

Internet Programming Exam

Monday, January 6th, 2014

This is a closed book exam: no documentation is allowed

WRITE CLEARLY!!!

1 Program Output (1 point)

What will be the output of the following program? If there are multiple possible outputs, show them *all*.

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

int main()
{
    if (fork()==0) {
        fork();
        if (execl("/bin/echo", "/bin/echo", "A", 0) == -1)
            printf("B\n");
        printf("C\n");
    }
    else
        if (execl("/bin/does_not_exist", "/bin/does_not_exist", "D", 0) == -1)
        {
            fork();
            printf("E\n");
        }
}
```

Assumptions:

- Assume that the call to `/bin/echo` succeeds, and (as expected) outputs all parameters passed to it, plus a new line at the end.
- Assume that printing of `"A\n"` is atomic, that is, no other process can be swapped in after printing the character but before printing the new line.
- Assume that program `/bin/does_not_exist` does not exist.

2 Questions for Brief Answers (4 points)

SAMPLE QUESTION: What does the `listen()` system call do?

SAMPLE ANSWER: It is relevant only in the context of TCP. It promotes a client socket to a server socket, and sets the *backlog*, that is, the size of the queue holding incoming clients that have not yet been `accept()`ed.

1. What does the `signal()` function do? When is it used?
2. What is the Port Mapper? What does it do?
3. 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?
4. What is a DOM tree? Show a simple example. What is a DOM tree useful for?
5. What is the role of a *Certificate Authority (CA)*? Explain (optionally draw) the communication between a server, a client, and a CA in a typical situation. What you need to show is *who* sends *what* information to *whom* and *when*.
6. In RPC, when would you prefer UDP as the transport protocol and when TCP?
7. Node A is sending a message to node B, and uses a public/private key pair for *signing* it. *Who* encrypts the message, and *which* key is used for encryption?
 - (a) A's private key?
 - (b) A's public key?
 - (c) B's private key?
 - (d) B's public key?
8. (continued from previous) : What does A achieve by signing a message to B?

3 ScroogeMcDuck.nl: Price Portal (5 points)

We are designing `www.ScroogeMcDuck.nl`, a web application that compares product prices among many online stores. When a user visits the web site, he can select through a list of products, and a listing of online stores that currently offer this product is displayed on the user's browser, along with each store's price. The list is displayed sorted in ascending price order.

Not all products are offered by every store, and product availabilities are dynamic (e.g., a store can start or cease offering some product). `ScroogeMcDuck.nl` maintains for each product the list of online stores that currently offer it. When a user queries about some product, our web application contacts all online stores that offer that product (and only these) to collect their latest prices.

We assume that the list of online stores registered with our application is fixed (and predefined) for this exercise.

3.1 Interface to online stores

Two APIs exist for communicating with the online stores: One to maintain the list of which products are available at each store, and one to retrieve the latest prices. Here focus on the first one, and ignore the second.

QUESTION 1: What policy would you choose for *updating product availability* information: *push* or *pull*, and **why**? By *push*, online stores actively send updates on product availabilities to ScroogeMcDuck.nl. By *pull*, ScroogeMcDuck.nl actively retrieves from an online store information on whether a certain product is offered.

QUESTION 2: We choose to use Java RMI for the communication between our application and the online stores. List the methods (names + input parameters + return values, but *not* the implementations!) that you would define to support the functionality you suggested in Question 1. The arguments or return values that (may) need to be passed are:

```
String store;    String product;    boolean availability;
```

QUESTION 3: If an online store has a C-based framework, what can it do to enable communication to ScroogeMcDuck.nl through RMI? Do *not* write code or function signatures. Simply explain the concept, stating in detail what runs where, in which language, and what mechanism(s) is(are) used for communication.

3.2 Interface to users (browsers)

Some online stores respond to a price polling really fast, but others take many seconds. In order to make ScroogeMcDuck.nl fast responsive, we cannot afford sending the price listing to the user only after all stores offering a certain product have responded, since some will be very late resulting in a poor user experience.

Instead, we want the user's browser to be dynamically updated with new entries as soon as the price quotes from the respective stores reach the server. And these entries should be sorted based on ascending price order.

QUESTION 4: Explain in detail how such functionality can be provided: which functionality must be provided at the Web server side? Which functionality must be provided at the Web browser side? How do they interact with each other?

3.3 Security

A business competitor —or a talented IntProg student!— could potentially intercept the communication between ScroogeMcDuck.nl and an online store, to mess around with product availability or pricing information.

QUESTION 5: Suggest a mechanism (describe the logic) for guaranteeing secure communication between ScroogeMcDuck.nl and registered online stores.

— Good luck!! —