

003

1. Написать программу со следующим алгоритмом:
 - ввести с клавиатуры символьную строку в буфер;
 - изменить порядок следования символов в строке на противоположный; положение символа 10 (\n) остается без изменений;
 - вывести результат на экран;
 - завершить программу.
2. Загрузить программу в отладчик. Какими способами это можно сделать?
3. Просмотреть дизассемблированный код программы, введя команду `disassemble _start`
4. Переключить дизассемблер GDB с синтаксиса АТТ на синтаксис Intel и снова выполнить показ дизассемблированного кода. Найти три отличия. Переписать в отчет адрес второй инструкции в формате 0x12345678.
5. Установить точку останова на второй инструкции, указав её адрес.
6. Выполнить программу. Что произошло?
7. Выполнить программу по шагам.
8. Посмотреть содержимое регистров в окне и с помощью команды `info r`.
9. Выполнить программу до места заполнения входного буфера. Вывести содержимое входного буфера в шестнадцатеричном формате и в символьном виде (команда `x`).
10. Выполнить 2 прохода цикла по шагам, контролируя значения регистров. Какие регистры изменяются в цикле?
11. Изменить число проходов цикла на 5.
12. Изменить содержимое выходного буфера. Вводить данные как символы и как десятичные числа.

004

1. Написать программу со следующим алгоритмом:
 - a. ввести символ с клавиатуры;
 - b. преобразовать полученный код в десятичную символьную запись;
 - c. вывести символ и его код.
2. Перевод числа в десятичную символьную запись оформить в виде подпрограммы.
3. Установите <https://github.com/eteran/edb-debugger>
4. Загрузить программу в отладчик. Это можно сделать двумя способами: написать в командной строке `edb --run имя_программы` или запустить `edb` и выбрать программу через пункт Open меню File.
5. Выполнить программу по шагам, нажимая кнопку Step Over панели инструментов или клавишу F7 (находясь в основном окне отладчика), до конца.
6. Поместить в программу точку останова на инструкции, следующей после ввода символа с клавиатуры — щелкнув правой кнопкой по нужной строке дизассемблированного кода и выбрав пункт Add Breakpoint всплывающего меню. Выполнить программу до точки останова, нажав клавишу F9 или кнопку Run панели инструментов. Иметь в виду, что ввод текста с клавиатуры в выполняемую программу осуществляется в отдельном окне EDB Output, а не в основном окне отладчика.
7. Вывести в окне дампа памяти содержимое входного буфера, щелкнув в подокне Data Dump правой кнопкой мыши и выбрав пункт Goto Address всплывающего меню. Адрес вводить в шестнадцатеричной нотации Си (начиная с символов 0x).

8. Зайти в процедуру перевода числа в десятичную запись. Выполнить 2 прохода цикла по F7 (Step Into), контролируя значения регистров. Какие регистры изменяются в цикле?
9. Остальные проходы цикла выполнить по F8 (Step Over). В чем разница?
10. Определить физический адрес выходного буфера в ОЗУ .
11. Вывести ячейки памяти, соответствующие выходному буферу, в подокне Data Dump в шестнадцатичном и в символьном виде.
12. Установить точку останова на инструкцию `div`. Выполнить программу, несколько раз нажав на F8 и наблюдая за изменением содержимого выходного буфера в подокне Data Dump. Каков результат? (Перевод чисел между шестнадцатичной и десятичной системами счисления можно упростить, воспользовавшись программой Калькулятор, выбрав в ней пункт меню Вид).
13. Изменить содержимое входного буфера и проверить, как это отражается на выполнении программы.

005

1. Написать программу со следующим алгоритмом:
 - a. вывести приглашение;
 - b. ввести с клавиатуры строку (предполагается, что она содержит десятичные цифры и любые буквы);
 - c. найти во введенной строке все цифры и для каждой найденной цифры установить в «1» в регистре `ax` бит, номер которого равен этой цифре;
 - d. вывести на экран содержимое регистра `ax` в виде нулей и единиц;
 - e. объяснить полученный результат.

006

1. Написать программу, размещающую в сегменте данных:
 - a. байт со знаком;
 - b. байт без знака;
 - c. слово со знаком;
 - d. число в формате «Короткое вещественное»;
 - e. строка из 3-х символов;
 - f. число в упакованном BCD– формате (два десятичных разряда)
2. Перейдите в сегмент данных и заполните таблицу

Относительный адрес операнда 16сс	Операнд 16сс	Операнд 2сс	Операнд 10сс	Длина операнда в байтах