

Лабораторная работа №8

Программирование цикла. Обработка аргументов командной строки.

Карачевцева Елизавета Васильевна

Содержание

1	Цель работы	3
2	Выполнение лабораторной работы	4
3	Самостотельная работа	12
	Вывод	15

1 Цель работы

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

2 Выполнение лабораторной работы

- 1) Я создала каталог lab08 и файл lab8-1.asm

```
evkarachevtseva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ mkdir lab08
evkarachevtseva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc$ cd lab08
evkarachevtseva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab08$ touch lab8-1.asm
evkarachevtseva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab08$ pwd
~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab08
evkarachevtseva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab08$ ls
lab8-1.asm
```

Рис. 2.1: Создание файла и каталога

- 2) В файл я ввела текст первой программы и создала исполняемый файл.

```
...vkarachevtseva/work/study/2024-2025/Архитектура компьютера/arch-pc/lab08/lab8-1.asm Измен
#include 'in_out.asm'
SECTION .data
msg1 db 'Введите N: ',0h
SECTION .bss
N: resb 10
SECTION .text
global _start
_start:
; ---- Вывод сообщения 'Введите N: '
mov eax,msg1
call sprint
; ---- Ввод 'N'
mov ecx, N
mov edx, 10
call sread
; ---- Преобразование 'N' из символа в число
mov eax,N
call atoi
mov [N],eax
; ----- Организация цикла
mov ecx,[N] ; Счетчик цикла, ecx=N
label:
mov [N],ecx
mov eax,[N]
call iprintLF ; Вывод значения N
loop label ; ecx=ecx-1 и если ecx не '0'
```

Рис. 2.2: Текст программы

```

evkarachevtseva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab08$ nasm -f elf lab8-1.asm
evkarachevtseva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
evkarachevtseva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab08$ ./lab8-1
Введите N: 5
5
4
3
2
1

```

Рис. 2.3: Запуск программы и проверка результата

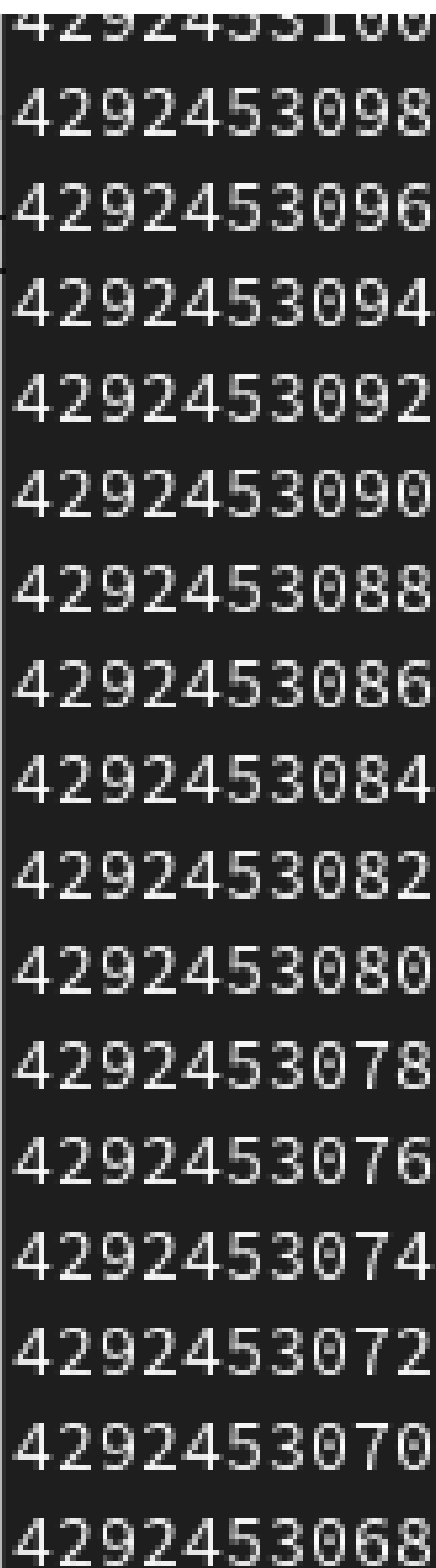
3) Я изменила текст программы, в теле цикла `label` добавила строку `sub eax,1`. Цикл закольцевался и стал бесконечным.

```

label:
sub ecx,1
mov [N],ecx
mov eax,[N]
call iprintLF ; Вывод значения N
loop label ; ecx=ecx-1 и если ecx не '0'
; переход на label
call quit

```

Рис. 2.4: Измененный текст программы



4292453100
4292453098
4292453096
4292453094
4292453092
4292453090
4292453088
4292453086
4292453084
4292453082
4292453080
4292453078
4292453076
4292453074
4292453072
4292453070
4292453068

Рис. 2.5: Запуск программы

- 4) Я изменила текст программы так, чтобы цикл и счетчик работал правильно. По итогу после изменения программы, число проходов циклов стало соответствовать числу введенному с клавиатуры.

```
label:
push ecx
sub ecx,1
mov [N],ecx
mov eax,[N]
call iprintLF ; Вывод значения N
pop ecx
loop label ; ecx=ecx-1 и если ecx не '0'
; переход на label
call quit
```

Рис. 2.6: Редактирование текста программы

```
evkarachevtseva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab
08$ nasm -f elf lab8-1.asm
evkarachevtseva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab
08$ ld -m elf_i386 -o lab8-1 lab8-1.o
evkarachevtseva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab
08$ ./lab8-1
Введите N: 5
4
3
2
1
0
```

Рис. 2.7: Запуск измененной программы

- 5) Я создала файл lab8-2.asm и ввела туда программу, которая выводит все аргументы, которые ввели. Программа выводит все 3 аргумента которые ввели, но в разной вариации.

```

...hevtseva/work/study/2024-2025/Архитектура компьютера/arch-pc/lab08/lab8-2.asm
%include 'in_out.asm'
SECTION .text
global _start
_start:
pop ecx ; Извлекаем из стека в ecx количество
; аргументов (первое значение в стеке)
pop edx ; Извлекаем из стека в edx имя программы
; (второе значение в стеке)
sub ecx, 1 ; Уменьшаем ecx на 1 (количество
; аргументов без названия программы)
next:
cmp ecx, 0 ; проверяем, есть ли еще аргументы
jz _end ; если аргументов нет выходим из цикла
; (переход на метку `_end`)
pop eax ; иначе извлекаем аргумент из стека
call sprintLF ; вызываем функцию печати
loop next ; переход к обработке следующего
; аргумента (переход на метку `next`)
_end:
call quit

```

Рис. 2.8: Текст программы для вывода аргументов

```

evkarachevtseva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab
08$ ./lab8-2 аргумент1 аргумент 2 'аргумент 3'
аргумент1
аргумент
2
аргумент 3

```

Рис. 2.9: Результаты работы программы

- 6) Я создала файл lab8-3.asm. Ввела текст программы и запустила ее. Программа вывела сумму чисел, которые я ввела.


```

...hevtseva/work/study/2024-2025/Архитектура компьютера/arch-pc/1
%include 'in_out.asm'
SECTION .data
msg db "Результат: ",0
SECTION .text
global _start
_start:
    pop ecx ; Извлекаем из стека в ecx количество
              ; аргументов (первое значение в стеке)
    pop edx ; Извлекаем из стека в edx имя программы
              ; (второе значение в стеке)
    sub ecx,1 ; Уменьшаем ecx на 1 (количество
              ; аргументов без названия программы)
    mov esi, 0 ; Используем esi для хранения
              ; промежуточных сумм
next:
    cmp ecx,0h ; проверяем, есть ли еще аргументы
    jz _end ; если аргументов нет выходим из цикла
              ; (переход на метку `_end`)
    pop eax ; иначе извлекаем следующий аргумент из стека
    call atoi ; преобразуем символ в число

```

Рис. 2.10: Текст программы lab8-3

```

evkarachevtseva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab
08$ nasm -f elf lab8-3.asm
evkarachevtseva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab
08$ ld -m elf_i386 -o lab8-3 lab8-3.o
evkarachevtseva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab
08$ ./lab8-3 1 2 3 4
Результат: 10
evkarachevtseva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab
08$ ./lab8-3 10 11 9
Результат: 30

```

Рис. 2.11: Результат работы программы

- 7) Я изменила программу, чтобы она выводила произведение введенных чисел.

```

%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
mov eax, 1

next:
cmp ecx,0
jz _end

pop eax
call atoi
mov ebx,eax
mov eax,esi
mul ebx
mov esi,eax
loop next

_end:
mov eax, msg
call sprint
mov eax, esi
call iprintLF
call quit

```

Рис. 2.12: Текст программы с произведением чисел

```
evkarachevtseva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab08$ nasm -f elf lab8-3.asm
evkarachevtseva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab08$ ld -m elf_i386 -o lab8-3
lab8-3.o
evkarachevtseva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab08$ ./lab8-3
Результат: 1
evkarachevtseva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab08$ ./lab8-3 1 2 3
Результат: 6
evkarachevtseva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab08$ ./lab8-3 1 2 3 4
Результат: 24
```

Рис. 2.13: Результаты работы программы с произведением

3 Самостотельная работа

Я написала программу, которая выводит сумму всех решений примера. В лабораторной работе №7, я получила 14 вариант, поэтому я писала программу для 14 варианта. Введенные числа я придумала сама, и посчитала их, чтобы проверить работу программы.

```

GNU nano 7.2
%include 'in_out.asm'
SECTION .data
prim DB 'f(x)=(x+1)*7',0
otv DB 'Результат: ',0
SECTION .text
GLOBAL _start
_start:

pop ecx

pop edx

sub ecx,1

mov esi,0

mov eax,prim
call sprintfLF
next:
cmp ecx,0
jz _end

mov ebx,7
pop eax
call atoi
add eax,1
mul ebx

add esi,eax

loop next

_end:
mov eax,otv
call sprintf
mov eax,esi
call iprintLF
call quit

```

Рис. 3.1: Текст программы в самостоятельной работе

```
evkarachevtseva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab08$ nasm -f elf lab8-4.asm
evkarachevtseva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab08$ ld -m elf_i386 -o lab8-4
lab8-4.o
evkarachevtseva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab08$ ./lab8-4 1 2 3 4
f(x)=(x+1)*7
Результат: 98
evkarachevtseva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab08$ mc
evkarachevtseva@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab08$ ./lab8-4 1 2 3 4 5
f(x)=(x+1)*7
Результат: 140
```

Рис. 3.2: Результаты работы программы

Вывод

Я приобрела навыки написания программы с использованием цикла.