CECS 327 Assignment 3 - Processes

20 points

**Assignment Description**. Answer the following questions from the Chapter 3 reading from your textbook.

Be through and complete with your answers. You may work on these questions with a partner (no

more than two working together), but both students must submit the document individually on Beachboard

Dropbox along with both students’ names on each submission.

1. Statically associating only a single thread with a lightweight process is not such a good idea. Why

not?

Making an association like this would make it so that all the threads would be kernel level and any performance gained would be lost.

2. Having only a single lightweight process per process is also not such a good idea. Why not?

Rather than like in the last question you now have only user level threads and any calls that block will brick the entire process.

3. Describe a simple scheme in which there are as many lightweight processes as there are runnable

threads.

Using a lightweight process pick a runnable thread. If a runnable thread is found a new lightweight process will be started in it’s place and if not the lightweight thread will terminate itself.

4. Constructing a concurrent server by spawning a process has some advantages and disadvantages compared

to multithreaded servers. Discuss a few.

Advantage: Since the processes are separate they cannot hurt each other.

Disadvantage: Making new processes is costly and it is also a missed opportunity to use threads which could have been cheaper to use.

5. How can we prevent an application from circumventing a window manager, and thus being able to

completely mess up a screen?

I believe you mentioned this in 326 also, but utilizing the microkernel approach you can make sure that all window operations required go through the kernel.

6. Is a server that maintains a TCP/IP connection to a client stateful or stateless? Why?

Because the server only maintains the connection on the client it is stateless. Since the server does not care about the local TCP/IP information on the OS of the client it is stateless.

7. Imagine a Web server that maintains a table in which client IP addresses are mapped to the most

recently accessed Web pages. When a client connects to the server, the server looks up the client in its

table, and if found, returns the registered page. Is this server stateful or stateless?

Since interactions between the client and the server can continue regardless of whether or not the information on the web page is present the server would be stateless. If it was stateful services would be interrupted until the server fully recovers from the error.

8. Consider a process P that requires access to file F which is locally available on the machine where

P is currently running. When P moves to another machine, it still requires access to F. If the

file-to-machine binding is fixed, how could the systemwide reference to F be implemented?

In order to implement this you would need to make a separate process that can handle remote requests for the file. The new process will act as a file server for P to access Q.

9. Describe in detail how TCP packets flow in the case of TCP handoff, along with the information on

source and destination addresses in the various headers.

From what I read the best way would be to let a front end execute a three-way handshake and then forward packets to a selected server. That server sends TCP PDUs with a source address shared with the front end.

10. Strong mobility in UNIX systems could be supported by allowing a process to fork a child on a remote

machine. Explain how this would work.

The new child process created by the fork would need the proper amount of resources reserved by the target OS. The appropriate processes and a memory map would need to be made for the child process. The parent’s image can now be copied to the child into the reserved space.