CS 39006: Assignment 6

Non-blocking I/O

Assignment Date: 14-Feb-2019

Deadline: 14-Feb-2019 5:00 PM

Objective:

The objective of this assignment is to implement a concurrent server where multiple clients can requests for same or different services and the server serves them concurrently with the help of nonblocking I/O operations. The implementation will help you to understand the functionality of the `fcntl()` system call used for manipulating file descriptors.

Problem Statement:

The assignment statement is completely similar to the Assignment 3, except that you need to use non-blocking I/O instead of the `select()` system call. Your task is to implement a server and two clients with two different service requests. The server can receive two different service requests from the clients as follows.

1. *Request for a bag of words*, where the server will forward a set of words from a file `word.txt` one after another over a stream socket. The file `word.txt` contains a set of words, one each at every line. Once the server receives a request for the words, it reads the words from the file and forwards them one after another, each as a null terminated string. The end of service is marked with an empty string (a string only with a null character). Once the client receives that empty string, it prints the number of words received and exits. You need to set the stream socket as a non-blocking socket using `fcntl()` system call with `O_NONBLOCK` using the file status flags `F_SETFL`. Note that here you have to make the entire socket non-blocking as the socket will wait over the `accept()` call. Once you accept a new connection, you will use a `fork()` to create a child process and then the child process will handle the word transfers.
2. *Request for the IP address corresponding to a domain name*, where the client requests for the IP address of a domain, say www.iitkgp.ac.in, over a datagram socket. The server looks up for the IP address by using the system call `gethostbyname().` The server returns this IP address to the client. The client prints the IP address and exits. Here you can make the `recvfrom()` call non-blocking by setting the `MSG_DONTWAIT` flag while receiving the data from the other host.

You have to implement two clients, one corresponding to each of the services as mentioned above. Note that the client requesting bag of words works over a stream socket, whereas the client requesting for the IP address works on a datagram socket.

Submission Instruction:

You should write three C programs - `nonblockingserver.c` (contains the server program), `bowclient.c` (for bag of words requests) and `dnsclient.c` (for IP address requests). **Ideally, you should not make any change in the client codes that you had done as a part of Assignment 3.** Keep these three files in a single compressed folder (zip or tar.gz) having the name <roll number>_Assignment6.zip or <roll number>_Assignment6.tar.gz. Upload this compressed folder at Moodle course page by the deadline (24th Jan 2019 2:00 PM).