BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
IS F462 : Network Programming
I Semester 2014-15
Assignment-1
Weightage: 12% (36M) Due Date of Submission: 31-OCT-2015

==========

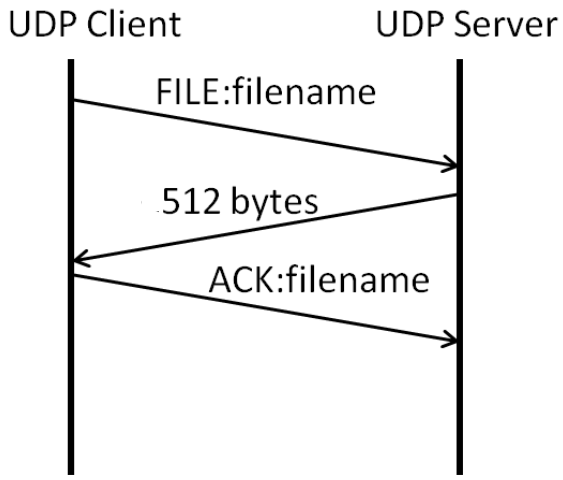
Important to Note:

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. Develop a UDP based client-server application for the following protocol. Server should support the following specifications.



- There are some files with the server. Assume that each file size is at least 512 bytes.
- Client sends "FILE:filename" to the server. Server looks up the file and if found it will send the data. If not it will send error datagram. Assume suitable error messages.
- Server sends data in blocks of 512 bytes. The last datagram will be zero length datagram to indicate end of file transfer.
- Client after receiving valid data 512 bytes or 0 bytes, it sends "ACK:filename:datagram_no" to the server.
- server accepts a port number as command-line argument. it bind at this port number.
- server supports multiple clients at the same time using process per client. When it receives a "FILE" request, it creates a new socket and binds it to a new port number. It creates a child and the child replies using this socket.
- Child sends data to the client. If it doesn't receive ACK within 2 seconds, it will resend the data. If the child receives ACK, then it will send another datagram. If ACK is not received, child will retry sending datagram for 3 times.

Implement client.c and server.c. Supply a readme.txt on how to execute your programs.

[12M]

P2. Implement a DNS query tool dnsquerytool.c for the following requirements. [4]

- It takes hostname, record type as command-line arguments. Input must be taken in this way only not through stdin.
- It prepares DNS request message as per RFC (http://www.ietf.org/rfc/rfc1035.txt). For simpler explanation you may refer to Lab7 lab files.
- It will use iterative querying mode. First trying with root server and then next level etc. Finally querying the name server for the record type.
- For every iteration it should print what is sent (section wise) and what is received from the name server (question. answer and authority sections).

[12M]

P3. Implement a strace-like program like_strace.c using ptrace() system call for the following requirements.

- It takes the program *prog* on the command-line along with its arguments.
  ./like_strace prog porg_arg1 prog_arg2
- It should print system call, its arguments and its return value for every system call called by *prog*.
- It also takes an option *-line*. When this option is given, your program should print each instruction executed by the *prog*.
- For help please refer to the https://github.com/nelhage/ministrace. This link contains partial implementation of strace. You can use this code only for understanding but not to be copied even partly.

[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===