РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 7

дисциплина: Архитектура компьютера

Студент:

Мартемьянов Максим Сергеевич

Студ.билет:

1032250312

Группа:

НКАбд-04-25

МОСКВА

2025г.

Цель работы

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

Выполнение лабораторной работы

Создайте каталог для программам лабораторной работы № 7, перейдите в него и создайте файл lab7-1.asm.

Создал каталог для программам лабораторной работы № 7, перешёл в него и создал файл lab7-1.asm

```
msmartemyanov@msmartemyanov:~$ mkdir ~/work/arch-pc/lab07
msmartemyanov@msmartemyanov:~$ cd ~/work/arch-pc/lab07
msmartemyanov@msmartemyanov:~/work/arch-pc/lab07$ touch lab7-1.asm
msmartemyanov@msmartemyanov:~/work/arch-pc/lab07$
```

Введите в файл lab7-1.asm текст программы из листинга 7.1.

Ввел в файл lab7-1.asm текст программы из листинга 7.1 и выполнил все действия.

```
%include 'in_out.asm'

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
    call sprintLF

_label2:
    mov eax, msg2
    call sprintLF

_label3:
    mov eax, msg3
    call sprintLF

_end:
    call quit
```

Создайте исполняемый файл и запустите его.

Создал и запусти файл.

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

Исправид файл и запустил исправленный код.

```
%include 'in_out.asm'

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

section .text
global _start
_start:

jmp _label3

_label1:
    mov eax, msg1
    call sprintLF; 'Сообщение №1'
    jmp _end
_label2:
    mov eax, msg2;
    call sprintLF; 'Сообщение №2'
    jmp _label1
_label3:
    mov eax, msg3
    call sprintLF; 'Сообщение №3'
    jmp _label2
_end:
    call quit
```

Исправленный код.

```
msmartemyanov@msmartemyanov:~/work/arch-pc/lab07$ nasm -f elf lab7-1.asm
msmartemyanov@msmartemyanov:~/work/arch-pc/lab07$ cd ~/work/arch-pc/lab07
msmartemyanov@msmartemyanov:~/work/arch-pc/lab07$ ./lab7-1
Сообщение № 3
Сообщение № 2
Сообщение № 1
msmartemyanov@msmartemyanov:~/work/arch-pc/lab07$
```

Создайте файл lab7-2.asm в каталоге ~/work/arch-pc/lab07 и введите в него текст программы из листинга 7.3. touch ~/work/arch-pc/lab07/lab7-2.asm.

Создал файл lab7-2.asm в каталоге ~/work/arch-pc/lab0 и ввёл в него текст программы из листинга 7.3.

```
msmartemyanov@msmartemyanov:~/work/arch-pc/lab07$ touch lab7-2.asm
msmartemyanov@msmartemyanov:~/work/arch-pc/lab07$
```

```
%include 'in_out.asm'
Файлы
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'
```

Создал и запустил данный файл.

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

Изучение структуры файлы листинга

Создайте файл листинга для программы из файла lab7-2.asm. Откройте файл листинга lab7-2.lst с помощью любого текстового редактора .

Создал файл и ввел в него тест из листинга. Открыл файл листинга lab7-2.lst с помощью текстового редактора.

```
msmartemyanov@msmartemyanov:~/work/arch-pc/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
msmartemyanov@msmartemyanov:~/work/arch-pc/lab07$ nano lab7-2.lst
msmartemyanov@msmartemyanov:~/work/arch-pc/lab07$
```

```
 GNU nano 7.2                                        lab7-2.lst
   27 00000105 E881FFFFFF                call atoi
   28 0000010A A3[0A000000]              mov [B], eax
   29
   30                                  ; --- Записываем 'A' в переменную 'max'
   31 0000010F 8B0D[35000000]            mov ecx, [A]
   32 00000115 890D[00000000]            mov [max], ecx
   33
   34                                  ; --- Сравниваем 'A' и 'C' (как символы)
   35 0000011B 3B0D[39000000]            cmp ecx, [C]
   36 00000121 7F0C                      jg check_B
   37 00000123 8B0D[39000000]            mov ecx, [C]
   38 00000129 890D[00000000]            mov [max], ecx
   39
   40                                  ; --- Преобразование 'max(A,C)' из символа в число
   41                                  check_B:
   42 0000012F B8[00000000]              mov eax, max
   43 00000134 E852FFFFFF                call atoi
   44 00000139 A3[00000000]              mov [max], eax
   45
   46                                  ; --- Сравниваем 'max(A,C)' и 'B' (как числа)
   47 0000013E 8B0D[00000000]            mov ecx, [max]
   48 00000144 3B0D[0A000000]            cmp ecx, [B]
   49 0000014A 7F0C                      jg fin
   50 0000014C 8B0D[0A000000]            mov ecx, [B]
   51 00000152 890D[00000000]            mov [max], ecx
   52
   53                                  ; --- Вывод результата
   54                                  fin:
```

Анализ:

1)32 00000115 890D[00000000] mov [max],ecx

Номер строки: 32

Адрес : 00000115

Машинный код: 890D[00000000]

890D – код операции mov

[00000000] – адрес переменно max

Исходный текст: mov[max],ecx

2)50 0000014C 8B0D[0A000000] mov ecx,[B]

Номер строки: 50

Адрес : 0000014C

Машинный код: 8B0D[0A000000]

8B0D – код операции mov

[0A000000] – адрес переменно max

Исходный текст: mov ecx[B]

Если допустить ошибку , то код не будет работать. При наличии ошибок транслятор не создаёт объектный файл (.o) Файл листинга либо не создаётся, либо остаётся от предыдущей успешной компиляции. Все ошибки выводятся только на экран.

Задания для самостоятельной работы.

Напишите программу нахождения наименьшей из 3 целочисленных переменных a,b и . Значения переменных выбрать из табл. 7.5 в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу. Вариант 15

```
%include 'in_out.asm'

SECTION .data
    msg db "Наименьшее число: ",0
    a dd 32
    b dd 6
    c dd 54

SECTION .bss
    min resd 1

SECTION .text
    global _start

_start:
    ; Записываем a в min
    mov eax, [a]
    mov [min], eax

    ; Сравниваем min и b
    mov ebx, [b]
    cmp eax, ebx
    jle check_c
    mov [min], ebx

check_c:
    ; Сравниваем min и c
    mov eax, [min]
```

```
msmartemyanov@msmartemyanov:~/work/arch-pc/lab07$ nasm -f elf sam115.asm
msmartemyanov@msmartemyanov:~/work/arch-pc/lab07$ ld -m elf_i386 sam115.o -o sam115
msmartemyanov@msmartemyanov:~/work/arch-pc/lab07$ ./sam115
Наименьшее число: 6
msmartemyanov@msmartemyanov:~/work/arch-pc/lab07$
```

Напишите программу, которая для введенных с клавиатуры значений x и a вычисляет значение заданной функции $f(x)$ и выводит результат вычислений. Вид функции $f(x)$ выбрать из таблицы 7.6 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу для значений x и a из 7.6.Вариант 15

```
%include 'in_out.asm'

SECTION .data
    msg_x db "Введите x: ",0
    msg_a db "Введите a: ",0
    res db "Результат: ",0

SECTION .bss
    x resd 1
    a resd 1

SECTION .text
    global _start

_start:
    ; Ввод x
    mov eax, msg_x
    call sprint
    mov ecx, x
    mov edx, 10
    call sread
    mov eax, x
    call atoi
    mov [x], eax

    ; Ввод a
    mov eax, msg_a
    call sprint
```

```
msmartemyanov@msmartemyanov:~/work/arch-pc/lab07$ nasm -f elf sam215.asm
msmartemyanov@msmartemyanov:~/work/arch-pc/lab07$ ld -m elf_i386 sam215.o -o sam215
msmartemyanov@msmartemyanov:~/work/arch-pc/lab07$ ./sam215
Введите x: 2
Введите a: 3
Результат: 13
msmartemyanov@msmartemyanov:~/work/arch-pc/lab07$ ./sam215
Введите x: 4
Введите a: 2
Результат: 14
msmartemyanov@msmartemyanov:~/work/arch-pc/lab07$
```

Вопросы для самопроверки

1. Нужен для отладки - показывает соответствие исходного кода машинному коду, адреса инструкций.

Отличается от исходника наличием машинного кода и адресов.

2. Состоит из: номера строки, адреса в памяти, машинного кода, исходного текста.

3. Через команды переходов:

1) Безусловные (jmp)

2)Условные (je, jg, jl и др.)

4. Безусловные:

jmp

Условные:

je/jne (равно/не равно),

jg/jl (больше/меньше со знаком),

ja/jb (выше/ниже без знака)

5. Команда cmp. cmp op1, op2 - вычисляет op1 - op2, не сохраняет результат, только устанавливает флаги.

6. Синтаксис условных переходов j метка Пример: je label, jg label

7. cmp eax, ebx jl less_label ; если eax < ebx jg greater_label ; если eax > ebx

8.

ZF (ноль) - равенство

CF (перенос) - беззнаковое сравнение

SF (знак) + OF (переполнение) - знаковое сравнение

PF (чётность) - проверка чётности

Вывод

В ходе проведенной лабораторной работы изучил команды условного и безусловного переходов. Приобрел навыки написания программ с использованием переходов,познакомился с назначением и структурой файла листинга

Список литературы

1. GDB: The GNU Project Debugger. — URL: https://www.gnu.org/software/gdb/.

2. GNU Bash Manual. — 2016. — URL: https://www.gnu.org/software/bash/manual/.

3. Midnight Commander Development Center. — 2021. — URL: https://midnight-commander. org/.

4. NASM Assembly Language Tutorials. — 2021. — URL: https://asmtutor.com/.

5. Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: http://www.amazon.com/LearningbashShell-Programming-Nutshell/dp/0596009658.

6. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.

7. The NASM documentation. — 2021. — URL: https://www.nasm.us/docs.php.

8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.

9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.

10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс, 2017.

11. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.

12. Расширенный ассемблер: NASM. — 2021. — URL: https://www.opennet.ru/docs/RUS/nasm/.

13. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВПетербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.

14. Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-е изд. — М. : МАКС Пресс, 2011. — URL: http://www.stolyarov.info/books/asm_unix.

15. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).

16. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science).