

Отчёт по лабораторной работе №8

Программирование цикла. Обработка аргументов командной строки.Файл

Исупов Олег Денисович

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Задания для самостоятельной работы	11
4	Выводы	13

Список иллюстраций

2.1	Создание файла	6
2.2	Заполнение файла	6
2.3	Проверка	7
2.4	Изменение	7
2.5	Проверка	7
2.6	Изменение и проверка	8
2.7	Создание файла	8
2.8	Заполнение файла	8
2.9	Проверка	9
2.10	Создание и заполнение файла	9
2.11	Проверка	9
2.12	Проверка с другими значениями	10
3.1	Создание файла	11
3.2	Заполнение файла	11
3.3	Проверка	12
3.4	Проверка с другими значениями	12

Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

1. Создайте каталог для программ лабораторной работы № 8, перейдите в него и создайте файл lab8-1.asm

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

Рис. 2.1: Создание файла

```
GNU nano 6.2 /home/odisupov/work/arch-pc/lab08/lab8-1.asm
;-----
; Программа вывода значений регистра 'ecx'
;-----
#include 'in_out.asm'
SECTION .data
msg1 db 'Введите N: ',0h
SECTION .bss
N: resb 10
SECTION .text
global _start
_start:
; ----- Вывод сообщения 'Введите N: '
mov eax,msg1
call sprint
; ----- Ввод 'N'
mov ecx, N
mov edx, 10
call sread
; ----- Преобразование 'N' из символа в число
mov eax,N
```

Рис. 2.2: Заполнение файла

```

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

```

Рис. 2.3: Проверка

```

label:
sub ecx,1 ; `ecx=ecx-1`
mov [N],ecx
mov eax,[N]
call iprintLF ; Вывод значения `N`
loop label ; `ecx=ecx-1` и если `ecx` не '0'
; переход на `label`
call quit

```

Рис. 2.4: Изменение

```

odisupov@odisupov-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 10
9
7
5
3
1

```

Рис. 2.5: Проверка

```

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

```

Рис. 2.6: Изменение и проверка

2. Создайте файл lab8-2.asm в каталоге ~/work/arch-pc/lab08 и введите в него текст программы из листинга 8.2

```

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

```

Рис. 2.7: Создание файла

```

;-----
; Обработка аргументов командной строки
;-----
%include 'in_out.asm'
SECTION .text
global _start
_start:
pop ecx ; Извлекаем из стека в `ecx` количество
; аргументов (первое значение в стеке)
pop edx ; Извлекаем из стека в `edx` имя программы
; (второе значение в стеке)
sub ecx, 1 ; Уменьшаем `ecx` на 1 (количество
; аргументов без названия программы)
next:
cmp ecx, 0 ; проверяем, есть ли еще аргументы
jz _end ; если аргументов нет выходим из цикла
; (переход на метку `_end`)
pop eax ; иначе извлекаем аргумент из стека
call sprintf ; вызываем функцию печати
loop next ; переход к обработке следующего

```

Рис. 2.8: Заполнение файла


```

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

```

Рис. 2.9: Проверка

3. Создайте файл lab8-3.asm в каталоге ~/work/archpc/lab08 и введите в него текст программы из листинга 8.3

```

GNU nano 6.2 /home/odisupov/work/arch-pc/lab08/lab8-3.asm *
#include 'in_out.asm'
SECTION .data
msg db "Результат: ",0
SECTION .text
global _start
_start:
pop ecx; Извлекаем из стека в `ecx` количество
; аргументов (первое значение в стеке)
pop edx; Извлекаем из стека в `edx` имя программы
; (второе значение в стеке)
sub ecx,1; Уменьшаем `ecx` на 1 (количество
; аргументов без названия программы)
mov esi, 0; Используем `esi` для хранения
; промежуточных сумм
next:
cmp ecx,0h; проверяем, есть ли еще аргументы
jz _end; если аргументов нет выходим из цикла
; (переход на метку `_end`)
pop eax; иначе извлекаем следующий аргумент из стека
call atoi; преобразуем символ в число

```

Рис. 2.10: Создание и заполнение файла

```

odisupov@odisupov-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab08$ ./lab8-3 12 13 7 10 5
Результат: 47
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab08$

```

Рис. 2.11: Проверка

```
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab08$ ./lab8-3 3 6 5
Результат: 90
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab08$
```

Рис. 2.12: Проверка с другими значениями

3 Задания для самостоятельной работы

1. Создайте файл lab8-4.asm в каталоге ~/work/arch-pc/lab08

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

Рис. 3.1: Создание файла

```
GNU nano 6.2 /home/odisupov/work/arch-pc/lab08/lab8-4.asm
#include 'in_out.asm'
SECTION .data
msg db "Результат: ",0
SECTION .bss
ans: RESB 80
SECTION .text
global _start
_start:
pop ecx
pop edx
sub ecx,1
mov esi,10
next:
cmp ecx,0h
jz _end
pop eax
call atoi
mul esi
sub eax,5
add [ans],eax
```

Рис. 3.2: Заполнение файла

```
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-4.asm
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-4 lab8-4.o
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab08$ ./lab8-4 2 4 6
Результат: 105
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab08$
```

Рис. 3.3: Проверка

```
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab08$ ./lab8-4 1 3 5
Результат: 75
odisupov@odisupov-VirtualBox:~/work/arch-pc/lab08$
```

Рис. 3.4: Проверка с другими значениями

4 Выводы

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