

Лабараторная работа №08.

НКАбд-01-24

Подготовил:

Холов Икром Комронович

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Задание для самостоятельной работы	18
4	Выводы	21

Список иллюстраций

2.1	Создал lab08-1.asm	6
2.2	Заполнил lab08-1	7
2.3	Проверил lab08-1	8
2.4	Изменил_lab08-11	9
2.5	Проверил_измененный_lab08-1	10
2.6	Исправил_lab08-1	11
2.7	Проверил_исправление	12
2.8	Создал_lab08-2	12
2.9	Заполнил lab08-2	13
2.10	Проверил_lab08-2	13
2.11	Создал lab08-3.asm	14
2.12	Заполнил lab08-3.asm	15
2.13	Проверил lab08-3.asm	16
2.14	Изменил_lab08-3	16
2.15	Проверил_измененную_lab08-3	17
3.1	Заполнил_hw	19
3.2	Проверил_hw	20

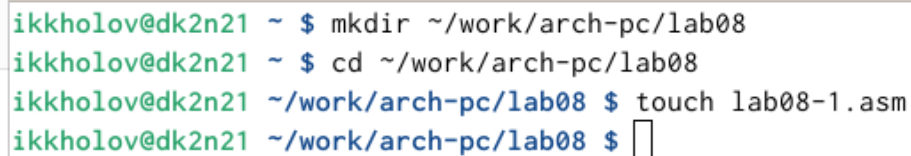
Список таблиц

1 Цель работы

Освоить написание программ с использованием циклов и обработкой аргументов командной строки.

2 Выполнение лабораторной работы

1. Создал каталог для программ, перешел в него и создал файл lab8-1.asm (рис. 2.1)



```
ikkholov@dk2n21 ~ $ mkdir ~/work/arch-pc/lab08
ikkholov@dk2n21 ~ $ cd ~/work/arch-pc/lab08
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ touch lab08-1.asm
ikkholov@dk2n21 ~/work/arch-pc/lab08 $
```

Рис. 2.1: Создал lab08-1.asm

2. Заполнил lab8-1.asm (рис. 2.2)

```

1 %include 'in_out.asm'
2
3 SECTION .data
4     msg1 db 'Введите N: ',0h
5
6 SECTION .bss
7     N: resb 10
8
9 SECTION .text
10 global _start
11
12 _start:
13     ; ----- Вывод сообщения 'Введите N: '
14     mov eax,msg1
15     call sprint
16
17     ; ----- Ввод 'N'
18
19     mov ecx, N
20     mov edx, 10
21     call sread
22
23     ; ----- Преобразование 'N' из символа в число
24
25     mov eax,N
26     call atoi
27     mov [N],eax
28
29     ; ----- Организация цикла
30
31     mov ecx,[N] ; Счетчик цикла, 'ecx=N'
32 label:
33     mov [N],ecx
34     mov eax,[N]
35     call iprintLF ; Вывод значения 'N'
36     loop label ; 'ecx=ecx-1' и если 'ecx' не '0'
37     ; переход на 'label'
38     call quit

```

Рис. 2.2: Заполнил lab08-1

3. Создал исполняемый файл и запустил его (рис. 2.3)

```
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ nasm -f elf lab08-1.asm
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab08-1 lab08-1.o
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ ./lab08-1
Введите N: 2
2
1
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ ./lab08-1
Введите N: 10
10
9
8
7
6
5
4
3
2
1
ikkholov@dk2n21 ~/work/arch-pc/lab08 $
```

Рис. 2.3: Проверил lab08-1

4. Изменил текст программы (рис. 2.4)


```

1 %include 'in_out.asm'
2
3 SECTION .data
4     msg1 db 'Введите N: ',0h
5
6 SECTION .bss
7     N: resb 10
8
9 SECTION .text
10 global _start
11
12 _start:
13     ; ----- Вывод сообщения 'Введите N: '
14     mov eax,msg1
15     call sprint
16
17     ; ----- Ввод 'N'
18
19     mov ecx, N
20     mov edx, 10
21     call sread
22
23     ; ----- Преобразование 'N' из символа в число
24
25     mov eax,N
26     call atoi
27     mov [N],eax
28
29     ; ----- Организация цикла
30
31     mov ecx,[N]      ; Счетчик цикла, 'ecx=N'
32 label:
33     sub ecx,1
34     mov [N],ecx
35     mov eax,[N]
36     call iprintLF    ; Вывод значения 'N'
37     loop label       ; 'ecx=ecx-1' и если 'ecx' не '0'
38     ; переход на 'label'
39     call quit

```

Рис. 2.4: Изменил_lab08-11

5. Создал исполняемый файл и запустил его (рис. 2.5)

```
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ gedit lab08-1.asm
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ nasm -f elf lab08-1.asm
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab08-1 lab08-1.o
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ ./lab08-1
Введите N: 10
9
7
5
3
1
ikkholov@dk2n21 ~/work/arch-pc/lab08 $
```

Рис. 2.5: Проверил_измененный_lab08-1

6. Изменил текст программы, с использованием стека в программу для сохранения корректности работы программы (рис. 2.6)

```

1 %include 'in_out.asm'
2
3 SECTION .data
4 msg1 db 'Введите N: ',0h
5
6 SECTION .bss
7 N: resb 10
8
9 SECTION .text
10 global _start
11
12 _start:
13     ; ----- Вывод сообщения 'Введите N: '
14 mov eax,msg1
15 call sprint
16
17     ; ----- Ввод 'N'
18
19 mov ecx, N
20 mov edx, 10
21 call sread
22
23     ; ----- Преобразование 'N' из символа в число
24
25 mov eax,N
26 call atoi
27 mov [N],eax
28
29     ; ----- Организация цикла
30
31 mov ecx,[N] ; Счетчик цикла, 'ecx=N'
32 label:
33 push ecx
34 sub ecx,1
35 mov [N],ecx
36 mov eax,[N]
37 call iprintLF ; Вывод значения 'N'
38 pop ecx
39 loop label ; 'ecx=ecx-1' и если 'ecx' не '0'
40 ; переход на 'label'
41 call quit

```

Рис. 2.6: Исправил_lab08-1

7. Проверка выполнения (рис. 2.7)

```
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ gedit lab08-1.asm
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ nasm -f elf lab08-1.asm
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab08-1 lab08-1.o
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ ./lab08-1
Введите N: 10
9
8
7
6
5
4
3
2
1
0
ikkholov@dk2n21 ~/work/arch-pc/lab08 $
```

Рис. 2.7: Проверил_исправление

8. Создал файл lab8-2.asm (рис. 2.8)

```
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ touch lab08-2.asm
ikkholov@dk2n21 ~/work/arch-pc/lab08 $
```

Рис. 2.8: Создал_lab08-2

9. Заполнил lab8-2.asm (рис. 2.9)

```

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

```

Рис. 2.9: Заполнил lab08-2

10. Создал исполняемый файл и запустил его, указав аргументы. Программа обработала 4 аргумента (рис. 2.10)

```

ikkholov@dk2n21 ~/work/arch-pc/lab08 $ nasm -f elf lab08-2.asm
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab08-2 lab08-2.o
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ ./lab08-2 аргумент1 аргумент 2 'аргумент 3'
аргумент1
аргумент
2
аргумент 3
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ 

```

Рис. 2.10: Проверил_lab08-2

11. Создал файл lab8-3.asm (рис. [@-fig:011])

```
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ touch lab08-3.asm
ikkholov@dk2n21 ~/work/arch-pc/lab08 $
```

Рис. 2.11: Создал lab08-3.asm

12. Заполнил lab8-3.asm (рис. 2.12)

```

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

```

Рис. 2.12: Заполнил lab08-3.asm

13. Создал исполняемый файл и запустил его (рис. 2.13)

```

ikkholov@dk2n21 ~/work/arch-pc/lab08 $ nasm -f elf lab08-3.asm
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab08-3 lab08-3.o
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ ./lab08-3 12 13 7 10 5
Результат: 47
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ █

```

Рис. 2.13: Проверил lab08-3.asm

14. Изменил текст программы для вычисления произведения аргументов командной строки (рис. 2.14)

```

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

```

Рис. 2.14: Изменил_lab08-3

15. Проверка программы (рис. 2.15)


```
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ nasm -f elf lab08-3.asm
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab08-3 lab08-3.o
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ ./lab08-3 2 2
Результат: 4
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ ./lab08-3 23 43
Результат: 989
ikkholov@dk2n21 ~/work/arch-pc/lab08 $
```

Рис. 2.15: Проверил_измененную_lab08-3

3 Задание для самостоятельной работы

1. Программа для нахождения вычисления значений функции $f(x)$ для разных x (рис. 3.1) (рис. 3.2)

```

1 %include 'in_out.asm'
2
3 SECTION .data
4 msg1 db 'Функция: f(x)=30x+11',0
5 msg2 db 'Результат: ',0
6
7 SECTION .text
8 global _start
9
10 _start:
11 mov eax, msg1
12 call sprintf
13
14 pop ecx
15 pop edx
16 dec ecx
17 mov esi,0
18
19 next:
20 cmp ecx, 0h
21 jz _end
22
23 pop eax
24 call atoi
25
26 mov edx, 30
27 mul edx
28 sub eax, 11
29
30 add esi, eax
31
32 loop next
33
34 _end:
35 mov eax, msg2
36 call sprintf
37
38 mov eax, esi
39 call iprintLF
40
41 call quit

```

Рис. 3.1: Заполнил_hw

```
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ nasm -f elf hw.asm
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o hw hw.o
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ ./hw 1
Функция:  $f(x)=30x-11$ 
Результат: 19
ikkholov@dk2n21 ~/work/arch-pc/lab08 $ ./hw 49
Функция:  $f(x)=30x-11$ 
Результат: 1459
ikkholov@dk2n21 ~/work/arch-pc/lab08 $
```

Рис. 3.2: Проверил_hw

4 Выводы

Освоил программу с использованием циклов и обработки аргументов командной строки