

Отчёт по лабораторной работе №7

Дисциплина: Архитектура компьютера

Ким Денис Вячеславович

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	15
	Список литературы	16

Список иллюстраций

4.1	Создание файла в новом каталоге	8
4.2	Ввод текста программы из листинга	9
4.3	Выполнение программы	9
4.4	Изменение текста программы	10
4.5	Запуск файла изменённой программы	10
4.6	Изменение текста программы в соответствии с условием	11
4.7	Запуск файла изменённой программы	11
4.8	Ввод программы из листинга в новый файл	12
4.9	Запуск файла	12
4.10	Удаление операнда	13
4.11	Ошибка в листинге	13
4.12	Проверка работы программы нахождения наименьшей из 3 целочисленных переменных	13
4.13	Проверка работы заданной программы	14

Список таблиц

3.1	Описание некоторых каталогов файловой системы GNU Linux . . .	7
-----	---	---

1 Цель работы

Изучить команды условного и безусловного переходов, приобрести навыки написания программ с использованием переходов и познакомиться с назначением и структурой файла листинга.

2 Задание

В ходе данной работы мне предстоит познакомиться с командами условного и безусловного перехода, приобрести навыки работы с ними изучить структуры файлов листинга, а также написать программы для выполнения заданий.

3 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы.

Например, в табл. 3.1 приведено краткое описание стандартных каталогов Unix.

Таблица 3.1: Описание некоторых каталогов файловой системы GNU Linux

Имя каталога	Описание каталога
/	Корневая директория, содержащая всю файловую систему
/bin	Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям
/etc	Общесистемные конфигурационные файлы и файлы конфигурации установленных программ
/home	Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя
/media	Точки монтирования для сменных носителей
/root	Домашняя директория пользователя root
/tmp	Временные файлы
/usr	Вторичная иерархия для данных пользователя

Более подробно про Unix см. в [1–4].

4 Выполнение лабораторной работы

Создаём каталог для программ лабораторной работы № 7, переходим в него и создаём файл lab7-1.asm: (рис. 4.1).



```
dvkim@dk3n55 ~ $ mkdir ~/work/arch-pc/lab07
dvkim@dk3n55 ~ $ cd ~/work/arch-pc/lab07
dvkim@dk3n55 ~/work/arch-pc/lab07 $ touch lab7-1.asm
dvkim@dk3n55 ~/work/arch-pc/lab07 $
```

Рис. 4.1: Создание файла в новом каталоге

Рассмотрим пример программы с использованием инструкции jmp. Вводим файл lab7-1.asm текст программы из листинга 7.1.: (рис. 4.2).


```

lab7-1.asm      [----]  0 L:[ 1+ 0  1/ 21] *(0  / 591b) 0037 0x
%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

```

Рис. 4.2: Ввод текста программы из листинга

Создаём исполняемый файл и запускаем его. : (рис. 4.3).

```

dvkim@dk3n55 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
dvkim@dk3n55 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
dvkim@dk3n55 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение № 2
Сообщение № 3
dvkim@dk3n55 ~/work/arch-pc/lab07 $ █

```

Рис. 4.3: Выполнение программы

Изменяем текст программы в соответствии с листингом 7.2.: (рис. 4.4).

```

lab7-1.asm      [----]  8 L:[  2+11  13/ 23] *(367 / 612b) 0010 0x
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'
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'
_end:
call quit

```

Рис. 4.4: Изменение текста программы

Проверяем работу файла: (рис. 4.5).

```

dvkim@dk3n55 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
dvkim@dk3n55 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
dvkim@dk3n55 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение № 2
Сообщение № 1
dvkim@dk3n55 ~/work/arch-pc/lab07 $ 

```

Рис. 4.5: Запуск файла изменённой программы

Изменяем текст программы добавив или изменив инструкции `jmp` так, чтобы вывод программы был как в условии: (рис. 4.6).

```

lab7-1.asm      [----] 31 L:[ 1+11 12/ 23] *(358 / 623b) 0010 0x0
#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 _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
_end:

```

Рис. 4.6: Изменение текста программы в соответствии с условием

Проверяем программу: (рис. 4.7).

```

dvkim@dk3n55 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
dvkim@dk3n55 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
dvkim@dk3n55 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение № 3
Сообщение № 2
Сообщение № 1

```

Рис. 4.7: Запуск файла изменённой программы

В качестве примера рассмотрим программу, которая определяет и выводит на экран наибольшую из 3 целочисленных переменных: А, В и С. Значения для А и С задаются в программе, значение В вводится с клавиатуры. Создаём файл lab7-2.asm в каталоге ~/work/arch-pc/lab07. Внимательно изучаем текст программы из листинга 7.3 и вводим в lab7-2.asm.: (рис. 4.8).

```

lab7-2.asm      [-M--]  9 L:[ 1+14 15/ 49] *(292 /1730b) 0110 0x0
#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

```

Рис. 4.8: Ввод программы из листинга в новый файл

Проверяем работу на различных значениях B, например, 3: (рис. 4.9).

```

dvkim@dk3n55 ~/work/arch-pc/lab07 $ nasm -f elf lab7-2.asm
dvkim@dk3n55 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-2 lab7-2.o
dvkim@dk3n55 ~/work/arch-pc/lab07 $ ./lab7-2
Введите B: 3
Наибольшее число: 50

```

Рис. 4.9: Запуск файла

Создаём файл листинга для программы из файла lab7-2.asm. Открываем его с помощью mcedit и изучаем его. 000000E8 B8[00000000] mov eax,msg1 000000ED E81DFFFFFF call sprint - эти строки отвечают за вывод сообщения 'Введите B:'

000000F2 B9[0A000000] mov ecx,B 18 000000F7 BA0A000000 mov edx,10 19 000000FC E842FFFFFF call sread - эти строки отвечают за приём введенного с клавиатуры значения

0000016D E869FFFFFF call quit - эта строка отвечает за выход из программы

Удалим один операнд в любой инструкции с двумя операндами: (рис. 4.10).

```

; ----- Вывод сообщения 'Введите B: '
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'

```

Рис. 4.10: Удаление операнда

В таком случае мы получаем ошибку, которая появляется в листинге: (рис. 4.11).

<pre> 22 0000010b e891ffff ло 23 0000010b a3[0A000000] 24 25 00000110 8b0d[35000000] 26 00000116 890d[00000000] 27 28 28 ***** 29 0000011c 7f0c eck_B', 30 0000011e 8b0d[39000000] 31 00000124 890d[00000000] </pre>	<pre> call atoi ; Вызов подпрограммы перевода символа в 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, ; Сравниваем 'A' и 'C' error: invalid combination of opcode and operands jg check_B ; если 'A>C', то переход на метку mov ecx,[C] ; иначе 'ecx = C' mov [max],ecx ; 'max = C' </pre>
--	---

Рис. 4.11: Ошибка в листинге

Выполняем задания для самостоятельной работы. Напишем программу нахождения наименьшей из 3 целочисленных переменных a, b, c. Значения переменных выберем из таблицы в соответствии с вариантом, полученным при выполнении лабораторной работы № 6 (у нас был 11 вариант). Создаём исполняемый файл и проверяем его работу: (рис. 4.12).

```

dvkim@dk1n22 ~/work/arch-pc/lab07 $ nasm -f elf lab7-3.asm
dvkim@dk1n22 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-3 lab7-3.o
dvkim@dk1n22 ~/work/arch-pc/lab07 $ ./lab7-3
Наименьшее число: 21
dvkim@dk1n22 ~/work/arch-pc/lab07 $ █

```

Рис. 4.12: Проверка работы программы нахождения наименьшей из 3 целочисленных переменных

Напишем программу, которая для введенных с клавиатуры значений x и a вычисляет значение заданной функции f(x) и выводит результат вычислений. Вид функции f(x) выбираем из таблицы вариантов заданий в соответствии с

вариантом, полученным при выполнении лабораторной работы № 6. Создаём исполняемый файл и проверяем его работу для значений x и a из 7.6.: (рис. 4.13).

```
dvkim@dk1n22 ~/work/arch-pc/lab07 $ nasm -f elf lab7-4.asm
dvkim@dk1n22 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-4 lab7-4.o
dvkim@dk1n22 ~/work/arch-pc/lab07 $ ./lab7-4
Введите a: 3
Введите x: 0
Результат: 12
dvkim@dk1n22 ~/work/arch-pc/lab07 $ ./lab7-4
Введите a: 2
Введите x: 1
Результат: 9
dvkim@dk1n22 ~/work/arch-pc/lab07 $ █
```

Рис. 4.13: Проверка работы заданной программы

5 Выводы

В ходе данной работы я изучил команды условного и безусловного переходов, приобрёл навыки написания программ с использованием переходов и познакомился с назначением и структурой файла листинга. Кроме того, я научился писать собственные программы для выполнения различных математических задач.

Список литературы

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.
2. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.