BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
I SEMESTER 2011-12
IS C462 : Network Programming
Assignment
Weightage: 15%   Due Date of Submission: 15-Nov-2011 midnight

----

Note:

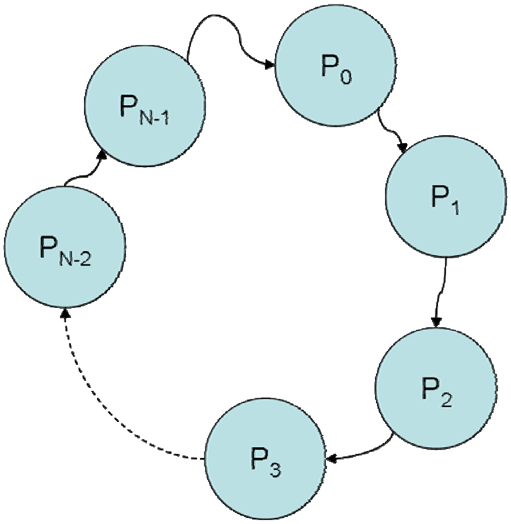
1. Assignment groups are of three members, maximum. Please submit group information latest by 21st October through email.
2. Marks will be awarded based on output and viva.
3. Copying cases will be dealt as per handout policy.
4. Create a directory for each problem like p1, p2 etc. The tarred file having named like <idno1>_<idno2>_<idno3>.tar needs to sent by email (haribabu.bits@gmail.com)
5. The assignment has 2% bonus marks.
6. **In each problem, print to console the data that is required to demonstrate the solution for the problem at hand.**

P1. Write a mini inetd daemon server. Assume that the following array of structures is available in your program with data for each service. [2]

```
struct service{
            int16_t port;
            char   protocol[MAX]; //tcp or udp
            char concurrent[MAX]; //yes or no
            char  serverprogram_path[MAX]; //path of the
                  executable
            };
```

P2. Write programs namely parent.c and game.c for the following requirements. Use appropriate IPC. [2]

a. Parent takes N, K as command-line argument and creates N children. Note: N<K. Every child knows only one of the other children as shown in the figure below. Every child is loaded with a game executable compiled from game.c. Parent sends a SIGUSR1 signal to one of the children. That child sends the integer K to the next child it knows. The next child reduces K by 1 and sends to the next child. Like this it goes on. Whenever a child receives it prints “process <pid> received <number>”. In the process of decrementing and sending, whichever child receives 0, it sends K to the next and it prints “i am a foolish process <pid>, defeated”. Next time when it receives an integer, it will simply forward it to the next child without decrementing. That means it will not participate in the game. This way, one after another, the children will be defeated and become passive observers of the game. If a child receives the same number K it has sent, that means it is the winner. That child will print “i am a truthful process <pid>, only the truth always wins>



P3. Chat servers generally use I/O multiplexing based concurrency due to uncertainty in timings of client’s responses. But to have stability and scalability in the system, it is planned that the server will use combination of forking with select() as the model for concurrency. No. of clients per process are defined to be CPP (clients per process). Assume a simple chat protocol that client sends two messages join and chat. In join message, client sends its nickname and in chat it sends the nickname of the person to whom it needs to be delivered and the contents. Write a program for the server that achieves concurrency for this protocol. Develop a simple client that sends join and chat messages for testing purpose. [2]

P4. Consider a new operator ‘||’. It will take the output of the previous program and pass it on as input to two different programs. Give a program segment for implementing the following command: `ls –l | uniq || grep ^d, grep ^-` [2]

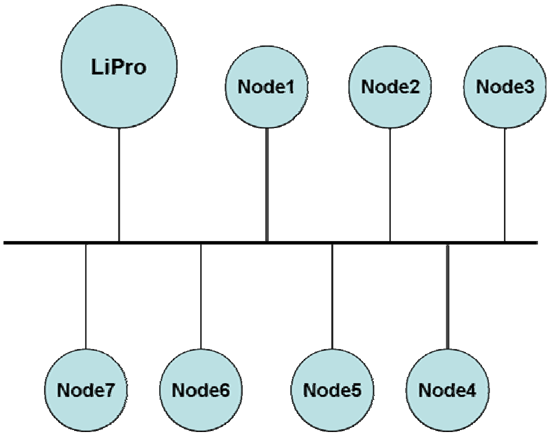
P5. It is a common requirement for a simple request-reply UDP based application to support reliability features at application layer level. Develop a set of API or functions that can enable any application to use UDP with reliability without any understanding of underlying implementation. At the most, application is supposed to call these API in a certain sequence. Develop a simple client and server that use these API. [2]

P6. Consider the following diagram. All nodes are connected over a LAN. Consider the need for group communication among the users on the LAN. There are a number of groups constituted by the administrator of this network. These groups are added/deleted/updated by the administrator in a dynamic way. That means nodes can’t remember the addresses of groups. These groups are maintained by the LiPro server in a data structure. Assume that the following are the groups:

Networks	multicast-ip	port
Databases	multicast-ip	port

The nodes every time when they start-up, send a broadcast request to well-known port number. If the LiPro server is up, it will reply with the list of groups, ips and port numbers. Nodes upon receiving the list, display it to the user. User selects the groups, he wants to join, iteratively. Nodes join the selected groups. Nodes also provide interfaces for sending messages to any of the selected groups. LiPro server takes well-known port number, group1-name, group1-ip, group1-port, group2-name, ... on the command-line. It displays the broadcast requests it receives from nodes with their addresses. The node takes the well-known port-number on command-line. It displays the messages received from groups.

Implement LiPro.c and Node.c for the above specifications. [2]



P7. Write a RPC client and server using SUN ONC framework. Server maintains a database of railways. List of trains, fare etc. When user types a train id, client fetches the details from the server and displays them on the screen. User should be able to perform transactions like booking a ticket etc. Assume dummy data in the database. Develop protocol file and accordingly implement RPC client and RPC server. [2]

P8. Write a thread-based-application using raw sockets that does the job of `traceroute` command. Generally traceroute application, for each TTL, sends UDP datagrams sequentially and waits for the ICMP replies. To make it faster, use multiple threads which can send UDP datagrams in parallel and receive replies. Finally the program should print thread id and the reply received. [2]

P9. Develop a web client application that communicates with a web server over secure connection. It takes URL and fetches the data and displays the data. [1]

==End of Assignment==