BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
IS F462 : Network Programming
II Semester 2014-15
Assignment-2
Weightage: 8% (24M)   Due Date of Submission: 23-APR-2015

==

Important to Note:

1. Group of maximum 3 students.
2. **Don't use temporary files and system() function.**
3. **<u>Only working programs will be evaluated. If there are compilation errors, it will not be evaluated.</u>**
4. *<u>Provide makefile for each problem.</u>*
5. Upload instructions are given at the end.
6. For any clarifications please contact me (khari@pilani.bits-pilani.ac.in).

Plagiarism will be thoroughly penalized.

==

P1. Write a program webserver.c for concurrent web server supporting HTTP GET requests. It supports concurrency through a event-driven model, using epoll() edge-triggered notifications.

- server maintains a message queue to queue the events.
- main thread waits on epoll_wait() to get IO notifications from kernel. For each IO notification, it adds an event to the message queue.
- "process" thread waits on message queue msgrcv() call. It gets a message and responds according to its state. While processing the message it may add new events to the queue.
- each client request goes through states: READING_REQUEST, HEADER_PARSING, READING_DISKFILE, WRITING_HEADER, WRITING_BODY, DONE. Next state is reached only after the previous one is complete.
- All IO (read and write) operations are non-blocking. This necessitates buffer management.
- data pertaining to client's request is stored in a central data structure such as hash table. This is to avoid unnecessary copying of data which may happen if we keep client data in the message queue itself.
- When a request reaches DONE state, connection remains open for another request. Client connection is closed only when EOF is received from client. That may happen during any state.
- web server testing tools such apache ab or httpperf can be used as clients.

[10]

P2. Write a C program *multicast.c* for IP Multicast that does the following.

- It takes multicast group ip and port on command line. It joins the group and waits for messages.
- It sends a "hello-" + time() every 15 seconds to count how many members are present in the group.
- Any member who receives "hello-"+time(), simply echoes the same.
- The member which has sent hello, counts the replies received within 5 seconds and displays count on the screen.
- Same program is run multiple times to create multiple members.
- Each member prints the sender ip and message received every time a message is received.

- member exits only when Ctrl-c is pressed. Before it exits, it sends "bye-" +time() to all in the group.

[7]

P3. Write a program synflood.c using raw sockets which works in the following way.

- it takes server hostname and port on command-line.
- it creates TCP SYN segment and appends ip header and sends to the remote host. It uses IP_HDRINCL socket option to include IP header.
- it sends TCP SYN segment every 1 second, every time with a different source ip and source port. These addresses/port can be random numbers in valid range.
- it should display the received replies (SYN+ACK) from the server using libpcap library.
- Program will not send any reply (ACK) segment to webserver.

[7]

How to upload?

- Create group.txt file and put idno, name of members into this file.
- Make a directory for each problem like P1, P2 etc and copy your source files into these directories.
- Tar all of them including group.txt into idno1_idno2_idno3_ass1.tar
- Upload on nalanda (http://nalanda).

===End of assignment 2===