CSC / ECE 573 Internet Protocols, Fall 2005

Homework #3

Due Date

- Due Tuesday, October 18, at 11:45pm

Instructions

- Homeworks should be submitted individually. We will use the standard submit utility for our class to submit all work, which means your work must be prepared electronically.
- Put your name, the assignment number, and date at the top of the first page. Put solutions in order (don't make the TA hunt for your solution).
- Do not plagiarize; that means, do not copy content from any source without permission from the instructor, and if permitted, acknowledge the source.
- This homework is worth a total of 100 points

Problems

1. I ran "netstat -avn" on a computer and found the following TCP ports in the "LISTEN" state. What services are these?
   ```
   25 111 113 135 139 143 445 631
   ```
   Is your computer running services other than these (use netstat to find out).

2. How can the select() call be used in a way such that it blocks if there are no active sockets, but only for a maximum of 5 seconds?

3. How can you do a non-blocking recv() on a socket, and how can you tell if there was any data read on return from the call?

4. What happens if a process attempts to bind to port to which some service is already bound?

5. For the string "32.14.5.6", what does inet_addr() return? What is the byte ordering of this result in memory on an Intel architecture machine?

6. Suppose a 16-bit integer with the decimal value 5000 is sent from a machine with a big-endian architecture to a machine with a little-endian architecture. Before sending, the sender does not do any byte-ordering conversions.

   1. If the receiver executes "y = x", where x is the received 16-bit quantity, what is the value of y as viewed on the receiver machine (if, for instance, it is printed out)?
   2. If the destination executes "y = ntohs(x)", where x is the received 16-bit quantity, what is the value of y?
   3. If the destination executes "y = ntohs(ntohs(x))", where x is the received 16-bit quantity, what is the value of y?

7. A server creates a "master" socket and then creates 3 “child” processes. Each child issues an accept() call on that socket, before any packets arrive. Is there anything wrong with that? Explain.

8. A parent process initializes a variable X to the value 33. It then creates (forks) 3 child *processes*. Each child process increments the value of X and prints out the new value to the standard output. What possible sequences of output values could be printed?

9. Answer the same question as above, but using threads rather than processes (no locking or mutual exclusion of any kind is used).

10. Three methods of concurrency are to fork() on demand (dynamically), to "pre-fork()" (statically), or to use delayed

process allocation.

1. If an application server receives 50 client requests per second, each request requires .010-.015 seconds to process, and the overhead of fork() is .01 seconds, what method would you use and why?

2. If an application server receives on average 10 client requests per second, each request requiring between .01 seconds and 50 seconds to process, and the overhead of fork() is .05 seconds, what method would you use and why?

3. If an application server receives 1 client request every 10 seconds, and each request requires .2 seconds to process, and the overhead of a fork() is .01 seconds, which method would you use and why?

11. In the programming example given in lecture using the "select()" function:

    1. How would you limit the maximum number of concurrent clients to 16?

    2. How would you arrange that "older" (long-running) connections get lower processing priority than "younger" connections?

    3. How would you make sure that any incoming connection requests will not have to wait more than 10 seconds before a connection is established by the server?

Programming

For these questions, you can use either java, C++, or C. Consider these practice exercises for the project. Include your source code in your assignment, not as a separate file, but simply inline as text. We may ask you to demonstrate your code for us in some cases.

If you wish to use any of the code that Comer/Stevens Vol. III (our optional textbook) provides, that is fine. See the website to obtain it (both Linux and Windows versions are available). You can also use other code from the web for this assignment, but if you do, cite the source of your code.

1. Modify `TCPechod` (or equivalent) so that it prints a message (to standard out) every time a client establishes or terminates a connection with it. In this message, include the dotted decimal representation of the IP address of the remote host connected to that socket, and the port number. (ex.: "Connection terminated by remote host 152.14.62.65, port number 1056") Test your program by connecting to it several times from different ports concurrently (using, for instance, Comer's `TCPecho` program.)

2. Write a UDP client and server application (suggestion: start with `UDPecho` and `UDPtimed`). The client should send a riddle to the server. The server responds to the client with the appropriate answer (which the client will print out). You may hardcode into your programs whatever riddle(s) you wish to use as examples.

3. Write a basic ftp client and server using TCP. The client simply connects to the server and then sends "get xxxx", where "xxxx" is the pathname of a file on that system. The server then reads the specified file and sends it to the client, which stores it in a temporary directory. The connection is then closed. If the file is not found the server should simply close the connection.

Created on October 8 , 2005
Last Modified October 8, 2005
Maintained by Douglas S. Reeves