BIRLA INSTITUTE OF TECHNOLOGY & SCIENCES, PILANI

II Semester 2010-2011 IS C462 Network Programming
Comprehensive Exam Part A (Closed Book)
Date: 5th May 2011 Maximum Time: 90 Mins Max. Marks: 20

Note:

- Write the answers in the comprehensive answer sheet provided
- **Answer the questions and their parts in sequence**
- Overwritten answers will not be accepted for rechecks
- After submitting part A, collect part B

Q1) Write the best answer. Each carries 0.25 mark.

1. The system call that converts active socket into passive socket
   `listen()`
2. The TCP connection state in which data transfer takes place
   ESTABLISHED
3. The call that creates a child process where the child always executes first
   `vfork()`
4. The following call is required to reading OOB data inline?
   `sockatmark()`

Q2) Write the purpose of the following calls in brief. Each question carries 0.25 mark

1. `dup2(): duplicates the descriptor pointing to the same file table entry`
2. `select(): used for I/O multiplexing`
3. `getsockname(): used for getting local end point address of the socket`
4. `socketpair(): used to create a socket-pair between parent and child`

Q3) Write the relevance (in at least two distinct points) of the following from network programmer's point of view. Each question carries 0.5 mark

1. RST segments:
   1) to report non-existent listening server for TCP connection
   2) to abort existing connection
2. Unix domain sockets
   1) to achieve faster communication if the processes are within the system
   2) to pass file descriptors
3. Raw sockets
   1) To access IP level packets
   2) To write services that need ICMP, IGMP protocols
4. ICMP
   1) To report error messages in IP layer
   2) To implement network measurement tools
5. BPF
   1) To access directly packets at data link layer
   2) To write services such as RARP
6. OOB data
   1) To support urgent communications
   2) To indicate control data

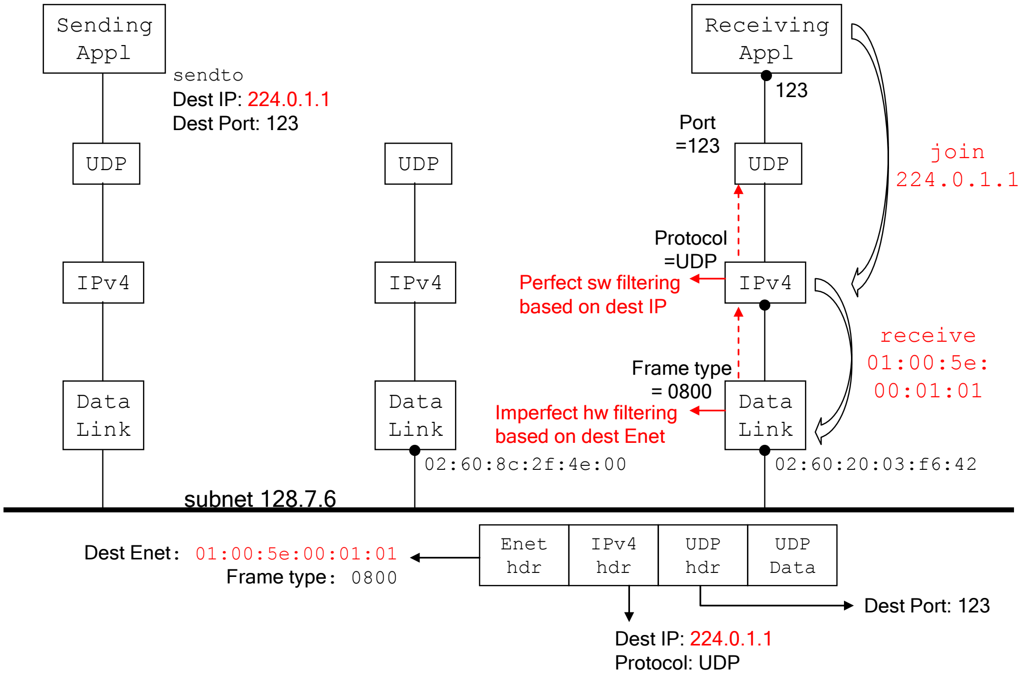
Q4) Answer <u>any 7</u> questions but following the sequence. Each carries 1 mark

1. Mention the parameters over which the concurrency model is chosen? Explain how they should be considered in making the decision?

```
Number of connections
Shared memory
Code complexity
Frequency of new connections
Length of connections
Stability
```

Context switching

2. Explain the multicasting in terms of message processing at different layers of TCP/IP stack. What would be done t join a multicast-group 239.0.0.1?



3. Explain the 'preforking without locking around accept()' model for server design. Mention its disadvantages and advantages.

Each child waits on accept() call. The number of pre-froking processes may not be enough to cater clients' needs. The advanateg is that process creation cost and load distribution cost is minimal.

4. Explain the most used technique for making network communication (the application layer data) secure? How to develop such network applications?

Tunneling using SSL. OpenSSH library.

5. A netprog student was claiming that he has developed a tool using which he can find out ips of all systems in the local network. Mention how it can be done.

UDP broadcasting and replies from servers running on all machines. Or raw sockets to read replies.

6. Mention and illustrate an application of pthread condition variables during coding of your assignments in this course.

7. Mention and illustrate an instance of using co-process in server design during coding of your assignments in this course.

8. During transition phase from IPv4 to IPv6, provision has been made for their interoperability. But it is stated that an IPv6 client running on a dual-stack host will have problem in interoperating with the IPv4 server running on a dual-stack host depending on the type of destination address the client chooses. Explain the issues if the client queries type A (IPv4-mapped-IPv6 address) or AAAA (IPv6 address) record from DNS for the server?

Type A will succedd because it is ipv4-mapped address which can be converted to ipv4 by dual stack. Ipv4 is understood by ipv4 server. In the other case it is not.

9. What is daemon process? Explain how the following code makes one a daemon process?

```
if ( (pid = fork()) != 0) exit(0);
setsid();
```

```
it creates session: its association with the terminal is removed.
```

Q5) Write the necessary code for the following. Each carries 1.5 marks.

1. Parent process creates N children and wants all the children to write data to a shared memory segment. Parent reads the data and prints. Every child writes in a shared memory location having offset as $N \mathrm{mod} childpid$.

   Parent:
   Create shared memory
   Attach it.
   Create N children

   Wait() for all
   Read and print

   Child:
   Write data at the specified location

2. Client that communicates with an ‘echo’ server running at ‘/var/echo.11’ using unix domain datagram protocol.

```
#include    "unp.h"

int
main(int argc, char **argv)
{
    int     sockfd;
    struct sockaddr_un servaddr, cliaddr;

    sockfd = Socket(AF_LOCAL, SOCK_DGRAM, 0);

    unlink(UNIXDG_PATH);
    bzero(&servaddr, sizeof(servaddr));
    servaddr.sun_family = AF_LOCAL;
    strcpy(servaddr.sun_path, UNIXDG_PATH);

    Bind(sockfd, (SA *) &servaddr, sizeof(servaddr));

    dg_echo(sockfd, (SA *) &cliaddr, sizeof(cliaddr));
}
```

Q6) Write the code for the following cases. 2*2.5 marks

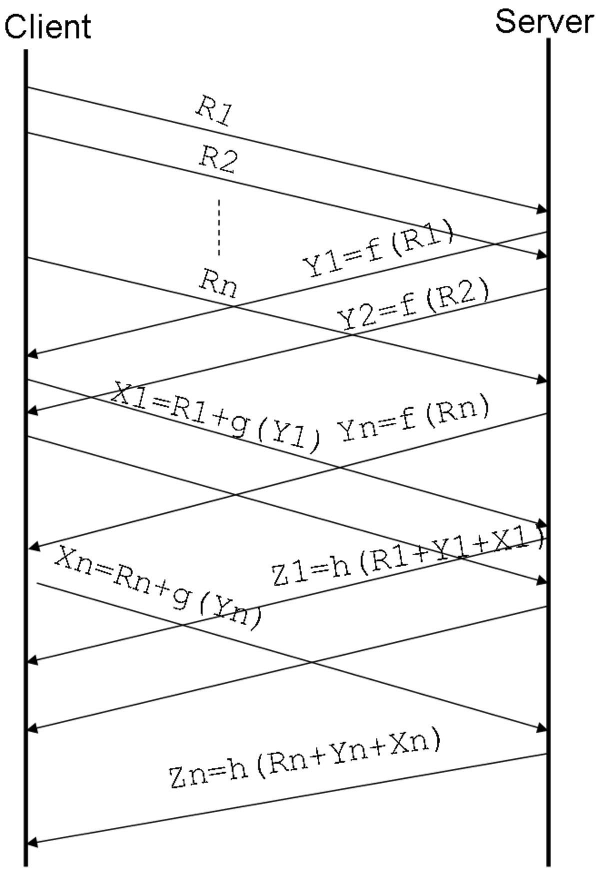
1. Write a mini *inetd* daemon server. Assume that the following array of structures is available in your program with data for each service.

```
      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
                  };
      struct service services[MAX_SERVICES];
```

```
The program:
Read each record from file
If tcp, create tcp socket. If udp, create udp socket.
Call, bind(), listen for the sockets.

Use select fopr listening over the sockets.
```

2. A client communicates with a server in a session protocol as per the following diagram over a single TCP connection. Write the code for the client. f, g, h are functions.



Client sends N messages.
Each message has a unique no.
keep r1….rN in a buffer.
When the relply comes apply function over
the reply and add r and send.

End of Part A