

Отчёт по лабораторной работе %9

Дисциплина: 'архитектура компьютеров'

Бабенко Роман Игоревич

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	15

Список иллюстраций

2.1	Создание каталога и файла	6
2.2	Программа из листинга	7
2.3	Проверка программы	8
2.4	Изменяем текст программы	8
2.5	Изменяем программу	9
2.6	Проверяем работу изменённой программы	9
2.7	Проверка второй программы	10
2.8	Программа для суммирования аргументов	10
2.9	Изменённая программа	11
2.10	Проверка программы	12
2.11	Написанная программы	13
2.12	Ура! Всё работает)	14

Список таблиц

1 Цель работы

Приобрести навыки написания программ с использованием циклов и обработкой аргументов командной строки.

2 Выполнение лабораторной работы

Создаём каталог для программ лабораторной работы %9 и файл lab9-1.asm
(рис. 2.1)

```
ribabenko@dk6n57 ~ $ mkdir ~/work/arch-pc/lab09
ribabenko@dk6n57 ~ $ cd ~/work/arch-pc/lab09
ribabenko@dk6n57 ~/work/arch-pc/lab09 $ touch lab9-1.asm
ribabenko@dk6n57 ~/work/arch-pc/lab09 $
```

Рис. 2.1: Создание каталога и файла

Переписываем в файл программу из листинга 9.1 (рис. 2.2)

```

%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

```

Рис. 2.2: Программа из листинга

Проверка работы файла (рис. 2.3)

```
ribabenko@dk6n57 ~/work/arch-pc/lab09 $ nasm -f elf lab9-1.asm
ribabenko@dk6n57 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o lab9-1 lab9-1.o
ribabenko@dk6n57 ~/work/arch-pc/lab09 $ ./lab9-1
Введите N: 5
5
4
3
2
1
```

Рис. 2.3: Проверка программы

Изменяем текст программы, и проверяем - это привело к её некорректной работе (рис. 2.4)

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

Рис. 2.4: Изменяем текст программы

Вносим изменения в текст программы, добавляем команды push и pop (рис. 2.5)


```

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

```

Рис. 2.5: Изменяем программу

Создаём исполняемый файл и проверяем его работу. Число проходов цикла соответствует N (рис. 2.6)

```

ribabenko@dk6n57 ~/work/arch-pc/lab09 $ nasm -f elf lab9-1.asm
ribabenko@dk6n57 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o lab9-1 lab9-1.o
ribabenko@dk6n57 ~/work/arch-pc/lab09 $ ./lab9-1
Введите N: 5
4
3
2
1
0
ribabenko@dk6n57 ~/work/arch-pc/lab09 $

```

Рис. 2.6: Проверяем работу изменённой программы

Создаём новый файл, записываем в него текст из листинга 9.2 и запускаем

его, указав предложенные аргументы. Программой было обработано четыре аргумента (рис. 2.7)

```
ribabenko@dk6n57 ~/work/arch-pc/lab09 $ nasm -f elf lab9-2.asm
ribabenko@dk6n57 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o lab9-2 lab9-2.o
ribabenko@dk6n57 ~/work/arch-pc/lab09 $ ./lab9-2
ribabenko@dk6n57 ~/work/arch-pc/lab09 $ ./lab9-2 аргумент1 аргумент 2 'аргумент 3'
аргумент1
аргумент
2
аргумент 3
ribabenko@dk6n57 ~/work/arch-pc/lab09 $
```

Рис. 2.7: Проверка второй программы

Также создаём файл lab9-3 и переписываем в него программу из листинга 9.3, проверяем работу файла (рис. 2.8)

```
ribabenko@dk6n57 ~/work/arch-pc/lab09 $ touch lab9-3.asm
nasm -f elf lab9-3.asm
ribabenko@dk6n57 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o lab9-3 lab9-3.o
ribabenko@dk6n57 ~/work/arch-pc/lab09 $ ./lab9-3
Результат: 0
ribabenko@dk6n57 ~/work/arch-pc/lab09 $ ./lab9-3 12 13 7 10 5
Результат: 47
ribabenko@dk6n57 ~/work/arch-pc/lab09 $
```

Рис. 2.8: Программа для суммирования аргументов

Далее изменяем программу так, чтобы она вычисляла произведение аргументов командной строки (рис. 2.9)

```

SECTION .data
msg db "Результат: ",0
SECTION .text
global _start

_start:
    pop ecx

    pop edx

    sub ecx,1

    mov esi, 1

next:
    cmp ecx,0h
    jz _end

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

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

```

Рис. 2.9: Изменённая программа

Проверяем работоспособность изменённой программы (рис. 2.10)

```
nasm -f elf lab9-3.asm
ribabenko@dk6n57 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o lab9-3 lab9-3.o
ribabenko@dk6n57 ~/work/arch-pc/lab09 $ ./lab9-3 10 3
Результат: 30
ribabenko@dk6n57 ~/work/arch-pc/lab09 $ ./lab9-3
Результат: 1
ribabenko@dk6n57 ~/work/arch-pc/lab09 $ ./lab9-3 1 2 3 4
Результат: 24
ribabenko@dk6n57 ~/work/arch-pc/lab09 $
```

Рис. 2.10: Проверка программы

#Задания для самостоятельной работы

Напишем программу, находящую сумму значений функции про седьмому варианту. (рис. 2.11)

```
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 eax, 2
    mov ebx, 3
    mul ebx
    add esi, eax
    loop next

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

Рис. 2.11: Написанная программы

Проверка написанной нами программы (рис. 2.12)

```
nasm -f elf zad.asm
ribabenko@dk6n57 ~/work/arch-pc/lab09 $ ld -m elf_i386 -o zad zad.o
ribabenko@dk6n57 ~/work/arch-pc/lab09 $ ./zad
Результат: 0
ribabenko@dk6n57 ~/work/arch-pc/lab09 $ ./zad 1 2
Результат: 21
ribabenko@dk6n57 ~/work/arch-pc/lab09 $ ./zad 1 2 3
Результат: 36
ribabenko@dk6n57 ~/work/arch-pc/lab09 $
```

Рис. 2.12: Ура! Всё работает)

3 Выводы

В ходе выполнения данной лабораторной работы я приобрёл навыки написания программ с использованием циклов и обработки аргументов командной строки.