Виктор Ващаев Андреевич

Лабораторная

Ващаев Виктор Андреевич

Содержание

# 1 Цель работы

Изучение команд условного и безусловного переходов. Приобретение навыков написания программ с использованием переходов. Знакомство с назначением и структурой файла листинга. # Задание Задание 1 Создаем каталог для программ лабораторной работы № 7, переходим в него и создаем файл lab7-1.asm. Задание 2 Рассматриваем пример программы с использованием инструкции jmp. Вводим в файл lab7-1.asm текст программы из листинга 7.1. Ошибочно написал jump) задание 3: Создаем файл lab7-2.asm в каталоге ~/work/arch-pc/lab07. Изучаем текст программы из листинга 7.3 и вводим его в lab7-2.asm. Задание 4: Создаем файл листинга для программы из файла lab7-2.asm. Открываем файл lab7-2.asm, удаляем один операнд в любой инструкции с двумя операндами. Выполняем трансляцию программы для получения файла листинга. разберём три строки по выбору из листинга, чтобы понять, что в них происходит.

Выбранные строки:

1 Строка 8: cmp byte [eax], 0 2 Строка 14: sub eax, ebx 3 Строка 35: int 80h

1 Строка 8: cmp byte [eax], 0

Описание:

Это команда сравнения (compare). Она сравнивает содержимое байта, на который указывает регистр eax, с числом 0. Регистр eax указывает на текущий символ строки. Если текущий символ равен 0 (конец строки), флаг результата операции обновляется.

Цель:

Проверить, достиг ли указатель конца строки. В строках, используемых в ассемблере, конец строки обозначается символом с кодом 0 (null-терминатор).

2 Строка 14: sub eax, ebx

Описание:

Команда вычитания. Она вычитает значение регистра ebx из регистра eax. В данном случае ebx содержит начальный адрес строки, а eax — адрес конца строки.

Цель:

Вычислить длину строки. После выполнения цикла в функции slen, eax указывает на конец строки. Разница между eax и ebx (адрес начала строки) равна количеству символов в строке.

3 Строка 35: int 80h

Описание:

Это программный прерывание, используемое для вызова системных функций в Linux. В данном случае: eax = 4 — номер системного вызова write (запись данных). ebx = 1 — файл, в который производится запись (1 означает стандартный вывод, т.е. экран). ecx — адрес данных, которые нужно вывести. edx — количество байт для вывода.

Цель:

Напечатать сообщение на экран. Используется для реализации функции вывода строки в функции sprint.

Эти строки показывают очень важные аспекты программы!!! проверку конца строки, вычисление длины строки и вывод строки на экран. Задание 4: любое название, ссылка: /home/victor/Pictures/Снимки экрана/4.png и еще одна ссылка /home/victor/Pictures/Снимки экрана/4.1.png Самостоятельная работа: требовалось в задание для самостоятельной работы

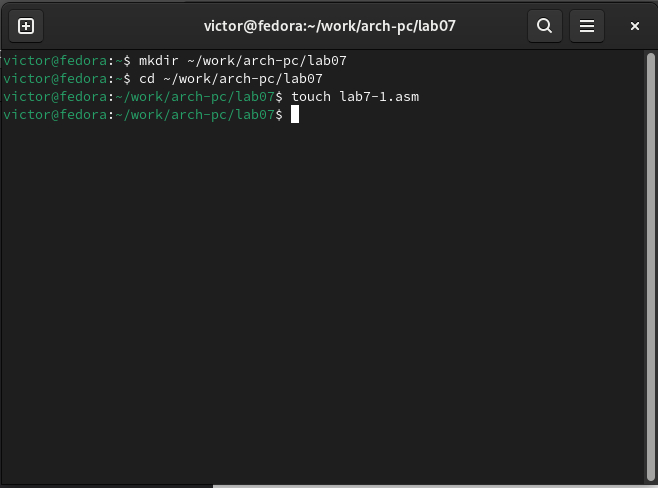
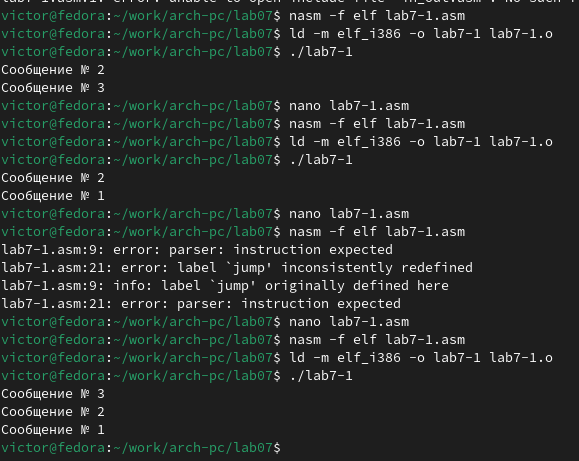
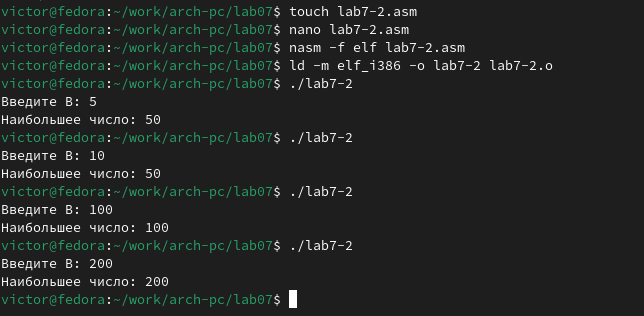
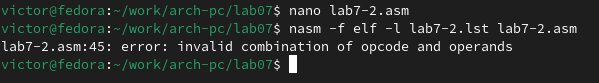
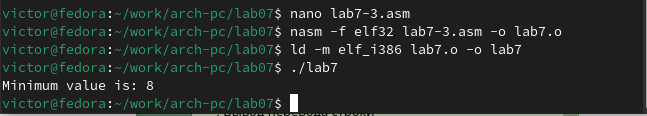
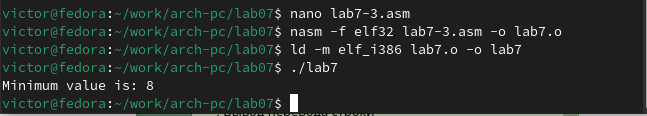
Написать программу для нахождения наименьшей из трёх целочисленных переменных aa, bb, и cc. Значения переменных выбрать из таблицы 7.5 согласно варианту, полученному при выполнении лабораторной работы № 7. Создать исполняемый файл и проверить его работу.  
  
Написать программу, которая принимает с клавиатуры значения xx и aa, вычисляет значение функции f(x)f(x) и выводит результат. Вид функции f(x)f(x) выбрать из таблицы 7.6 согласно варианту, полученному при выполнении лабораторной работы № 7. Создать исполняемый файл и проверить его работу для заданных значений xx и aa из таблицы 7.6.

# 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 про- граммы без каких-либо условий.

# 3 Выполнение лабораторной работы

Изображения в ходе выполнения:

   задание 4   /home/victor/Pictures/Снимки экрана/самостоятельная 2.png 

# 4 Выводы

Я научился писать код и команды безусловного и условного переходов в Nasm. Это очень мне поможет в будущей профессии, оказалось не легко. Команды безусловного и условного переходов в Nasm упрощают работу. # Список литературы{.unnumbered}