

Отчет по лабораторной работе №7

Дисциплина: архитектура компьютера

Провоторов Антон Григорьевич

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	7
4	Выполнение заданий для самостоятельной работы	10
5	Выводы	13

Список иллюстраций

3.1	Создание каталога lab07 и файла lab7-1.asm	7
3.2	Ввод в файл текст программы из листинга	7
3.3	Создание исполняемого файла и его запуск.Результ совпадает с примером	7
3.4	Изменение текста программы в соответствие с текстом листинга . .	8
3.5	Ввод текста программы из листинга	8
3.6	Создание исполняемого файла и его запуск	9
3.7	Создание файла листинга для программы из файла lab7-2.asm . .	9
3.8	В файле lab7-2.asm убрал один из двух операндов в инструкции с двумя операндами	9
3.9	Выполнил трансляцию с получением файла листинга.(Файл изме- нился)	9
3.10	Изменения в файле листинга после убирание одного из операндов в инструкции с двумя перандами	9
4.1	Создание файла lab7-3.asm	10
4.2	Написание текста программы	11
4.3	Создание исполняемого фАйла и его запуск	11
4.4	Создание файла lab7-4.asm	11
4.5	Написание текста программы	12
4.6	Создание исполняемого файла и его запуск со значениями x1,a1 .	12
4.7	Запуск со значениями x2,a2	12

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 Теоретическое введение

Для реализации ветвлений в ассемблере используются так называемые команды передачи управления или команды перехода. Можно выделить 2 типа переходов: • условный переход – выполнение или не выполнение перехода в определенную точку программы в зависимости от проверки условия. • безусловный переход – выполнение передачи управления в определенную точку программы без каких-либо условий.

3 Выполнение лабораторной работы

```
agprovotorov@dk6n55 ~ $ cd /afs/.dk.sci.pfu.edu.ru/home/a/g/agprovotorov/work/arch-ps
agprovotorov@dk6n55 ~/work/arch-ps $ mkdir lab07
agprovotorov@dk6n55 ~/work/arch-ps $ cd ~/work/arch-ps/lab07
bash: cd: /afs/.dk.sci.pfu.edu.ru/home/a/g/agprovotorov/work/arch-ps/lab07: Нет такого файла или каталога
agprovotorov@dk6n55 ~/work/arch-ps $ cd /afs/.dk.sci.pfu.edu.ru/home/a/g/agprovotorov/work/arch-ps/lab07
agprovotorov@dk6n55 ~/work/arch-ps/lab07 $ touch lab7-1.asm
agprovotorov@dk6n55 ~/work/arch-ps/lab07 $
```

Рис. 3.1: Создание каталога lab07 и файла lab7-1.asm

```
GNU nano 6.4 /afs/.dk.sci.pfu.edu.ru/home/a/g/agprovotorov/work/arch-ps/lab07/lab7-1.asm
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data
msg1: DB 'Сообщение No 1',0
msg2: DB 'Сообщение No 2',0
msg3: DB 'Сообщение No 3',0
SECTION .text
GLOBAL _start
_start:
jmp _label2
_label1:
mov eax, msg1 ; Вывод на экран строки
call sprintf ; 'Сообщение No 1'
_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintf ; 'Сообщение No 2'
_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintf ; 'Сообщение No 3'
_end:
call quit ; вызов подпрограммы завершения
```

Рис. 3.2: Ввод в файл текст программы из листинга

```
agprovotorov@dk6n55 ~/work/arch-ps/lab07 $ nasm -f elf lab7-1.asm
agprovotorov@dk6n55 ~/work/arch-ps/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
agprovotorov@dk6n55 ~/work/arch-ps/lab07 $ ./lab7-1
Сообщение No 2
Сообщение No 3
```

Рис. 3.3: Создание исполняемого файла и его запуск. Результат совпадает с примером

```

#include 'in_out.asm' ; подключение внешнего файла
SECTION .data
msg1: DB 'Сообщение No 1',0
msg2: DB 'Сообщение No 2',0
msg3: DB 'Сообщение No 3',0
SECTION .text
GLOBAL _start
_start:
jmp _label2
_label1:
mov eax, msg1 ; Вывод на экран строки
call sprintf ; 'Сообщение No 1'
jmp _end
_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintf ; 'Сообщение No 2'
jmp _label1
_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintf ; 'Сообщение No 3'
_end:
call quit ; вызов подпрограммы завершения

```

Рис. 3.4: Изменение текста программы в соответствии с текстом листинга

```

agprovotorov@dk6n55 ~/work/arch-ps/lab07 $ nasm -f elf lab7-1.asm
agprovotorov@dk6n55 ~/work/arch-ps/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
agprovotorov@dk6n55 ~/work/arch-ps/lab07 $ ./lab7-1
Сообщение No 2
Сообщение No 1

```

```

GNU nano 6.4 /afs/dk.sci.pfu.edu.ru/home/a/g/agprovotorov/work/arch-ps/lab07/lab7-2.asm
#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 sprintf
; ----- Ввод 'B'
mov ecx,B
mov edx,10
call read
; ----- 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
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'

```

Рис. 3.5: Ввод текста программы из листинга


```
agprovotorov@dk6n55 ~/work/arch-ps/lab07 $ nasm -f elf lab7-2.asm
agprovotorov@dk6n55 ~/work/arch-ps/lab07 $ ld -m elf_i386 -o lab7-2 lab7-2.o
agprovotorov@dk6n55 ~/work/arch-ps/lab07 $ ./lab7-2
Введите B: 1
Наибольшее число: 50
```

Рис. 3.6: Создание исполняемого файла и его запуск

```
agprovotorov@dk8n74 ~/work/arch-ps/lab07 $ nasm -f elf -l lab7-2.lst lab7-2.asm
```

Рис. 3.7: Создание файла листинга для программы из файла lab7-2.asm

```
check_B:
mov eax,
call atoi ; Вызов подпрограммы перевода символа в число
mov [max],eax ; запись преобразованного числа в 'max'
; ***** Сравниваем 'max(A,C)' и 'B' (как числа)
```

Рис. 3.8: В файле lab7-2.asm убрал один из двух операндов в инструкции с двумя операндами

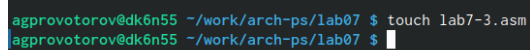
```
agprovotorov@dk6n55 ~/work/arch-ps/lab07 $ nasm -f elf -l lab7-2.lst lab7-2.asm
lab7-2.asm:23: error: invalid combination of opcode and operands
```

Рис. 3.9: Выполнил трансляцию с получением файла листинга.(Файл изменился)

```
22 00000106 E891FFFFFF call atoi ; Вызов подпрограммы перевода символа в число
23 mov EBX ; запись преобразованного числа в 'B'
23 ***** error: invalid combination of opcode and operands
24 ; ----- Записываем 'A' в переменную 'max'
```

Рис. 3.10: Изменения в файле листинга после убирание одного из операндов в инструкции с двумя перандами

4 Выполнение заданий для самостоятельной работы

A terminal window with a dark background. The prompt is 'agprovotorov@dk6n55 ~/work/arch-ps/lab07 \$'. The command 'touch lab7-3.asm' has been entered and executed. The next line shows the prompt 'agprovotorov@dk6n55 ~/work/arch-ps/lab07 \$' with a cursor.

```
agprovotorov@dk6n55 ~/work/arch-ps/lab07 $ touch lab7-3.asm
agprovotorov@dk6n55 ~/work/arch-ps/lab07 $
```

Рис. 4.1: Создание файла lab7-3.asm

```

lab7-3.asm [----] 9 L: [ 1+38 39/ 39] *(470 / 470b) <EOF>
#include "lab7-3.asm"
section .data
msg2 db "Наименьшее число: ",0h
A dd 40
B dd 17
C dd 23
section .bss
min resb 10
section .text
global _start
_start:

mov eax, B
call atoi
mov [B],eax
mov ecx, [A]
mov [min], ecx
cmp ecx, [C]

jl check_B ,
mov ecx, [C]
mov [min],ecx
check_B:
mov eax,min
call atoi
mov [min],eax

mov ecx,[min]
cmp ecx,[B]
jl fin
mov ecx,[B]
mov [min],ecx

fin:
mov eax, msg2
call sprint
mov eax,[min]
call iprintLF
call quit

```

Рис. 4.2: Написание текста программы

```

agprovotorov@dk8n74 ~/work/arch-ps/lab07 $ nasm -f elf lab7-3.asm
agprovotorov@dk8n74 ~/work/arch-ps/lab07 $ ld -m elf_i386 -o lab7-3 lab7-3.o
agprovotorov@dk8n74 ~/work/arch-ps/lab07 $ ./lab7-3
Наименьшее число: 17

```

Рис. 4.3: Создание исполняемого файла и его запуск

```

agprovotorov@dk8n74 ~/work/arch-ps/lab07 $ touch lab7-4.asm

```

Рис. 4.4: Создание файла lab7-4.asm

```

lab7-4.asm      [----]  0 L: [ 21+30  51/ 62] *(743 / 909b) 0032 0x020
mov  eax,msgx
call sprint
mov  ecx,X
mov  edx,10
call sread
mov  eax,X
call atoi
mov  [X],eax

mov  eax,msga
call sprint
mov  ecx,A
mov  edx,10
call sread
mov  eax,A
call atoi
mov  [A],eax

mov  ecx,[A] ; 'ecx = A'
mov  [max],ecx ; 'max = A'

cmp  ecx,[X] ; Сравниваем 'A' и 'C'
jg   _label3
jmp  _label2

_label2:
mov  eax, msg2 ; Вывод на экран строки
call sprintf ; %s
jmp  _end
_label3:
mov  edx,2
mov  eax, [A]
mul  edx
mov  ecx,[X]
neg  ecx
add  eax,ecx

call sprintf
_end:
call quit ; вызов подпрограммы завершения

```

Рис. 4.5: Написание текста программы

```

agrovotorov@dk8n74 ~/work/arch-ps/lab07 $ nasm -f elf lab7-4.asm
agrovotorov@dk8n74 ~/work/arch-ps/lab07 $ ld -m elf_i386 -o lab7-4 lab7-4.o
agrovotorov@dk8n74 ~/work/arch-ps/lab07 $ ./lab7-4
Введите X:1
Введите A:2
Ошибка сегментирования (стек памяти сброшен на диск)

```

Рис. 4.6: Создание исполняемого файла и его запуск со значениями x1,a1

```

agrovotorov@dk8n74 ~/work/arch-ps/lab07 $ ./lab7-4
Введите X:2
Введите A:1
8

```

Рис. 4.7: Запуск со значениями x2,a2

5 Выводы

Изученил команды условного и безусловного переходов. Приобретенил навыки написания программ с использованием переходов.Познакомился с назначением и структурой файла листинга.