BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

IS F462 : Network Programming

I Semester 2014-15

Assignment-2

Weightage: 8% (24M)   Due Date of Submission: 25-NOV-2015

Important to Note:

1. Group of maximum 3 students.
2. **Don't use temporary files and system() function.**
3. **Only working programs will be evaluated. If there are compilation errors, it will not be evaluated.**
4. *Provide makefile for each problem.*
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. Consider a problem of counting number of hosts in a given IPv4 subnet. It is achieved by sending a broadcast datagram and counting the number of ICMP_PORT_UNREACHABLE messages. Design and implement a program that achieves this. Your program should ensure that even if multiple instances of the same program is run on a host, it gives correct results.

[12M]

P2. Network administrator after observing that bandwidth being wasted for huge downloads by several users, thought of setting up a common download server for the community. The server will act as cache as well as asynchronous file downloader. User enters a URL in the browser. URL can be referring to any website with a file extension such as doc, ppt, pdf, exe, zip etc. the browser is configured to contact the proxy or download-server. If the download-server has the file within its cache, it will send it. Otherwise it will display a message "Request from <client ip>: <url>: \n File not available. Request again after some time". The connection is closed. Meanwhile the server takes a note of the request. In the background, it initiates a HTTP request to the remote server to download the file. But the requests are made in parallel. It makes as many requests as the remote server can handle. Each request is a HTTP request with a range header. The following HTTP request is requesting the range 500-999 bytes of the file netprog102handout.pdf.

```
GET /faculty/khari/NetProg102/netprog102handout.pdf HTTP/1.1
Host: csis
Range: bytes=500-999
```

Every request can specify a maximum of 500 bytes. If the file size is such that it can't be fetched using acceptable number of requests at one time, then as soon as one request is complete, another request is made. When all requests issued are complete, the tool combines the fragments downloaded into one single file. The server makes parallel requests using multiple threads.

Use mutexes and condition variables to protect the shared data and for communication respectively. Implement a community_server.c that acts as a concurrent server for the browser-based-clients and also asynchronously fulfills those requests if it doesn't have a readily available matching file.

[12M]

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===