

# РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра математики и механики

## ОТЧЕТ

### ПО ЛАБОРАТОРНОЙ РАБОТЕ № 8

дисциплина:     *Архитектура компьютера*

Студент: Кондрацкая Александра Евгеньевна

Группа: НММбд-02-24

МОСКВА

2024 г.

## Оглавление

|                                       |   |
|---------------------------------------|---|
| 1 Цель работы.....                    | 3 |
| 2 Задание.....                        | 4 |
| 3 Выполнение лабораторной работы..... | 5 |
| 4 Выводы.....                         | 6 |

# 1 Цель работы

Освоить навыки программирования с применением циклов и управления аргументами командной строки.

## 2 Задание

- Реализация циклов в NASM
- Обработка аргументов командной строки

### 3 Выполнение лабораторной работы

Создадим каталог для программ лабораторной работы № 8, перейдем в него и создадим файл lab8-1.asm (рис. 1)

```
aecondrackaya@dk3n55 ~ $ mkdir ~/work/arch-pc/lab08
aecondrackaya@dk3n55 ~ $ cd ~/work/arch-pc/lab08
aecondrackaya@dk3n55 ~/work/arch-pc/lab08 $ touch lab8-1.asm
aecondrackaya@dk3n55 ~/work/arch-pc/lab08 $ ls
lab8-1.asm
aecondrackaya@dk3n55 ~/work/arch-pc/lab08 $
```

Рис. 1 Создание каталога

Введём в файл lab8-1.asm текст программы из листинга 8.1. (рис. 2)

```
1 %include 'in_out.asm'
2 SECTION .data
3 msg1 db 'Введите N: ',0h
4 SECTION .bss
5 N: resb 10
6 SECTION .text
7 global _start
8 _start:
9 ; ----- Вывод сообщения 'Введите N: '
10 mov eax,msg1
11 call sprint
12 ; ----- Ввод 'N'
13 mov ecx, N
14 mov edx, 10
15 call sread
16 ; ----- Преобразование 'N' из символа в число
17 mov eax,N
18 call atoi
19 mov [N],eax
20 ; ----- Организация цикла
21 mov ecx,[N] ; Счетчик цикла, 'ecx=N'
22 label:
23 mov [N],ecx
24 mov eax,[N]
25 call iprintLF ; Вывод значения 'N'
26 loop label ; 'ecx=ecx-1' и если 'ecx' не '0'
27 ; переход на 'label'
28 call quit
```

Рис. 2 Ввод программы

Создадим исполняемый файл и проверим его работу (рис. 3)

```
aekondrackaya@dk3n55 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
lab8-1.asm:1: error: unable to open include file 'in_out.asm': No such file or directory
aekondrackaya@dk3n55 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
aekondrackaya@dk3n55 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
ld: невозможно найти lab: Нет такого файла или каталога
ld: невозможно найти 8-1.o: Нет такого файла или каталога
aekondrackaya@dk3n55 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
aekondrackaya@dk3n55 ~/work/arch-pc/lab08 $ ./lab8-1
Введите N: 5
5
4
3
2
1
aekondrackaya@dk3n55 ~/work/arch-pc/lab08 $
```

Рис. 3 Создание исполняемого файла

Изменим текст программы добавив изменение значение регистра ecx в цикле (рис. 4)

```
1 %include 'in_out.asm'
2 SECTION .data
3 msg1 db 'Введите N: ',0h
4 SECTION .bss
5 N: resb 10
6 SECTION .text
7 global _start
8 _start:
9 ; ----- Вывод сообщения 'Введите N: '
10 mov eax,msg1
11 call sprint
12 ; ----- Ввод 'N'
13 mov ecx, N
14 mov edx, 10
15 call sread
16 ; ----- Преобразование 'N' из символа в число
17 mov eax,N
18 call atoi
19 mov [N],eax
20 ; ----- Организация цикла
21 mov ecx,[N] ; Счетчик цикла, 'ecx=N'
22 label:
23 sub ecx,1 ; 'ecx=ecx-1'
24 mov [N],ecx
25 mov eax,[N]
26 call iprintLF
27 loop label
```

Создадим исполняемый файл и проверим (рис. 5)

```
4293686560
4293686558
4293686556
4293686554
4293686552
4293686550
4293686548
4293686546
4293686544
4293686542
4293686540
4293686538
4293686536
4293686534
4293686532
```

Рис.5 Создание исполняемого файла

Регистр выдаёт ошибку и бесконечный ввод.

Исправим ошибку (рис. 6)

```
label:
push ecx ; добавление значения ecx в стек
sub ecx,1
mov [N],ecx
mov eax,[N]
call iprintLF
pop ecx
loop label
```

Рис. 6 Исправим ошибку

Создадим исполняемый файл и проверим (рис. 7)

```
aecondrackaya@dk3n55 ~/work/arch-pc/lab08 $ ./lab8-1
Введите N: 3
2
1
0
```

Рис. 7 Исполняемый файл

Теперь всё работает правильно. Число выводов совпадает с числом N.

Создадим файл lab8-2.asm в каталоге ~/work/arch-pc/lab08 (рис. 8)

```
aecondrackaya@dk3n55 ~/work/arch-pc/lab08 $ touch lab8-2.asm
aecondrackaya@dk3n55 ~/work/arch-pc/lab08 $ ls
in_out.asm  lab8-1  lab8-1.asm  lab8-1.o  lab8-2.asm
aecondrackaya@dk3n55 ~/work/arch-pc/lab08 $
```

Рис. 8 Создание файл

введём в него текст программы из листинга 8.2. (рис. 9)

```
1 %include 'in_out.asm'
2 SECTION .text
3 global _start
4 _start:
5 pop ecx ; Извлекаем из стека в 'ecx' количество
6 ; аргументов (первое значение в стеке)
7 pop edx ; Извлекаем из стека в 'edx' имя программы
8 ; (второе значение в стеке)
9 sub ecx, 1 ; Уменьшаем 'ecx' на 1 (количество
10 ; аргументов без названия программы)
11 next:
12 cmp ecx, 0 ; проверяем, есть ли еще аргументы
13 jz _end ; если аргументов нет выходим из цикла
14 ; (переход на метку '_end')
15 pop eax ; иначе извлекаем аргумент из стека
16 call sprintLF ; вызываем функцию печати
17 loop next ; переход к обработке следующего
18 ; аргумента (переход на метку 'next')
19 _end:
20 call quit
```

Рис. 9 Ввод программы

Создадим исполняемый файла и запустим(рис. 10)

```
aecondrackaya@dk3n55 ~/work/arch-pc/lab08 $ ./lab8-2 аргумент1 аргумент 2 'аргумент 3'
аргумент1
аргумент
2
аргумент 3
aecondrackaya@dk3n55 ~/work/arch-pc/lab08 $
```

Рис. 10 Исполняемый файл  
программа обрабатывает три аргумента



Создадим файл lab8-3.asm в каталоге ~/work/arch-pc/lab08 (рис. 11)

```
aekondrackaya@dk3n55 ~/work/arch-pc/lab08 $ touch lab8-3.asm
aekondrackaya@dk3n55 ~/work/arch-pc/lab08 $ ls
in_out.asm  lab8-1  lab8-1.asm  lab8-1.o  lab8-2  lab8-2.asm  lab8-2.o  lab8-3.asm
aekondrackaya@dk3n55 ~/work/arch-pc/lab08 $
```

Рис. 11 Создание файлов

Введем в него команду из листинга 8.3 (рис. 12)

```
1 %include 'in_out.asm'
2 SECTION .data
3 msg db "Результат: ",0
4 SECTION .text
5 global _start
6 _start:
7  pop ecx ; Извлекаем из стека в 'ecx' количество
8  ; аргументов (первое значение в стеке)
9  pop edx ; Извлекаем из стека в 'edx' имя программы
10 ; (второе значение в стеке)
11 sub ecx,1 ; Уменьшаем 'ecx' на 1 (количество
12 ; аргументов без названия программы)
13 mov esi, 0 ; Используем 'esi' для хранения
14 ; промежуточных сумм
15 next:
16 cmp ecx,0h ; проверяем, есть ли еще аргументы
17 jz _end ; если аргументов нет выходим из цикла
18 ; (переход на метку '_end')
19 pop eax ; иначе извлекаем следующий аргумент из стека
20 call atoi ; преобразуем символ в число
21 add esi,eax ; добавляем к промежуточной сумме
22 ; след. аргумент 'esi=esi+eax'
23 loop next ; переход к обработке следующего аргумента
24 _end:
25 mov eax, msg ; вывод сообщения "Результат: "
26 call sprint
27 mov eax, esi ; записываем сумму в регистр 'eax'
28 call iprintLF ; печать результата
29 call quit ; завершение программы
```

Рис. 12 Ввод программы

Создадим исполняемый файл и запустим (рис. 13)

```

aekondrackaya@dk3n55 ~/work/arch-pc/lab08 $ nasm -f elf lab8-3.asm
aekondrackaya@dk3n55 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-3 lab8-3.o
aekondrackaya@dk3n55 ~/work/arch-pc/lab08 $ ./lab8-3
Результат: 0
aekondrackaya@dk3n55 ~/work/arch-pc/lab08 $ ./main 12 13 7 10 5
bash: ./main: Нет такого файла или каталога
aekondrackaya@dk3n55 ~/work/arch-pc/lab08 $ ./lab8-3 12 13 7 10 5
Результат: 47
aekondrackaya@dk3n55 ~/work/arch-pc/lab08 $ █

```

Рис. 13 Исполняемый файл

Изменение файл, чтобы происходило умножение (рис. 14)

```

1 %include 'in_out.asm'
2 SECTION .data
3 msg DB "Результат: ",0
4 SECTION .text
5 GLOBAL _start
6 _start:
7 pop ecx
8 pop edx
9 sub ecx,1
10 mov esi, 1
11 mov eax, 1
12 next:
13 cmp ecx,0
14 jz _end
15 pop eax
16 call atoi
17 mov ebx,eax
18 mov eax,esi
19 mul ebx
20 mov esi,eax
21 loop next
22 _end:
23 mov eax,msg
24 call sprint
25 mov eax,esi
26 call iprintLF
27
28 call quit

```

Создадим исполняемый файл и запустим (рис. 15)

```
aecondrackaya@dk3n55 ~/work/arch-pc/lab08 $ nasm -f elf lab8-3.asm
aecondrackaya@dk3n55 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-3 lab8-3.o
aecondrackaya@dk3n55 ~/work/arch-pc/lab08 $ ./lab8-3 3 4 2 8
Результат: 192
aecondrackaya@dk3n55 ~/work/arch-pc/lab08 $
```

Рис. 15 Исполняемый файл

## САМОСТОЯТЕЛЬНАЯ РАБОТА

Создадим файл lab8-4.asm ,напишем туда листинг для решения функции варианта №13.

```
aecondrackaya@dk3n55 ~/work/arch-pc/lab08 $ touch lab8-4.asm
aecondrackaya@dk3n55 ~/work/arch-pc/lab08 $ ls
in_out.asm lab8-1 lab8-1.asm lab8-1.o lab8-2 lab8-2.asm lab8-2.o lab8-3 lab8-3.asm lab8-3.o lab8-4.asm
```

Рис. 16 Создание файла

```
1 %include 'in_out.asm'
2
3 SECTION .data
4 prim DB 'f(x) = 12x - 7',0
5 otv DB 'Результат: ',0
6
7 SECTION .text
8 GLOBAL _start
9
10 _start:
11     pop ecx
12     sub ecx, 1
13
14     mov esi, 0
15     mov eax, prim
16     call sprintf
17
18 next:
19     cmp ecx, 0
20     jz _end
21
22     pop eax
23     call atoi
24
25     ; Теперь выполняем f(x) = 12x - 7
26     mov ebx, 12
27     imul eax, ebx
28     sub eax, 7
29
30     add esi, eax
31
32     dec ecx
33     jmp next
34
35 _end:
36     mov eax, otv
37     call sprintf
38     mov eax, esi
39     call iprintLF
40     call quit
```

Рис. 17 Ввод программы

```
aecondrackaya@dk3n55 ~/work/arch-pc/lab08 $ nasm -f elf lab8-4.asm
aecondrackaya@dk3n55 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-4 lab8-4.o
aecondrackaya@dk3n55 ~/work/arch-pc/lab08 $ ./lab8-4 1 2 3 4
f(x) = 12x - 7
Результат: 44
```

Рис 18 Исполняемый файл и результат

## **4 Выводы**

Успешное освоение навыков программирования позволит эффективно применять циклы и управлять аргументами командной строки.