Отчёт по лабораторной работе №7

Дисциплина: архитектура компьютеров и операционные системы

Пестова Ева Константиновна

Содержание

# 1 Цель работы

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

# 2 Задание

1. Реализация переходов в NASM.
2. Изучение структуры файлы листинга.
3. Задания для самостоятельной работы.

# 3 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.

Безусловный переход выполняется инструкцией jmp. Инструкция cmp является одной из инструкций, которая позволяет сравнить операнды и выставляет флаги в зависимости от результата сравнения. Инструкция cmp является командой сравнения двух операндов и имеет такой же формат, как и команда вычитания.

Листинг (в рамках понятийного аппарата NASM) — это один из выходных файлов, создаваемых транслятором. Он имеет текстовый вид и нужен при отладке программы, так как кроме строк самой программы он содержит дополнительную информацию.

# 4 Выполнение лабораторной работы

1. Реализация переходов в NASM

Создаю каталог для программ лабораторной работы № 7, перехожу в него и создаю файл lab7-1.asm (рис. [??]).

![Создание файлов для лабораторной работы](data:application/octet-stream;base64,)

Создание файлов для лабораторной работы

Ввожу в файл lab7-1.asm текст программы из листинга 7.1 (рис. [??]).

![Ввод текста программы из листинга 7.1](data:application/octet-stream;base64,)

Ввод текста программы из листинга 7.1

Создаю исполняемый файл и запускаю его (рис. [??]).

![Запуск программного кода](data:application/octet-stream;base64,)

Запуск программного кода

Таким образом, использование инструкции jmp \_label2 меняет порядок исполнения инструкций и позволяет выполнить инструкции начиная с метки \_label2, пропустив вывод первого сообщения. Изменю программу таким образом, чтобы она выводила сначала ‘Сообщение № 2’, потом ‘Сообщение № 1’ и завершала работу. Для этого изменяю текст программы в соответствии с листингом 7.2 (рис. [??]).

![Изменение текста программы](data:application/octet-stream;base64,)

Изменение текста программы

Создаю исполняемый файл и проверяю его работу (рис. [??]). ![Создание исполняемого файла](data:application/octet-stream;base64,)

Затем изменяю текст программы, добавив в начале программы jmp \_label3, jmp \_label2 в конце метки jmp \_label3, jmp \_label1 добавляю в конце метки jmp \_label2, и добавляю jmp \_end в конце метки jmp \_label1 (рис. [??]).

![Изменение текста программы](data:application/octet-stream;base64,)

Изменение текста программы

Получаю следующий вывод программы (рис. [??]):

![Вывод программы](data:application/octet-stream;base64,)

Вывод программы

Рассмотрим программу, которая определяет и выводит на экран наибольшую из 3 целочисленных переменных: A,B и C. Значения для A и C задаются в программе, значение B вводиться с клавиатуры. Создаю файл lab7-2.asm в каталоге ~/work/arch-pc/lab07 (рис. [??]).

![Создание файла](data:application/octet-stream;base64,)

Создание файла

Текст программы из листинга 7.3 ввожу в lab7-2.asm (рис. [??]).

![Ввод текста программы из листинга 7.3](data:application/octet-stream;base64,)

Ввод текста программы из листинга 7.3

Создаю исполняемый файл и проверяю его работу (рис. [??]).

![Проверка работы файла](data:application/octet-stream;base64,)

Проверка работы файла

Файл работает корректно.

1. Изучение структуры файлы листинга

Создаю файл листинга для программы из файла lab7-2.asm (рис. [??]).

![Создание файла листинга](data:application/octet-stream;base64,)

Создание файла листинга

Открываю файл листинга lab7-2.lst с помощью текстового редактора и внимательно изучаю его формат и содержимое (рис. [??]).

![Изучение файла листинга](data:application/octet-stream;base64,)

Изучение файла листинга

В представленных трех строчках содержаться следующие данные (рис. [??]):

![Выбранные строки файла](data:application/octet-stream;base64,)

Выбранные строки файла

“2” - номер строки кода, “; Функция вычисления длинны сообщения” - комментарий к коду, не имеет адреса и машинного кода. “3” - номер строки кода, “slen” - название функции, не имеет адреса и машинного кода. “4” - номер строки кода, “00000000” - адрес строки, “53” - машинный код, “push ebx” - исходный текст программы, инструкция “push” помещает операнд “ebx” в стек.

Открываю файл с программой lab7-2.asm и в выбранной мной инструкции с двумя операндами удаляю выделенный операнд (рис. [??]).

![Удаление выделенного операнда из кода](data:application/octet-stream;base64,)

Удаление выделенного операнда из кода

Выполняю трансляцию с получением файла листинга (рис. [??]).

![Получение файла листинга](data:application/octet-stream;base64,)

Получение файла листинга

На выходе я не получаю ни одного файла из-за ошибки:инструкция mov (единственная в коде содержит два операнда) не может работать, имея только один операнд, из-за чего нарушается работа кода.

1. Задания для самостоятельной работы

Пишу программу нахождения наименьшей из 3 целочисленных переменных a, b и c. Значения переменных выбираю из табл. 7.5 в соответствии с вариантом, полученным при выполнении лабораторной работы № 6. Мой вариант под номером 8, поэтому мои значения - 52, 33 и 40 (рис. [??]).

![Написание программы](data:application/octet-stream;base64,)

Написание программы

Создаю исполняемый файл и проверяю его работу, подставляя необходимые значение (рис. [??]).

![Запуск файла и проверка его работы](data:application/octet-stream;base64,)

Запуск файла и проверка его работы

Программа работает корректно.

1. Пишу программу, которая для введенных с клавиатуры значений х и а вычисляет значение и выводит результат вычислений заданной для моего варианта функции f(x): 3 \* a, если a < 3 х + |, если а >= 3 (рис. [??]).

![Написание программы](data:application/octet-stream;base64,)

Написание программы

Создаю исполняемый файл и проверяю его работу для значений х и а соответственно (рис. [??]), (рис. [??]).

![Запуск файла и проверка его работы](data:application/octet-stream;base64,)

Запуск файла и проверка его работы

![Запуск файла и проверка его работы](data:application/octet-stream;base64,)

Запуск файла и проверка его работы

Программа работает корректно.

# 5 Выводы

По итогам данной лабораторной работы я изучила команды условного и безусловного переходов, приобрела навыки написания программ с использованием переходов и ознакомилась с назначением и структурой файла листинга, что поможет мне при выполнении последующих лабораторных работ.