Birla Institute of Technology & Science, Pilani
Pilani Campus

I SEMESTER 2021-2022
Assignment-2

Course No.: **IS F462** **Course Title:** **Network Programming**
Deadline: **As on Canvas** **Maximum Marks:** **60M (15%)**

Note:

- Maximum of three students per group. Upload code in Canvas.
- Name your file idno1_idno2_idno3_assignment1.tar .

P1. Consider that a TCP server is designed in the following manner:

- Server creates a process pool of N processes. Each process can create a pool of T threads. Each thread accepts a connection on a shared listening descriptor.
- Whenever a process completes processing CPP number of clients across all its threads, server terminates that process and adds a new process to the pool. This is to avoid accumulation of memory leaks in the process.
- Thread processes client's request as per the chat protocol described here: Client can send three messages *JOIN <name>, LEAV* and *CHAT <targetname> <msg>*. In *JOIN* message, client sends its nickname and in *CHAT* message it sends the nickname of the person to whom it needs to be delivered and the contents. When server receives a *CHAT* message, it sends the message on the socket matching the nickname.

Server accepts port no, N, T and CPP as command line parameters. For the above requirements, write a program to implement the server.

Deliverables:

- multi_process_thread_server.c
- pdf file explaining design decisions

[20M]

P2. Consider designing and implementing an event-driven web server for the following requirements.

- Web server is a single process. In this web server HTTP request is processed in stages. Each request is divided into multiple stages such as accepting connection, reading request, processing request, sending the result etc. These stages shouldn't be long running so that latency of responses experienced by clients will be minimum.
- It uses the standard non-blocking read, write, and accept system calls on sockets, and the select/poll/epoll_wait system call to test for I/O completion
- When it has to read a file not present in memory, it uses helper processes for reading files. *mincore* call is used to check if a file is in main memory. It uses mmap() to map a file in disk to memory. When loading the file is done, it notifies the server process. There is no data transfer between child process and parent process. Number of helper processes is fixed.

- Web server also handles dynamic requests using FastCGI interface. Any request coming with an extension .cgi should go to FastCGI process and it will execute a file with that name. The output of the should be sent back to the client via web server.

Deliverables:

- event_driven_server.c
- pdf file explaining design decisions

[22M]

P3. Consider that we want to know the following characteristics of hosts. Given a network mask as the command line argument, write a program to do the following:

(a) IP addresses of all hosts which are live in the network. [Send ICMP ECHO REQUEST and read ICMP ECHO reply messages]
(b) Port numbers (both UDP/TCP) that are open in each system. [Use connect call for TCP and ICMP PORT UNREACHABLE messages for UDP]

Deliverables:

- scan.c
- pdf file explaining design decisions

[18M]

--&--