение $f(x_1) + f(x_2) + \dots + f(x_n)$. Значения x_i передаются как аргументы. Вид функции $f(x)$ выбрать из таблицы 8.1 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу на нескольких наборах $x = x_1, x_2, \dots, x_n$.

4.1)Создаем файл и открываем его через Midnight

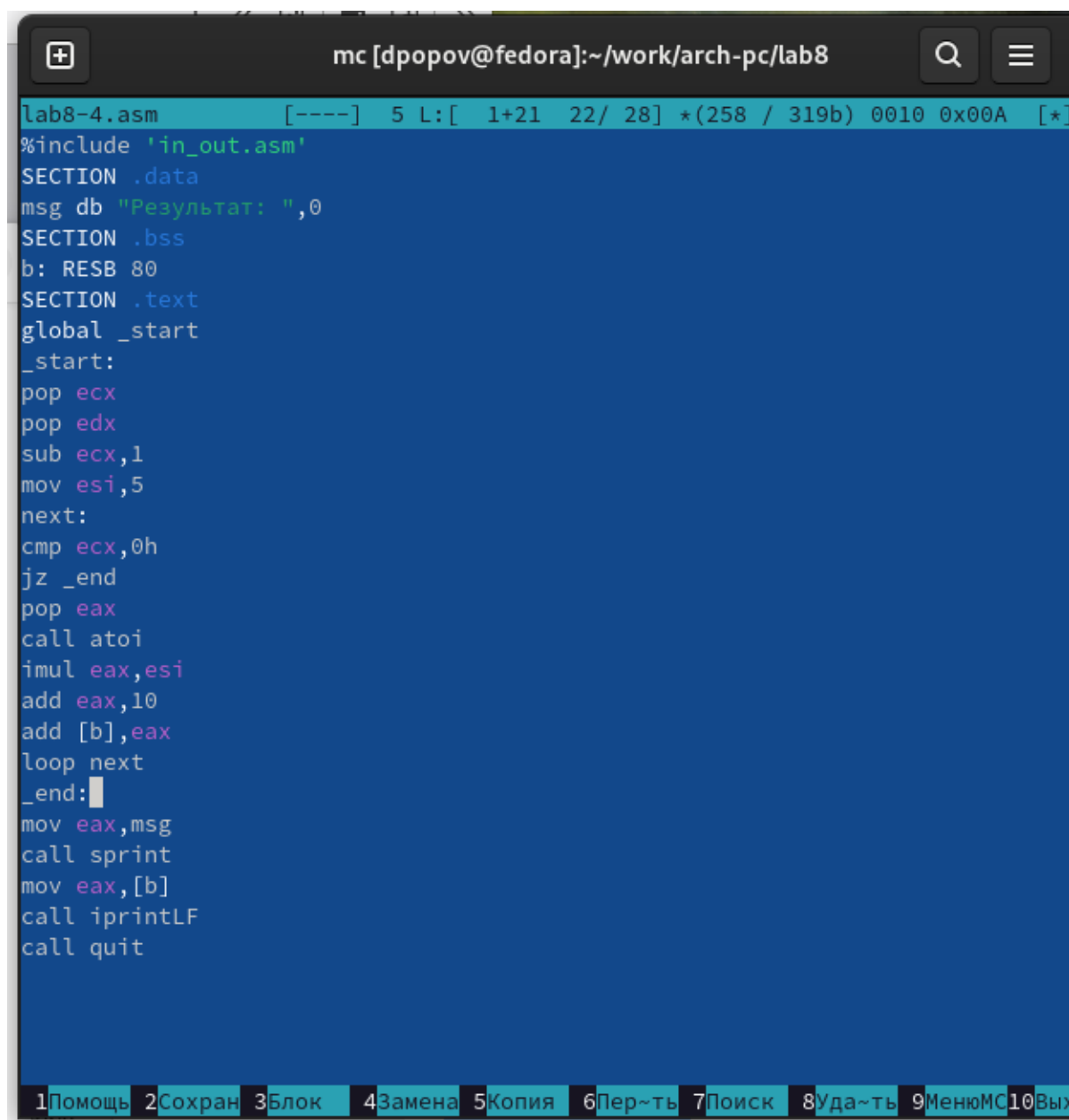
Commander

A terminal window with a dark background and green text. The prompt is 'dpopov@fedora:~/work/arch-pc/lab8\$'. The first command is 'touch lab8-4.asm'. The second command is 'mc' followed by a cursor.

```
dpopov@fedora:~/work/arch-pc/lab8$ touch lab8-4.asm
dpopov@fedora:~/work/arch-pc/lab8$ mc
```

Рис. 1: проверяем

4.2)Открываем файл и пишем код



```
lab8-4.asm [-----] 5 L:[ 1+21 22/ 28] *(258 / 319b) 0010 0x00A [*]
#include 'in_out.asm'
SECTION .data
msg db "Результат: ",0
SECTION .bss
b: RESB 80
SECTION .text
global _start
_start:
pop ecx
pop edx
sub ecx,1
mov esi,5
next:
cmp ecx,0h
jz _end
pop eax
call atoi
imul eax,esi
add eax,10
add [b],eax
loop next
_end:
mov eax,msg
call sprint
mov eax,[b]
call iprintLF
call quit
```

1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер~ть 7Поиск 8Уда~ть 9МенюМС10Вых

Рис. 2: код

4.3)Проверяем

```
drogov@fedora:~/work/arch-pc/lab8$ mc
drogov@fedora:~/work/arch-pc/lab8$ nasm -f elf lab8-4.asm
drogov@fedora:~/work/arch-pc/lab8$ ld -m elf_i386 -o lab8-4 lab8-4.o
drogov@fedora:~/work/arch-pc/lab8$ ./lab8-4 2 3 4
Результат: 75
drogov@fedora:~/work/arch-pc/lab8$
```

Рис. 3: проверка

5)Выводы

Мы научились решать программы с использованием циклов и обработкой аргументов командной строки.