S-38.3610 Network Programming / Examination 16.5.2012

Answers to separate form.
Remember to state your name and student number

1. **Describe how the following operations and their functionalities differ between TCP and UDP. Tell also what happens in the network as a result of these calls (5p)**

   - connect
   - bind
   - send
   - recv
   - close

2. **You get a C program that maintains student data in a linked list like this:**

```
struct student {
        char name[80];  // contains null-terminated string
        char gender;    // M or F
        uint8_t birthdate;
        uint8_t birthmonth;
        uint16_t birthyear;
        uint32_t studentnumber; // 32-bit integer
        struct student *next;// pointer to next element in linked list
                             // NULL on last element

};
```

Your task is to design functions that send and receive the student data over the Internet to another instance of the student database program using TCP. An additional requirement is to be efficient in the number of bytes transmitted over network, e.g., to prefer binary number formats instead of text encoding, and to avoid unnecessary unused space between the fields.

How would you send the linked list of students to the network? How would the receiving end be able to construct the same linked list of students? How does the receiver know when all students are received? Note that the two communicating programs may be running on different hardware platforms. Describe the design of your functions and main considerations in sending and receiving the student data correctly across different Internet hosts. You can skip the connection establishment phase, and just focus on sending and receiving data. (6p)

3. **Connection establishment with TCP can fail, for example, when a) client attempts connection to a port that server is not listening, or b) when server host is not connected to network. Describe what happens in these failure situations and how does the client application notice the error. How would the situation be different with UDP socket? (4p)**

4. **You need to implement function connect_by_name(char *name, char *service, int type, int secs), that opens connection to server with hostname and port (or service name) as indicated in parameters. The parameters are as follows:**
   - name: the hostname (or string IP addresss) to connect
   - service: the service name or port number to connect
   - type: socket type (stream or datagram)
   - secs: maximum number of seconds before the call returns. The execution of the call must not take longer than this time.

   The call returns socket fd if connect was successful, or -1 if there was error, or if 'secs' number of seconds passed from the call. The call must support both IPv6 and IPv4.

   Outline how would you implement this function. Pseudocode-level is sufficient (no need for C code), but you should point out the main Posix API calls needed and the main logic (conditions, loops, etc.) in your solution. For each function call, describe shortly why you are calling it and what the function does. Remembering the exact parameter format of functions is not important. (5p)

5. **Your task is to design a multiuser chat server, where messages written by each user are shown to all users currently connected to the server. New clients can join the session and clients can leave the session at any time. When a client joins or leaves session, all other connected clients should be informed about it. Any (mis)behavior of one client should not block others from continuing chat. Discuss the suitability of the following design approaches for the server, and what benefits or challenges are in each of them, considering this particular task (6p)**

   - Single-threaded server, multiplexing communication operations with select
   - Separate process per client, created as client connects the server
   - Separate thread per client