# Отчёт по лабораторной работе №8

## По теме: Программирование цикла. Обработка аргументов командной строки.

Выполнил: Пателепень Филипп Максимович, НММбд-04-24.

## 8.1. Цель работы

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

## 8.3. Ход выполнения лабораторной работы

8.3.1. Реализация циклов в NASM

1. Я создал каталог для программ лабораторной работы № 8, перешёл в него и создал файл lab8-1.asm:

```
philipp916@philipp916-VirtualBox:~$ mkdir ~/work/arch-pc/lab08
philipp916@philipp916-VirtualBox:~$ cd ~/work/arch-pc/lab08
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ touch lab8-1.asm
```

2. Далее я ввел в файл lab8-1.asm текст программы из листинга 8.1. Создал исполняемый файл и проверьте его работу:

```
 GNU nano 4.8              /home/philipp916/work/arch-pc/lab08/lab8-1.asm              Изменён
%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
```

```
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 26
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
```

3. Я изменил текст программы добавив изменение значение регистра ecx в цикле, создал исполняемый файл и проверил его работу:

```
label:
sub ecx,1
mov [N],ecx
mov eax,[N]
call iprintLF
loop label
```

```
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 26
25
23
21
19
17
15
13
11
9
7
5
3
1
```

По итогу вывелись 13 значений. Все они являются нечетными, т.к. из-за команды ‘ecx,1’
из каждого следующего значения вычиталась единица. ‘ecx,1 = ecx - 1’

4. Для использования регистра ecx в цикле и сохранения корректности работы программы можно использовать стек. Я внес изменения в текст программы добавив команды ‘push’ и ‘pop’ (добавления в стек и извлечения из стека) для сохранения значения счетчика цикла loop:

```
label:
push ecx
sub ecx,1
mov [N],ecx
mov eax,[N]
call iprintL
pop ecx

loop label

call quit
```

5. Я создал исполняемый файл и проверил его работу. Теперь число проходов цикла соответствует значению *N* введенному с клавиатуры:

```
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 10
9
8
7
6
5
4
3
2
1
0
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$
```

8.3.2. Обработка аргументов командной строки

6. Я создал файл lab8-2.asm в каталоге ~/work/arch-pc/lab08 и ввел в него текст программы из листинга 8.2.:

```
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ touch lab8-2.asm
```

```
 GNU nano 4.8              /home/philipp916/work/arch-pc/lab08/lab8-2.asm              Изменён
%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
```

7. Потом создал исполняемый файл и запустил его. В результате программа вывела все 3 аргумента, которые были введены, но в разной вариации:

```
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-2.asm
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-2 lab8-2.o
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ ./lab8-2 аргумент1 аргумент 2 'аргумент 3'
аргумент1
аргумент
2
аргумент 3
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$
```

8. Я создал файл lab8-3.asm в каталоге ~/work/archpc/lab08 и ввел в него текст программы из листинга 8.3.:

```
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ touch lab8-3.asm
 GNU nano 4.8              /home/philipp916/work/arch-pc/lab08/lab8-3.asm              Изменён
%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
```

9. Я создал исполняемый файл и запустил его, указав аргументы. Программа вывела в терминал сумму чисел, которые я написал ранее:

```
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ ./lab8-3
Результат: 0
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ ./lab8-3 1 2 3 4
Результат: 10
```

10. Я изменил текст программы из листинга 8.3 для вычисления произведения аргументов командной строки,создал исполняемый файл и запустил его. В качестве проверки я ввел несколько комбинаций чисел. Программа работает корректно:

```
 GNU nano 4.8              /home/philipp916/work/arch-pc/lab08/lab8-3.asm              Изменён
%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
```

```
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ ./lab8-3 1 2 3 4 5 6
Результат: 720
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ ./lab8-3 1 916 100
Результат: 91600
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ ./lab8-3 10 11 12 13 14
Результат: 240240
```

## 8.4. Выполнение самостоятельной работы

1. Для выполнения самостоятельной работы  я создал файл lab8-4.asm:

```
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ touch lab8-4.asm
```

2. Далее я написал программу, которая находит сумму значений функции f(x) для x = x1 , x2 , ..., xn, т.е. программа выводит значение f(x1 ) + f(x2 ) + ... + f(xn). Значения xi передаются как аргументы. Вид моей функции - ‘15x + 2’ (11 вариант, полученный при выполнении лабораторной работы 6):

```
 GNU nano 4.8                 /home/philipp916/work/arch-pc/lab08/lab8-4.asm
%include 'in_out.asm'

SECTION .data
primer db "Функция: f(x)=15x+2 ",0
msg db "Результат: ",0

SECTION .text
global _start

_start:

        pop ecx
        pop edx
        sub ecx,1
        mov esi,0
        mov eax,primer
        call sprintLF

next:
        cmp ecx,0
        jz _end

        mov ebx,15
        pop eax
        call atoi
        mul ebx
        add eax,2
        add esi,eax
        loop next

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

3. Я создал исполняемый файл и проверил его работу на нескольких наборах x = x1 , x2 , ..., xn:

```
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-4.asm
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-4 lab8-4.o
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ ./lab8-4 1 2
Функция: f(x)=15x+2
Результат: 49
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ ./lab8-4 1 2 3
Функция: f(x)=15x+2
Результат: 96
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$ ./lab8-4 1 2 3 4
Функция: f(x)=15x+2
Результат: 158
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab08$
```

## Вывод

Вывод: выполняя данную лабораторную работу я приобрел полезные навыки написания программ с использованием цикла, а также обработки аргументов командной строки.

## Список литературы


1. GDB: The GNU Project Debugger. — URL: https://www.gnu.org/software/gdb/.

2. GNU Bash Manual. — 2016. — URL: https://www.gnu.org/software/bash/manual/.

3. Midnight Commander Development Center. — 2021. — URL: https://midnight-commander.org/.

4. NASM Assembly Language Tutorials. — 2021. — URL: https://asmtutor.com/.

5. *Newham C.* Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: http://www.amazon.com/Learningbash-Shell-Programming-Nutshell/dp/0596009658.

6. *Robbins A.* Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.

7. The NASM documentation. — 2021. — URL: https://www.nasm.us/docs.php.

8. *Zarrelli G.* Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.

9. *Колдаев В. Д.*, *Лупин С. А.* Архитектура ЭВМ. — М. : Форум, 2018.

10. *Куляс О. Л.*, *Никитин К. А.* Курс программирования на ASSEMBLER. — М. : Солон-Пресс, 2017.

11. *Новожилов О. П.* Архитектура ЭВМ и систем. — М. : Юрайт, 2016.

12. Расширенный ассемблер: NASM. — 2021. — URL: https://www.opennet.ru/docs/RUS/nasm/.

13. *Робачевский А.*, *Немнюгин С.*, *Стесик О.* Операционная система UNIX. — 2-е изд. — БХВ-Петербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.

14. *Столяров А.* Программирование на языке ассемблера NASM для ОС Unix. — 2-е изд. — М. : МАКС Пресс, 2011. — URL: http://www.stolyarov.info/books/asm_unix.

15. *Таненбаум Э.* Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).

16. *Таненбаум Э.*, *Бос Х.* Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science).