Network Programming Exam

August 26th, 2009

This is a closed book exam: no documentation is allowed

---

1 Program Output (1 point)

What will be the output of the following program? In which order will the messages appear on screen?

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>

int main() {
  if (fork()==0) {
    fork();
    if (execl("/bin/echo", "/bin/echo", "/bin/echo", "foo", 0) == -1) {
      fork();
    }
    printf("bar\n");
  }
  else {
    fork();
    if (execl("/bin/does_not_exist", "/bin/does_not_exist",
              "/bin/does_not_exist", "baz", 0) == -1) {
      fork();
    }
    printf("bat\n");
  }
}
```

The `/bin/echo` program simply outputs all parameters passed to it, plus a new line at the end. The `does_not_exist` program does not exist.

2 Questions (5 points)

1. What is a zombie process? How can you destroy it?
2. In Sun-RPC, when is it wrong to use UDP as the transport protocol?
3. Can a CORBA client access an RMI server? Can an RMI client access a CORBA server?
4. Any JSON document is actually a valid JavaScript statement. Why do we need a parser library to generate a Javascript structure from it rather than simply executing this code?
5. When does GPG ask a user for his/her passphrase?

3 A Self-Updating Anti-virus Program (4 points)

We want to write a new anti-virus program. As all good anti-viruses do, our program automatically refreshes its list of virus descriptions so that the computer can be protected against new viruses. In this application, it is essential to reduce the delay between the publication of a virus description and the moment when the description is known by the computers to protect.

3.1 Initial Design

We decided to write our anti-virus in C. We want to design it such that it contacts a given server every 10 minutes to check if there are any new viruses in the database. If so, then the virus descriptions are downloaded. A virus description is a text string of arbitrary size.

We now have two problems to solve: how to issue a task every 10 minutes, and how to check for updates.

Describe the design of a program that can handle these requirements and explain why they solve the problem. Give a brief sketch of a C program which runs a task periodically in addition to its other duties. You do not need to sketch the code which actually contacts the server for updates.

3.2 Scale

We expect that our anti-virus will be successful and will soon have 1,000,000 users in the near future. This means that the server should treat on average 1666 requests per second. This is well beyond the capacity of the server. We decide not to upgrade the server itself, but to change the structure of the application.

In the new application structure, each client registers with the server so that the server can contact its clients when a new virus description is published. The problem now it to contact 1,000,000 clients in the shortest amount of time. The delay to contact one client is on average 1 second, so sequentially contacting one client after the other would take too much time.

Describe the structure of a single-process, single-thread program which can contact clients sufficiently fast so that each client receives the information at most one hour after a virus description is published.

We assume that client machines remain online permanently.

3.3 Security

Clients want to make sure that virus descriptions are authentic and actually originate from the authorized source.

Describe how such guarantees can be provided:

1. Which information(s) must be known to the clients beforehand
2. What the server must do before sending a virus description to a client
3. What the client must do to authenticate a virus description received from the server.

— the end —