

Отчёт по лабораторной работе №8

Дисциплина: Архитектура компьютера

Мария Данииловна Гольцова

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	17
	Список литературы	18

Список иллюстраций

2.1	Создание каталога и файла	6
2.2	Текст программы 1	7
2.3	Работа программы 1	7
2.4	Изменение в регистре	8
2.5	Работа программы 2	8
2.6	Изменения	9
2.7	Работа программы 3	9
2.8	Создание файла	10
2.9	Текст программы	10
2.10	Результат	10
2.11	Создание файла	11
2.12	Текст программы	11
2.13	Результат	11
2.14	Изменение программы для произведения	13
2.15	Работа программы	14
2.16	Создание файла	14
2.17	Программа варианта 1	15
2.18	Работа программы	16
2.19	Работа программы	16

Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

Создала каталог для программ лабораторной работы №8, перешла в него и создала файл lab8-1.asm (рис. 2.1).

```
mdgoljcova@dk5n55 ~ $ mkdir ~/work/arch-pc/lab08  
mdgoljcova@dk5n55 ~ $ cd ~/work/arch-pc/lab08  
mdgoljcova@dk5n55 ~/work/arch-pc/lab08 $ touch lab8-1.asm  
mdgoljcova@dk5n55 ~/work/arch-pc/lab08 $
```

Рис. 2.1: Создание каталога и файла

Ввела в файл lab8-1.asm текст программы из листинга 8.1 (рис. 2.2).

```

lab8-1.asm      [-M--] 24 L:[ 1+ 5 6/ 31] *(275 / 844b) 0010 0x00A
;-----
; 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
call atoi
mov [N],eax
; ----- Организация цикла
mov ecx,[N] ; Счетчик цикла, 'ecx=N'
label:
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. 2.2: Текст программы 1

Создала исполняемый файл и проверила его работу (рис. 2.3).

```

mdgoljcovadk5n55 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
mdgoljcovadk5n55 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
mdgoljcovadk5n55 ~/work/arch-pc/lab08 $ ./lab8-1
Введите N: 2
2
1
mdgoljcovadk5n55 ~/work/arch-pc/lab08 $

```

Рис. 2.3: Работа программы 1

Изменила текст программы, добавив изменение значение в регистра ecx в цикле (рис. 2.4).

```

lab8-1.asm      [-M--] 13 L: [ 1+28 29/ 31] *(742 / 763b) 0010
; -----
; 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
call atoi
mov [N],eax
; ----- Организация цикла
mov ecx,[N] ; Счетчик цикла, 'ecx=N'
label:
sub ecx,1 ; 'ecx=ecx-1'
mov [N],ecx
mov eax,[N]
call iprintLf
loop label
call quit

```

Рис. 2.4: Изменение в регистре

Создала исполняемый файл и проверила его работу (рис. 2.5).

```

mdgoljcova@dk5n55 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
mdgoljcova@dk5n55 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
mdgoljcova@dk5n55 ~/work/arch-pc/lab08 $ ./lab8-1
Введите N: 2
1
mdgoljcova@dk5n55 ~/work/arch-pc/lab08 $

```

Рис. 2.5: Работа программы 2

Внесла изменения в текст программы, добавив команды `push` и `pop` (добавления в стек и извлечения из стека) для сохранения значения счетчика цикла `loop` (рис. 2.6).


```

lab8-1.asm      [-M--] 10 L:[ 1+31 32/ 33] *(872 / 882b) 001
;-----
; 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
call atoi
mov [N],eax
; ----- Организация цикла
mov ecx,[N] ; Счетчик цикла, 'ecx=N'
label:
push ecx ; добавление значения ecx в стек
sub ecx,1
mov [N],ecx
mov eax,[N]
call iprintLF
pop ecx ; извлечение значения ecx из стека
loop label
call quit

```

Рис. 2.6: Изменения

Создала исполняемый файл и проверила его работу. Программа показывает все предыдущие числа до 0, не включая заданное N (рис. 2.7).

```

mdgoljcova@dk5n55 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
mdgoljcova@dk5n55 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
mdgoljcova@dk5n55 ~/work/arch-pc/lab08 $ ./lab8-1
Введите N: 2
1
0
mdgoljcova@dk5n55 ~/work/arch-pc/lab08 $

```

Рис. 2.7: Работа программы 3

Создала файл lab8-2.asm в каталоге ~/work/arch-pc/lab08 (рис. 2.8).

```
mdgoljcova@dk5n55 ~/work/arch-pc/lab08 $ touch lab8-2.asm
mdgoljcova@dk5n55 ~/work/arch-pc/lab08 $
```

Рис. 2.8: Создание файла

Ввела в него текст программы из листинга 8.2 (рис. 2.9).

```
lab8-2.asm      [-M--] 66 L:[ 1+ 2 3/ 23] *(207 /1151b) 0010
;
; -----
; 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
; аргумента (переход на метку 'next')
_end:
call quit
```

Рис. 2.9: Текст программы

Создала исполняемый файл и запустила его, указав аргументы (рис. 2.10).

```
mdgoljcova@dk5n55 ~/work/arch-pc/lab08 $ nasm -f elf lab8-2.asm
mdgoljcova@dk5n55 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-2 lab8-2.o
mdgoljcova@dk5n55 ~/work/arch-pc/lab08 $ ./lab8-2 аргумент1 аргумент 2 'аргумент 3'
аргумент1
аргумент
2
аргумент 3
mdgoljcova@dk5n55 ~/work/arch-pc/lab08 $
```

Рис. 2.10: Результат

Создала файл lab8-3.asm в каталоге ~/work/arch-pc/lab08 (рис. 2.11).

```
mdgoljcov@dk5n55 ~/work/arch-pc/lab08 $ touch lab8-3.asm
mdgoljcov@dk5n55 ~/work/arch-pc/lab08 $
```

Рис. 2.11: Создание файла

Ввела в него текст программы из листинга 8.3 (рис. 2.12).

```
lab8-3.asm [-M--] 7 L: [ 1+ 5 6/ 29] *(103 /1428b) 00
#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
add esi,eax ; добавляем к промежуточной сумме
; след. аргумент 'esi=esi+eax'
loop next ; переход к обработке следующего аргумента
_end:
mov eax, msg ; вывод сообщения "Результат: "
call sprint
mov eax, esi ; записываем сумму в регистр 'eax'
call iprintLF ; печать результата
call quit ; завершение программы
```

Рис. 2.12: Текст программы

Создала исполняемый файл и запустила его, указав аргументы (рис. 2.13).

```
mdgoljcov@dk5n55 ~/work/arch-pc/lab08 $ nasm -f elf lab8-3.asm
mdgoljcov@dk5n55 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-3 lab8-3.o
mdgoljcov@dk5n55 ~/work/arch-pc/lab08 $ ./lab8-3 12 13 7 10 5
Результат: 47
mdgoljcov@dk5n55 ~/work/arch-pc/lab08 $
```

Рис. 2.13: Результат

Изменила текст программы из листинга 8.3 для вычисления произведения аргументов командной строки (рис. 2.14).

```
lab8-3.asm      [-M--] 8
%include 'in_out.asm'
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. 2.14: Изменение программы для произведения

Создала исполняемый файл и запустила его, указав аргументы (рис. 2.15).

```
mdgoljcova@dk5n55 ~/work/arch-pc/lab08 $ nasm -f elf lab8-3.asm
mdgoljcova@dk5n55 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-3 lab8-3.o
mdgoljcova@dk5n55 ~/work/arch-pc/lab08 $ ./lab8-3 2 11 7 10 5
Результат: 7700
mdgoljcova@dk5n55 ~/work/arch-pc/lab08 $
```

Рис. 2.15: Работа программы

#Задание для самостоятельной работы

Создала файл lab8-4.asm в каталоге ~/work/arch-pc/lab08 (рис. 2.16).

```
mdgoljcova@dk4n62 ~ $ cd ~/work/arch-pc/lab08
mdgoljcova@dk4n62 ~/work/arch-pc/lab08 $ touch lab8-4.asm
mdgoljcova@dk4n62 ~/work/arch-pc/lab08 $
```

Рис. 2.16: Создание файла

Напишите программу в соответствии со своим вариантом 1, которая находит сумму значений функции $f(x)$ для $x = x_1, x_2, \dots, x_n$, т.е. программа должна выводить значение $f(x_1) + f(x_2) + \dots + f(x_n)$ (рис. 2.17).

```

lab8-4.asm      [-----]  7  L:[ 1+
#include 'in_out.asm'
SECTION .data
msg db "Ответ: ",0
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
mov ebx,2
mul ebx
add eax,15
add esi,eax
loop next
_end:
mov eax, msg
call sprint
mov eax, esi
call iprintLF
call quit

```

Рис. 2.17: Программа варианта 1

Создала исполняемый файл и проверила его работу на аргументах 1 2 3 4 (рис. 2.18).

```
mdgoljcova@dk4n62 ~/work/arch-pc/lab08 $ nasm -f elf lab8-4.asm
mdgoljcova@dk4n62 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-4 lab8-4.o
mdgoljcova@dk4n62 ~/work/arch-pc/lab08 $ ./lab8-4 1 2 3 4
Ответ: 80
mdgoljcova@dk4n62 ~/work/arch-pc/lab08 $
```

Рис. 2.18: Работа программы

Проверила его работу на аргументах 7 10 23 3 (рис. 2.19).

```
mdgoljcova@dk4n62 ~/work/arch-pc/lab08 $ ./lab8-4 7 10 23 3
Ответ: 146
mdgoljcova@dk4n62 ~/work/arch-pc/lab08 $
```

Рис. 2.19: Работа программы

3 Выводы

В ходе работы я приобрела навыки написания программ с использованием циклов и обработкой аргументов командной строки.

Список литературы