

# **тчёт по лабораторной работе №8**

**Дисциплина: архитектура компьютера**

Сидельников Андрей Владимирович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Самостоятельное задание</b>	<b>14</b>
<b>4</b>	<b>Выводы</b>	<b>16</b>

## Список таблиц

# Список иллюстраций

figno	Создание файла . . . . .	6
figno	Файл с программой . . . . .	7
2.1	Запуск программы . . . . .	7
figno	Изменённый файл . . . . .	8
2.2	Запуск команды . . . . .	8
figno	Исполняемый файл . . . . .	9
2.3	Запуск программы . . . . .	10
figno	Файл листинг . . . . .	10
2.4	Запуск программы . . . . .	11
figno	Листинг программы . . . . .	11
2.5	Запуск программы . . . . .	12
figno	Изменение файла . . . . .	12
2.6	Запуск изменённой программы . . . . .	13
figno	Программа в файле . . . . .	15
3.1	Проверка программы . . . . .	15

# 1 Цель работы

Целью работы является приобретение навыков написания программ с использованием циклов и обработкой аргументов командной строки.

## 2 Выполнение лабораторной работы

Создал файл lab8-1.asm и вписал в него команду их листинга 8.1. (рис. 2.1).

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

Создание файла

```
lab8-1.asm
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
```

Файл с программой

```
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 lab8-1.o -o lab8-1
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 5
5
4
3
2
1
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 3
3
2
1
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab08$
```

Рис. 2.1: Запуск программы

Программа запускает бесконечный цикл при нечетном N и выводит только нечетные числа при четном N. (рис. 2.2).

```
lab8-1.asm
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
28 ; переход на `label`
29 call quit
```

Изменённый файл

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

Рис. 2.2: Запуск команды

Создал исполняемый файл и проверил его работу. Программа выводит числа от N-1 до 0, число проходов цикла соответствует N (рис. 2.3).



## lab8-1.asm

```
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 push ecx ; добавление значения ecx в стек
24 sub ecx,1
25 mov [N],ecx
26 mov eax,[N]
27 call iprintLF
28 pop ecx ; извлечение значения ecx из стека
29 loop label
30 ; переход на `label`
31 call quit
```

Исполняемый файл

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

Рис. 2.3: Запуск программы

Создал файл lab8-2.asm и ввел в него текст программы из листинга 8.2 и запустил его (рис. 2.4). Программа обработала 5 аргументов. Аргументами считаются слова и числа, разделенные пробелом.

```
lab8-2.asm
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
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 sprintf ; вызываем функцию печати
17 loop next ; переход к обработке следующего
18 ; аргумента (переход на метку `next`)
19 _end:
20 call quit
```

Файл листинг

```

avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-2.asm
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 lab8-2.o -o lab8-2
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab08$ ./lab8-2 1 2 3
1
2
3
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab08$ ./lab8-2 1 2 3
1
2
3
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab08$ ./lab8-2 1 2 '3'
1
2
3
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab08$ ./lab8-2 аргумент1 аргумент 2 'аргумент 3'
аргумент1
аргумент
2
аргумент 3

```

Рис. 2.4: Запуск программы

Создал файл lab8-3.asm и ввел в него текст программы из листинга 8.3, и запустил его (рис. 2.5).

```

lab8-3.asm
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

```

Листинг программы

```

avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab08$ touch lab8-3.asm
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 lab8-3.o -o lab8-3
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab08$ ./lab8-3
Результат: 0
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab08$ ./lab8-3 12 13 7 10 2
Результат: 44

```

Рис. 2.5: Запуск программы

Изменил текст программы из листинга 8.3 для вычисления произведения аргументов командной строки (рис. 2.6).

```

lab8-3.asm
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
10 ; программы
11 sub ecx,1 ; Уменьшаем `ecx` на 1 (количество
12 ; аргументов без названия программы)
13 mov esi, 1 ; 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
20 ; стека
21 call atoi ; преобразуем символ в число
22 mov ebx,eax
23 mov eax,esi
24 mul ebx
25 mov esi,eax ; добавляем к промежуточной сумме
26 ; след. аргумент `esi=esi+eax`
27 loop next ; переход к обработке следующего
28 ; аргумента
29 _end:
30 mov eax, msg ; вывод сообщения "Результат: "
31 call sprintf
32 mov eax, esi ; записываем сумму в регистр `eax`
33 call iprintLF ; печать результата
34 call quit ; завершение программы

```

Изменение файла

```
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 lab8-3.o -o lab8-3
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab08$ ./lab8-3 12 13 7 10 2
Результат: 21840
```

Рис. 2.6: Запуск изменённой программы

### 3 Самостоятельное задание

Для варианта 6  $f(x) = 4x - 3$ , при подстановки разных значений переменной, ответы сходятся (рис. 3.1).

```

lab8-4.asm
1 %include 'in_out.asm'
2 SECTION .data
3 msg db "Результат: ",0
4 fx: db 'f(x)= 4x - 3',0
5
6 SECTION .text
7 global _start
8 _start:
9 mov eax, fx
10 call sprintLF
11 pop ecx
12 pop edx
13 sub ecx,1
14 mov esi, 0
15
16 next:
17 cmp ecx,0h
18 jz _end
19 pop eax
20 call atoi
21 mov ebx,4
22 mul ebx
23 sub eax,3
24 add esi,eax
25
26 loop next
27
28 _end:
29 mov eax, msg
30 call sprint
31 mov eax, esi
32 call iprintLF
33 call quit

```

Программа в файле

```

avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab08$ touch lab8-4.asm
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-4.asm
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 lab8-4.o -o lab8-4
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab08$ ./lab8-4
f(x)= 4x - 3
Результат: 0
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab08$ ./lab8-4 1
f(x)= 4x - 3
Результат: 1
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab08$ ./lab8-4 2
f(x)= 4x - 3
Результат: 5
avsideljnikov@avsideljnikov-VirtualBox:~/work/arch-pc/lab08$ ./lab8-4 1 2 3 4 55
f(x)= 4x - 3
Результат: 245

```

Рис. 3.1: Проверка программы

## 4 Выводы

Я освоил работы со стеком, циклом и аргументами на ассемблере `naasm`.