EE458
Real Time Programming

Lecture 4 - In Class Exercise
Using the C stdio Routines

In this exercise you will write a simple C program using the C stdio routines.

This exercise does not provide step-by-step instructions for adding files to the virtual hard drive, running QEMU, copying files, etc. This is deliberate. Refer to previous exercises for help before asking the instructor for assistance.

Using the stdio Routines

1. Copy the hello_world_c directory that was used in the previous exercise to directory addem under your rtems working directory. (You may want to issue a make clean command to clear out old object and executable files before copying the directory.) Change the name of the test.c program in the addem directory to addem.c. Edit the Makefile and change all occurrences of the word test to addem. Change the name of the program created from hello.exe to addem.exe.

2. Modify the addem.c program so that it prompts the user to enter first an integer and then a floating point number (of type double). The program should then display the sum of the two numbers. It then loops back and prompts for another integer. The application should exit if the entered integer value is 0. Use only the C stdio routines. When run the program output should look like this (user input is shown in bold):

   Enter an integer: 23
   Enter a double: 44.7
   The sum of the two numbers is 67.7

   Enter an integer: 0
   Goodbye!

3. Run make to build addem.exe and copy it to your QEMU virtual hard drive. Run your new application (addem.exe) in QEMU and verify that it works correctly.

Notes:

- You must replace the line
  
  #define CONFIGURE_APPLICATION_DOES_NOT_NEED_CLOCK_DRIVER

  with

  #define CONFIGURE_APPLICATION_NEEDS_CLOCK_DRIVER

  at the bottom of the source code file (before the “#include <rtems/confdefs.h>” line).

- You may need to add fflush(stdout) statements before any input statements in order to see any prior output. Most operating systems will automatically flush standard output when any input statement is executed (to ensure the user will see any prompt that might have been printed), but RTEMS behavior seems to vary from release to release.

- If the program does not run correctly, you may want to compile, run and debug a native version of the code.