

EECS 338 Homework 4: Interprocess Communication

(NOTE: This assignment will be updated later with some more tips and a rubric.)

General requirements:

- Due on the posted due date.
- Upload a single, compressed file (e.g. zip) to Canvas that contains all required files.
- Include either a single makefile that compile all programs or a separate makefile for each.
- Include a typed document with a description of any techniques that you used to control the output order.
- All work should be your own, as explained in the Academic Integrity policy from the syllabus.

Instructions: The purpose of this assignment is to work with sockets, pipes, and threads.

1. Using C sockets (not Java), create a server that sends a client a question, receives the client's answer, and sends the client the correct answer with a message indicating whether the client was correct. The server should print one message to indicate when it is listening and a second message to indicate whether the client's answer was correct. The client should print the server's question, allow the user to type an answer, and print the server's response. You may use a fixed port number and IP address of 127.0.0.1. Include output in your report for two cases: where the client answers correctly, and where the client answers incorrectly. Below are two examples.

Example where client answers correctly...

Server:	Client output (user input in bold):
<pre>\$./server Listening for client... Client was correct.</pre>	<pre>\$./hw4_client How many steps does it take for a client to change a light bulb? 3 Correct! The 3 steps are: create a socket, connect to it, and send a new bulb.</pre>

Example where client answers incorrectly...

Server:	Client output (user input in bold):
<pre>\$./server Listening for client... Client was correct.</pre>	<pre>\$./hw4_client How many steps does it take for a client to change a light bulb? 1 Nope. The 3 steps are: create a socket, connect to it, and send a new bulb.</pre>

2. The program "shubert.c" demonstrates the use of the Shubert function, which is a test case for optimization algorithms. Using fork() and pipe communication, find the minimum value of the Shubert function for $-2.0 \leq x_1, x_2 \leq 2.0$. Use at two processes such that each performs an equal amount of work. The parent process should print the global minimum value. Include the output in your report.
3. Repeat #2 using POSIX threads with one child (two threads total). Include the output in your report. Additionally, explain in 2 - 3 sentences whether you like #2 or #3 as a solution to this analysis problem.

Interested students can find more information and graphs of the Shubert function here:

- <http://www.sfu.ca/~ssurjano/shubert.html>
- <http://profesores.elo.utfsm.cl/~tarredondo/info/soft-comp/functions/node28.html>

Tips:

- *Compiling on eecslinab servers:* To use the “cos” function on an eecslinab server, you need to link the math library using “-lm”. For example: `gcc -o prog prog.c -lm`
- *More tips to come...*

Rubric is forthcoming...