Отчёт по лабораторной работе №8

Дисциплина: Архитектура компьютера

Студент: Верниковская Екатерина Андреевна

Содержание

# Цель работы

Приобрести навыки написания программ с использованием циклов и обработкой аргументов командной строки.

# Задание

1. Создать каталог для программ лабораторной работы №8 и в нём создать файл «lab8-1.asm».
2. Ввести в файл «lab8-1.asm» определённый текст программы с использованием инструкции loop. Создать исполняемый файл и запустить его.
3. Изменить текст программы. Снова создать исполняемый файл и запустить его.
4. Опять изменить текст программы, добавив команды push и pop, создать исполняемый файл и запустить его.
5. Создать файл «lab8-2.asm» и ввести в него определённый текст программы, которая выводит на экран аргументы командной строки. Создать исполняемый файл и запустить его.
6. Создать файл «lab8-3.asm» и ввести в него определённый текст программы, которая вычисляет сумму аргументов командной строки. Создать исполняемый файл и запустить его.
7. Изменить программу для вычисления произведения аргументов командной строки. Снова создать исполняемый файл и запустить его.
8. Написать программу, которая находит сумму значений функции f(x) для x = x1, x2, …, xn. Программа должна выводить значение f(x1) + f(x2) + … + f(xn). Вид функции брать из определённой таблицы, в соответствии с полученным вариантом (В нашем случае это 17 вариант).
9. Создать исполняемый файл и проверить его работу на нескольких наборах x = x1, x2, … xn.

# Выполнение лабораторной работы

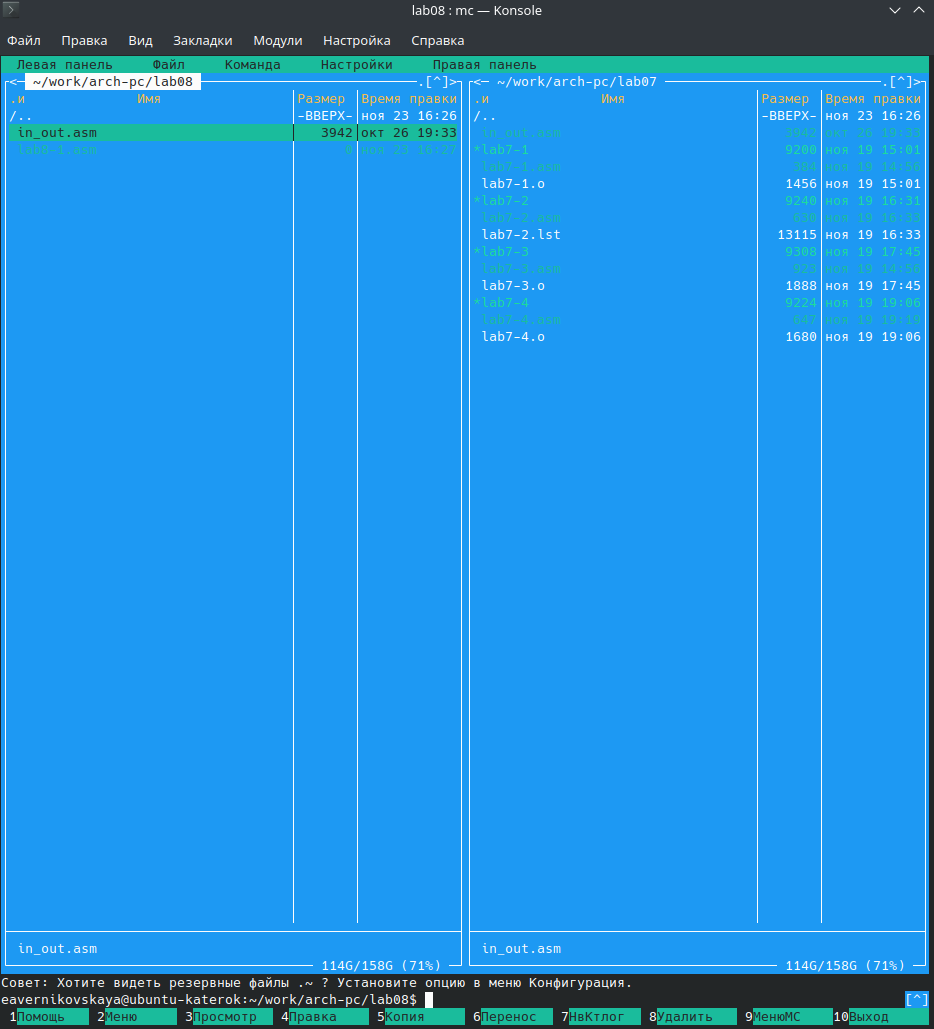
## Реализация циклов в NASM

В созданном каталоге «~/work/arch-pc/lab08» создаём файл «lab8-1.asm» (рис. [-@fig:001])



Создание первого файла

Копируем из каталога «~/work/arch-pc/lab07» файл «in\_out.asm» (рис. [-@fig:002])

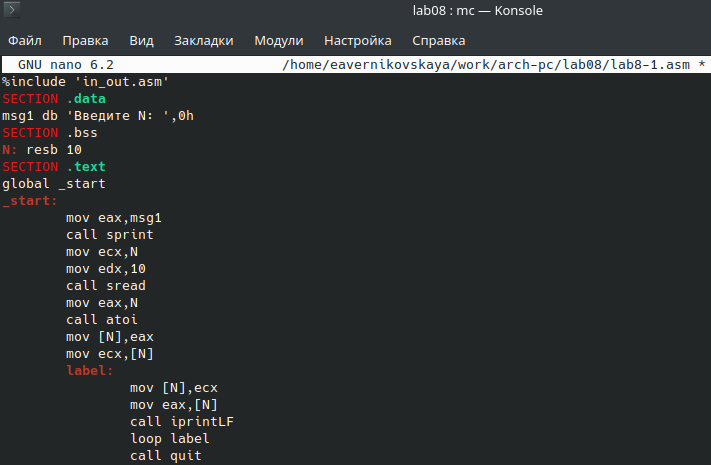


Копирование файла «in\_out.asm»

Вводим нужный текст программы с использованием инструкции loop (рис. [-@fig:003])

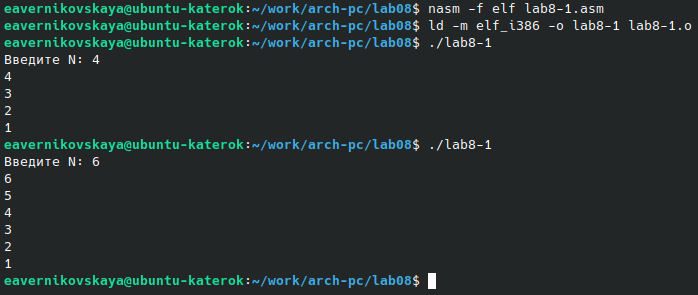
Текст программы:

%include 'in\_out.asm'  
SECTION .data  
msg1 db 'Введите N: ',0h  
SECTION .bss  
N: resb 10  
SECTION .text  
global \_start  
\_start:  
 mov eax,msg1  
 call sprint  
 mov ecx,N  
 mov edx,10  
 call sread  
 mov eax,N  
 call atoi  
 mov [N],eax  
 mov ecx,[N]  
 label:  
 mov [N],ecx  
 mov eax,[N]  
 call iprintLF  
 loop label  
 call quit



Ввод текста программы с инструкцией loop

Создаём исполняемый файл и запускаем его (рис. [-@fig:004])

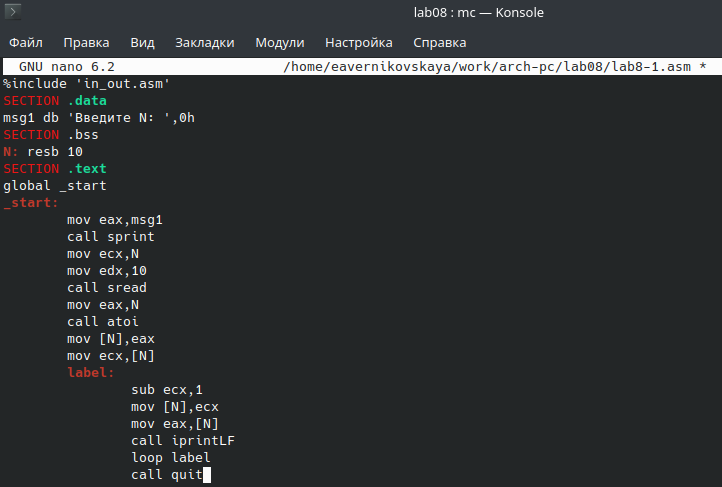


Создание исполняемого файла и его запуск

Изменяем текст программы, добавив изменение значение регистра ecx в цикле label (рис. [-@fig:005])

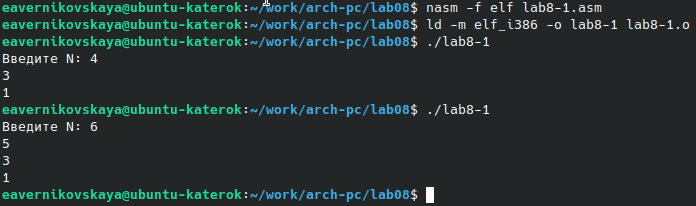
Изменённый текст программы:

%include 'in\_out.asm'  
SECTION .data  
msg1 db 'Введите N: ',0h  
SECTION .bss  
N: resb 10  
SECTION .text  
global \_start  
\_start:  
 mov eax,msg1  
 call sprint  
 mov ecx,N  
 mov edx,10  
 call sread  
 mov eax,N  
 call atoi  
 mov [N],eax  
 mov ecx,[N]  
 label:  
 sub ecx,1  
 mov [N],ecx  
 mov eax,[N]  
 call iprintLF  
 loop label  
 call quit



Изменение программы

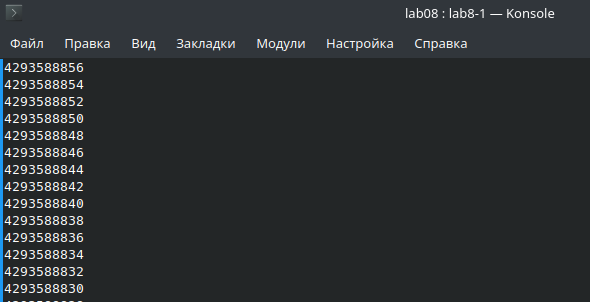
Снова создаём исполняемый файл и запускаем его (рис. [-@fig:006]), (рис. [-@fig:007]) и (рис. [-@fig:008])



Исполняемый файл + запуск 1

Исполняемый файл + запуск 2

Исполняемый файл + запуск 2



Исполняемый файл + запуск 3

Ответы на вопросы:

1. Какие значения принимает регистр ecx в цикле?

* Регистр ecx принимает некорректные значения в цикле.

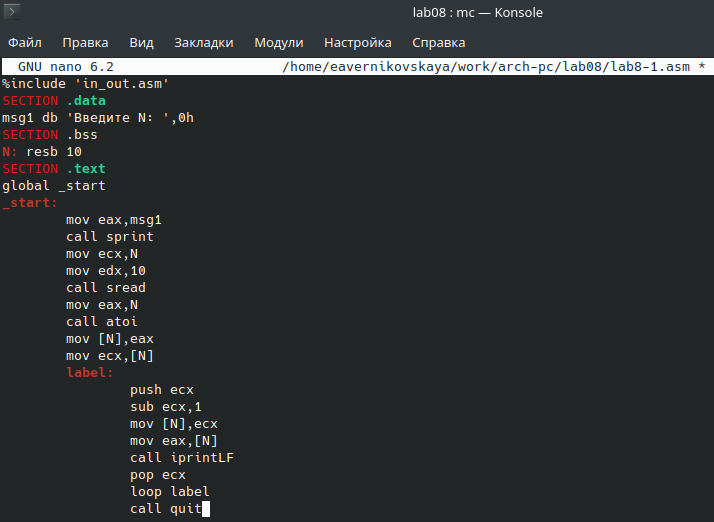
1. Соответствует ли число проходов цикла значению N введённому с клавиатуры?

* Нет. Число проходов не соответствует значению N введённому с клавиатуры.

Снова изменяем текст программы так, добавив команды push и pop (рис. [-@fig:009])

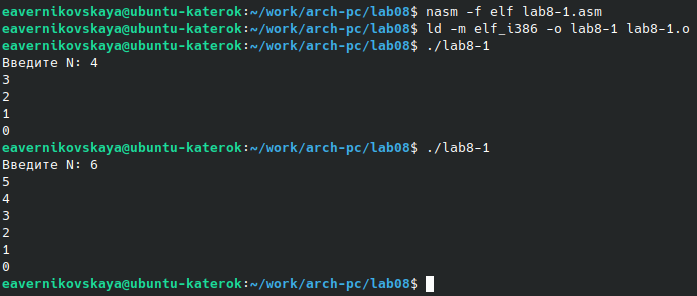
Изменённый текст программы:

%include 'in\_out.asm'  
SECTION .data  
msg1 db 'Введите N: ',0h  
SECTION .bss  
N: resb 10  
SECTION .text  
global \_start  
\_start:  
 mov eax,msg1  
 call sprint  
 mov ecx,N  
 mov edx,10  
 call sread  
 mov eax,N  
 call atoi  
 mov [N],eax  
 mov ecx,[N]  
 label:  
 push ecx  
 sub ecx,1  
 mov [N],ecx  
 mov eax,[N]  
 call iprintLF  
 pop ecx  
 loop label  
 call quit



Изменение программы

Опять создаём исполняемый файл и запускаем его (рис. [-@fig:010])



Исполняемый файл + запуск

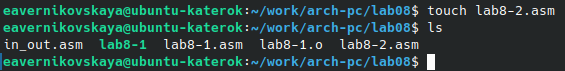
Ответы на вопросы:

1. Соответствует ли в данном случае число проходов цикла значению N введённому с клавиатуры?

* Да, теперь соответствует.

## Обработка аргументов командной строки

Создаём файл «lab8-2.asm» (рис. [-@fig:011])

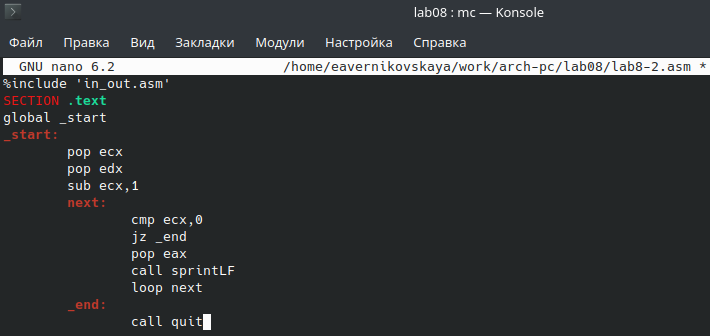


Создание файла «lab8-2.asm»

Вводим текст программы, которая выводит на экран аргументы командной строки (рис. [-@fig:012])

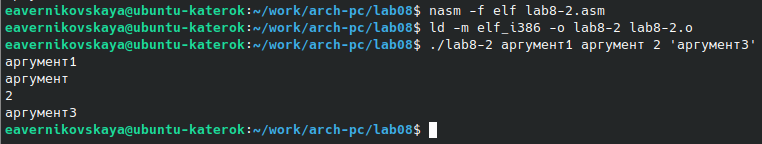
Текст программы:

%include 'in\_out.asm'  
SECTION .text  
global \_start  
\_start:  
 pop ecx  
 pop edx  
 sub ecx,1  
 next:  
 cmp ecx,0  
 jz \_end  
 pop eax  
 call sprintLF  
 loop next  
 \_end:  
 call quit



Ввод текста программы

Создаём исполняемый файл и запускаем его, указав аргументы (рис. [-@fig:013])



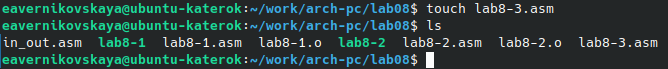
Создание исполняемого файла и его запуск

Ответы на вопросы:

1. Сколько аргументов было обработано программой?

* Программой было обработано 4 аргумента.

Создаём файл «lab8-3.asm» (рис. [-@fig:014])

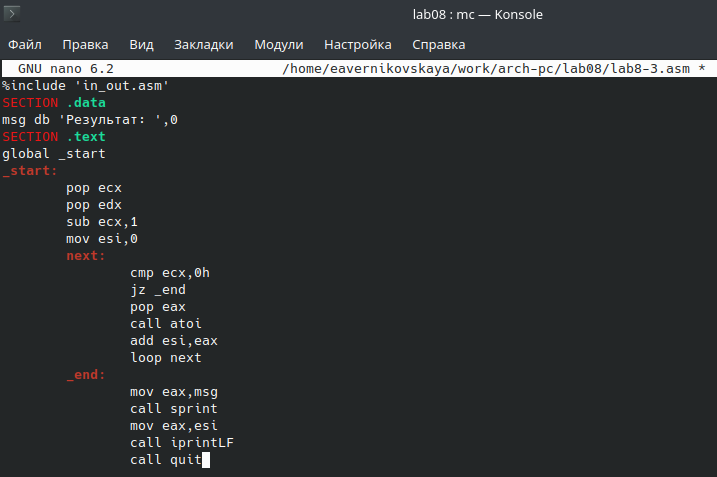


Создание файла «lab8-3.asm»

Вводим текст программы, которая вычисляет сумму аргументов командной строки (рис. [-@fig:015])

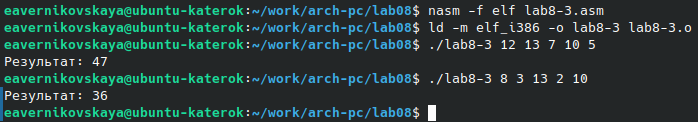
Текст программы:

%include 'in\_out.asm'  
SECTION .data  
msg db 'Результат: ',0  
SECTION .text  
global \_start  
\_start:  
 pop ecx  
 pop edx  
 sub ecx,1  
 mov esi,0  
 next:  
 cmp ecx,0h  
 jz \_end  
 pop eax  
 call atoi  
 add esi,eax  
 loop next  
 \_end:  
 mov eax,msg  
 call sprint  
 mov eax,esi  
 call iprintLF  
 call quit



Ввод текста программы

Создаём исполняемый файл и запускаем его, указав аргументы (рис. [-@fig:016])

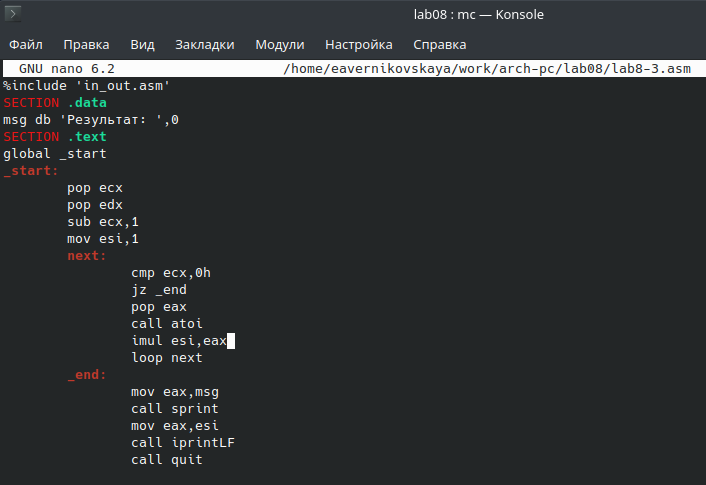


Создание исполняемого файла и его запуск

Изменяем текст программы так, чтобы она вычисляла произведение аргументов командной строки (рис. [-@fig:017])

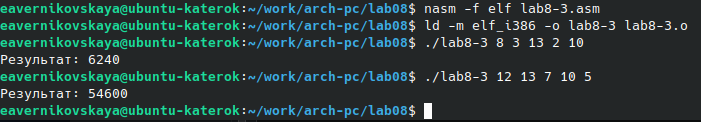
Изменённый текст программы:

%include 'in\_out.asm'  
SECTION .data  
msg db 'Результат: ',0  
SECTION .text  
global \_start  
\_start:  
 pop ecx  
 pop edx  
 sub ecx,1  
 mov esi,1  
 next:  
 cmp ecx,0h  
 jz \_end  
 pop eax  
 call atoi  
 imul esi,eax  
 loop next  
 \_end:  
 mov eax,msg  
 call sprint  
 mov eax,esi  
 call iprintLF  
 call quit



Изменение программы

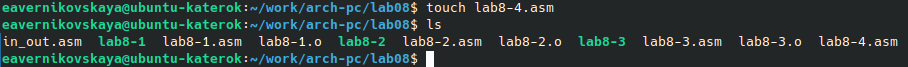
Создаём исполняемый файл и запускаем его, указав аргументы (рис. [-@fig:018])



Создание исполняемого файла и его запуск

## Задание для самостоятельной работы

Создаём файл «lab8-4.asm» (рис. [-@fig:019])

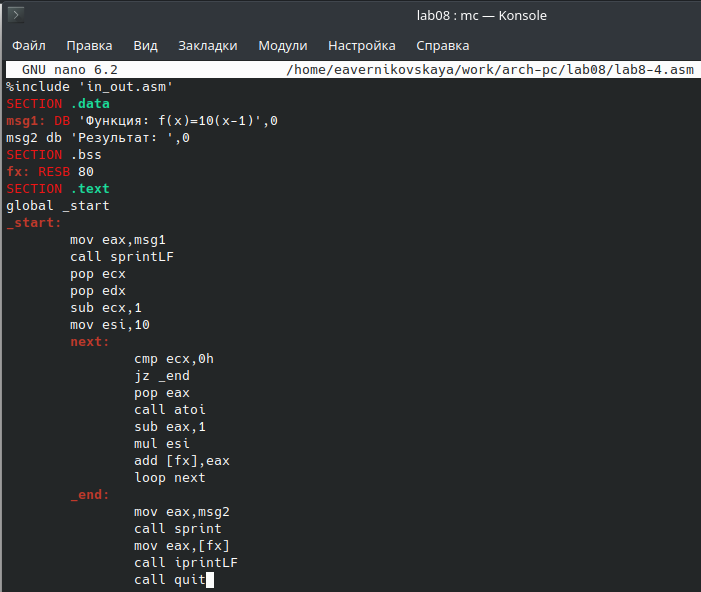


Создание файла «lab8-4.asm»

Вводим текст программы. Программа находит сумму значений функции f(x)=10(x-1) для разных x. Программа выводит значение f(x1) + f(x2) + … + f(xn) (рис. [-@fig:020])

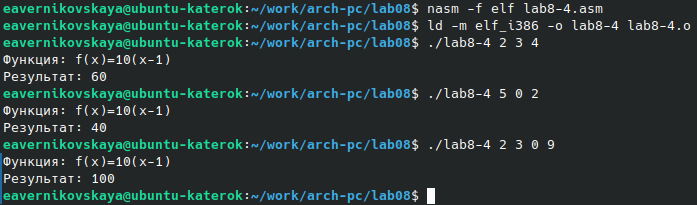
Текст программы:

%include 'in\_out.asm'  
SECTION .data  
msg1: DB 'Функция: f(x)=10(x-1)',0  
msg2 db 'Результат: ',0  
SECTION .bss  
fx: RESB 80  
SECTION .text  
global \_start  
\_start:  
 mov eax,msg1  
 call sprintLF  
 pop ecx  
 pop edx  
 sub ecx,1  
 mov esi,10  
 next:  
 cmp ecx,0h  
 jz \_end  
 pop eax  
 call atoi  
 sub eax,1  
 mul esi  
 add [fx],eax  
 loop next  
 \_end:  
 mov eax,msg2  
 call sprint  
 mov eax,[fx]  
 call iprintLF  
 call quit



Ввод текста программы

Создаём исполняемый файл и запускаем его, указав аргументы - разные значения x (рис. [-@fig:021])



Создание исполняемого файла и его запуск

# Выводы

В ходе выполнения лабораторной работы мы приобрели навыки написания программ с использованием циклов и обработкой аргументов командной строки.