

Отчёт по лабораторной работе №7

Дисциплина: Архитектура компьютера

Арсений Андреевич Шалин

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выполнение самостоятельной работы	10
4	Выводы	15
	Список литературы	16

Список иллюстраций

2.1	Задание 7.3.1.1	6
2.2	Задание 7.3.1.2А	6
2.3	Задание 7.3.1.2Б	6
2.4	Задание 7.3.1.2В	7
2.5	Задание 7.3.1.2Г	7
2.6	Задание 7.3.1.2Д	7
2.7	Задание 7.3.1.2Е	8
2.8	Задание 7.3.1.3А	8
2.9	Задание 7.3.2А	9
2.10	Задание 7.3.2Б	9
2.11	Задание 7.3.2В	9
3.1	Задание 7.4.1	10
3.2	Задание 7.4.2	10

Список таблиц

1 Цель работы

Изучение команд условного и безусловного переходов. Приобретение навыков написания программ с использованием переходов. Знакомство с назначением и структурой файла листинга.

2 Выполнение лабораторной работы

Создал файл lab7-1.asm в каталоге для программ лабораторной работы №7 (рис. 2.1).

```
[aashalin@aashalin lab07]$ touch lab7-1.asm
[aashalin@aashalin lab07]$ ls
lab7-1.asm
```

Рис. 2.1: Задание 7.3.1.1

Скопировал программу из листинга 7.1 в lab7-1.asm (рис. 2.2).

```
%include 'in_out.asm' ; подключение внешнего файла

SECTION .data
msg1: DB 'Сообщение № 1',0
msg2: DB 'Сообщение № 2',0
msg3: DB 'Сообщение № 3',0

SECTION .text
GLOBAL _start
_start:

jmp _label2

_label1:
mov eax, msg1 ; Вывод на экран строки
call sprintf ; 'Сообщение № 1'

_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintf ; 'Сообщение № 2'

_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintf ; 'Сообщение № 3'

_end:
call quit ; вызов подпрограммы завершения
```

Рис. 2.2: Задание 7.3.1.2А

Создал и запустил исполняемый файл, убедился, что результат соответствует результату в задании (рис. 2.3).

```
[aashalin@aashalin lab07]$ ./lab7-1
Сообщение № 2
Сообщение № 3
```

Рис. 2.3: Задание 7.3.1.2Б

Скопировал программу из листинга 7.2 в lab7-1.asm (рис. 2.4).

```
jmp _label2

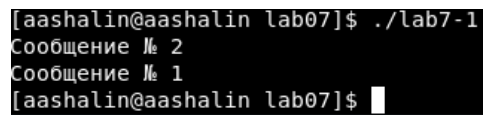
_label1:
mov eax, msg1 ; Вывод на экран строки
call sprintf ; 'Сообщение № 1'
jmp _end

_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintf ; 'Сообщение № 2'
jmp _label1

_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintf ; 'Сообщение № 3'
```

Рис. 2.4: Задание 7.3.1.2В

Создал исполняемый файл и проверил его работу (рис. 2.5).



```
[aashalin@aashalin lab07]$ ./lab7-1
Сообщение № 2
Сообщение № 1
[aashalin@aashalin lab07]$
```

Рис. 2.5: Задание 7.3.1.2Г

Модифицировал lab7-1.asm так, чтобы он выдал сообщение №3, затем №2, затем №1 (рис. 2.6).

```
jmp _label3

_label1:
mov eax, msg1 ; Вывод на экран строки
call sprintf ; 'Сообщение № 1'
jmp _end

_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintf ; 'Сообщение № 2'
jmp _label1

_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintf ; 'Сообщение № 3'
jmp _label2
```

Рис. 2.6: Задание 7.3.1.2Д

Создал исполняемый файл и проверил его работу (рис. 2.7).

```
[aashalin@aashalin lab07]$ ./lab7-1
Сообщение № 3
Сообщение № 2
Сообщение № 1
[aashalin@aashalin lab07]$
```

Рис. 2.7: Задание 7.3.1.2Е

Скопировал программу из листинга 7.3 в lab7-2.asm (рис. 2.8).

```
%include 'in_out.asm'
section .data
msg1 db 'Введите B: ',0h
msg2 db "Наибольшее число: ",0h
A dd '20'
C dd '50'
section .bss
max resb 10
B resb 10
section .text
global _start
_start:
; ----- Вывод сообщения 'Введите B: '
mov eax,msg1
call sprint
; ----- Ввод 'B'
mov ecx,B
mov edx,10
call sread
; ----- Преобразование 'B' из символа в число
mov eax,B
call atoi ; Вызов подпрограммы перевода символа в число
mov [B],eax ; запись преобразованного числа в 'B'
; ----- Записываем 'A' в переменную 'max'
mov ecx,[A] ; 'ecx = A'
mov [max],ecx ; 'max = A'
; ----- Сравниваем 'A' и 'C' (как символы)
cmp ecx,[C] ; Сравниваем 'A' и 'C'
jg check_B ; если 'A>C', то переход на метку 'check_B',
mov ecx,[C] ; иначе 'ecx = C'
mov [max],ecx ; 'max = C'
; ----- Преобразование 'max(A,C)' из символа в число
check_B:
mov eax,max
call atoi ; Вызов подпрограммы перевода символа в число
mov [max],eax ; запись преобразованного числа в 'max'
; ----- Сравниваем 'max(A,C)' и 'B' (как числа)
mov ecx,[max]
cmp ecx,[B] ; Сравниваем 'max(A,C)' и 'B'
jg fin ; если 'max(A,C)>B', то переход на 'fin',
mov ecx,[B] ; иначе 'ecx = B'
mov [max],ecx
; ----- Вывод результата
fin:
mov eax, msg2
call sprint ; Вывод сообщения 'Наибольшее число: '
mov eax,[max]
call iprintLF ; Вывод 'max(A,B,C)'
call quit ; Выход
```

Рис. 2.8: Задание 7.3.1.3А

Создал листинг lab7-2.lst (рис. 2.9).


```
[aashalin@aashalin lab07]$ nasm -f elf -l lab7-2.lst lab7-2.asm
lab7-2.asm:21: error: invalid combination of opcode and operands
[aashalin@aashalin lab07]$ ls -l
total 40
-rw-rw-r--. 1 aashalin aashalin 3942 Nov 21 12:18 in_out.asm
-rwxrwxr-x. 1 aashalin aashalin 1444 Nov 21 12:25 lab7-1
-rw-rw-r--. 1 aashalin aashalin 724 Nov 21 12:25 lab7-1.asm
-rw-rw-r--. 1 aashalin aashalin 1456 Nov 21 12:25 lab7-1.o
-rwxrwxr-x. 1 aashalin aashalin 1596 Nov 21 12:33 lab7-2
-rw-rw-r--. 1 aashalin aashalin 1776 Nov 23 16:39 lab7-2.asm
-rw-rw-r--. 1 aashalin aashalin 14581 Nov 23 16:40 lab7-2.lst
drwxrwxr-x. 5 aashalin aashalin 114 Nov 21 12:17 report
[aashalin@aashalin lab07]$
```

Рис. 2.9: Задание 7.3.2А

В этих трёх строках сначала еах перемещён в еsi, затем 0 в еах, затем 0 в есх.

```
132 000000A0 89C6          <1>      mov     esi, eax
133 000000A2 B800000000      <1>      mov     eax, 0
134 000000A7 B900000000      <1>      mov     ecx, 0
```

Удалил один элемент (рис. 2.10).

```
; ----- Ввод 'B'
mov ecx,B
mov edx,10
call sread
; ----- Преобразование 'B' из символа в число
mov eax|
```

Рис. 2.10: Задание 7.3.2Б

lab7-2.asm не создаётся, в листинг идёт ошибка (рис. 2.11).

```
195                                     ; ----- Преобразование 'B' из символа в число
196                                     mov eax
197 ***** error: invalid combination of opcode and operands
```

Рис. 2.11: Задание 7.3.2В

3 Выполнение самостоятельной работы

После написания программы для нахождения наименьшей из трёх переменных a , b и c в текстовом файле lab7-3.asm (вариант 12: 99,29,26, Листинг 1) создал исполняемый файл lab7-3 и проверил его работу (рис. 3.1).

```
[aashalin@aashalin lab07]$ nasm -f elf lab7-3.asm
[aashalin@aashalin lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[aashalin@aashalin lab07]$ ./lab7-3
Наименьшее число: 26
[aashalin@aashalin lab07]$
```

Рис. 3.1: Задание 7.4.1

После написания программы для вычисления значения заданной функции $f(x)$ в текстовом файле lab7-4.asm (вариант 12: ax , $x < 5$ или $x - 5$, $x \geq 5$, Листинг 2) создал исполняемый файл lab7-4 и проверил его работу (рис. 3.2).

```
[aashalin@aashalin lab07]$ ./lab7-4
Введите x: 3
Введите a: 7
Результат: 21
[aashalin@aashalin lab07]$ ./lab7-4
Введите x: 6
Введите a: 4
Результат: 1
[aashalin@aashalin lab07]$
```

Рис. 3.2: Задание 7.4.2

Листинг 1., lab7-3.asm

```
%include 'in_out.asm'
section .data
```

```

msg2 db "Наименьшее число: ",0h
A dd '99'
B dd '29'
C dd '26'
section .bss
min resb 10
; B resb 10
section .text
global _start
_start:
; ----- Вывод сообщения 'Введите B: '
; mov eax,msg1
; call sprint
; ----- Ввод 'B'
; mov ecx,B
; mov edx,10
; call sread
; ----- Преобразование 'B' из символа в число
mov eax,B
call atoi ; Вызов подпрограммы перевода символа в число
mov [B],eax ; запись преобразованного числа в 'B'
; ----- Записываем 'A' в переменную 'min'
mov ecx,[A] ; 'ecx = A'
mov [min],ecx ; 'min = A'
; ----- Сравниваем 'A' и 'C' (как символы)
cmp ecx,[C] ; Сравниваем 'A' и 'C'
jnl check_B ; если 'A<C', то переход на метку 'check_B',
mov ecx,[C] ; иначе 'ecx = C'
mov [min],ecx ; 'min = C'

```

```

; ----- Преобразование 'min(A,C)' из символа в число
check_B:
    mov eax,min
    call atoi ; Вызов подпрограммы перевода символа в число
    mov [min],eax ; запись преобразованного числа в 'min'
; ----- Сравниваем 'min(A,C)' и 'B' (как числа)
    mov ecx,[min]
    cmp ecx,[B] ; Сравниваем 'min(A,C)' и 'B'
    jl fin ; если 'min(A,C)<B', то переход на 'fin',
    mov ecx,[B] ; иначе 'ecx = B'
    mov [min],ecx
; ----- Вывод результата
fin:
    mov eax, msg2
    call sprint ; Вывод сообщения 'Наибольшее число: '
    mov eax,[min]
    call iprintLF ; Вывод 'min(A,B,C)'
    call quit ; Выход

```

Листинг 2., lab7-4.asm

```

%include 'in_out.asm'
section .data
    msgx db 'Введите x: ',0h
    msga db 'Введите a: ',0h
    msgr db "Результат: ",0h
section .bss
    res resb 10
    X resb 10
    A resb 10
section .text

```

```

global _start
_start:
; ----- Вывод сообщения 'Введите x: '
mov eax,msgx
call sprint
; ----- Ввод 'x'
mov ecx,X
mov edx,10
call sread
; ----- Преобразование 'x' из символа в число
mov eax,X
call atoi ; Вызов подпрограммы перевода символа в число
mov [X],eax ; запись преобразованного числа в 'x'

; ----- Вывод сообщения 'Введите a: '
mov eax,msga
call sprint
; ----- Ввод 'a'
mov ecx,A
mov edx,10
call sread
; ----- Преобразование 'a' из символа в число
mov eax,A
call atoi ; Вызов подпрограммы перевода символа в число
mov [A],eax ; запись преобразованного числа в 'a'

mov ebx,[X]

```

```
cmp ebx,5 ; Сравниваем 'x' и 5
jge check_X ; если 'x≥5', то переход на метку 'check_X',
```

```
mov eax,[X]
mov ebx,[A]
mul ebx
mov [res],eax
jmp fin
```

check_X:

```
mov eax,[X]
sub eax,5
mov [res],eax
```

; ----- Вывод результата

fin:

```
mov eax, msgr
call sprint ; Вывод сообщения 'Результат: '
mov eax,[res]
call iprintLF ; Вывод 'res'
call quit ; Выход
```

Ссылка на отчёт лабораторной №7, сделанный в Markdown. https://github.com/arsenyshalin/study_2024-2025_arh-pc/tree/master/labs/lab07/report

4 Выводы

- Изучены команды условного и безусловного переходов.
- Приобретены навыки написания программ с использованием переходов
- С назначением и структурой файла листинга ознакомлен.
- Создана программа для вычисления выражения из вариантов, полученных из номера студенческого билета в лабораторной №6.

Список литературы