|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **National University of Computer and Emerging Sciences, Lahore Campus** | | | | |
| C:\Users\saif\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.Word\final design.jpg | **Course:** | **Computer Networks** | **Course Code:** | **CL-307** |
| **Program:** | **BS (Computer Science)** | **Semester:** | **Spring 2018** |
| **Duration:** | **120 Minutes** | **Total Marks:** | **25** |
| **Paper Date:** | **20-March-2018** | **Weight** | **25 %** |
| **Section:** | **A, B** | **Page(s):** | **4** |
| **Exam:** | **Mid-Term** | **Reg. No.** |  |
| **Instructions:** | **READ ALL INSTRUCTIONS CAREFULLY.**  1. Understanding the question paper is also part of the exam, so do not ask any clarification. Make suitable ASSUMPTIONS.  2**.** Final Submissionsshould be done in your respective section folder on **sandata/xeon/Spring2018/Computer Network**. Each question related files must be in **separate folder (Question 1, Question 2)** and all the separate folders must be in a single zip file. **Zip file must be renamed after your roll number e.g., “14L-4125”. Multiple submissions are not allowed** **(if done, only first one will be considered)**  3. Your cell phones/smart watches should be turned off and placed upside down.  4. It is your responsibility to protect your code and save it from being copied. If you don’t protect it, all matching codes will be considered copy/cheating cases. **No leniency on plagiarism.**  5. Any kind of cheat sheet/code if found in your PC will result in immediate disqualification from Final Exam and ‘**F’** as final grade in Computer Networks Lab. So make sure you delete everything from Desktop of your windows as well as Ubuntu. Also delete all the files permanently from Recycle Bin and Trash respectively for Windows and Ubuntu. Delete all files from your Z Drives before starting the exam.  6. You are immediately disqualified from the exam if:  i. You are seen talking, whispering, borrowing or looking at someone’s PC  ii. A USB is found attached to your PC  iii. You are seen using cell phone/smart watch.  iv. You are caught accessing internet. | | | |

**Part 1 TCP SOCKET PROGRAMMING (Marks: 15)**

**\*\*\*\*Submission: You have to submit your (Roll-No\_Client.c) and (Roll-No\_Server.c) files in a folder named Question 1\*\*\*\***

You have to write a TCP Socket Programming based basic Check IN / Check OUT System. Server should be in the listening state all the time. Server will have separate thread for each client. Multiple clients will connect to the server and they will check in to the server by sending their Roll Noalong with the name **(YY-AAAA-NAME-CI)** to the server. Each client will check out from the system by sending **(YY-AAAA-NAME-CO)** statement to the server. You must provide proper comments in your code wherever necessary.

**For Check In/Check Out System**:

Make sure that the client’s input string must be received inside the thread routine and then passed to the pthread\_join method for database manipulation (Add/Remove). You are not required to make multiple clients doing Check In/Check Out at the same time. You must display your server database on server in response to each request made by client to either Check In/Check Out

***Note: The input string of the client must be passed to the pthread\_join() function via pthread\_exit() call. After that you will use the received input string (in the pthread\_join()) to implement your Check In/ Check out system using steps described below. Any Student failing to do so will be awarded straightaway a zero.***

Your system must cater the following cases.

**Check in:**

* Client will send the message with roll no **“YY-AAAA-NAME-CI”** to check in to the server.
* If client checks in and it is not already present in the database then it should be added in the server database and server must send the message **“Welcome Student YY-AAAA-NAME. You are checked in.”**
* If client is already in the server database then server must send the message **“You are already checked in.”**

**Check out:**

* Client will send the message with roll no **“YY-AAAA-NAME-CO”** to check out from the server.
* If client is present in the database then it should be removed from the server database and server must send the message **“Goodbye Student YY-AAAA-NAME. You are checked out.”**
* If client is not in the server database and tries to check out then server must send the message **“You didn’t check in today. Contact System Administrator.”**

**Note: Make suitable assumptions wherever necessary. Write proper comments in your code. Your 50% Marks will be deducted if your code does not compile successfully. Properly manipulate your Server Database after adding/removing client.**

**Multithreading Socket Programming Syntax**

**Socket Programming:**

* int socket(int domain, int type, int protocol);
* domain = AF\_INET, AF\_INET6
* type = SOCK\_STREAM, SOCK\_DGRAM
* protocol = 0(preferred), IPPROTO\_TCP, IPPROTO\_UDP, IPPROTO\_ICMP
* int bind(int socket, struct sockaddr \*name, int namelen)
* struct sockaddr\_in {

short sin\_family; // e.g. AF\_INET, AF\_INET6

unsigned short sin\_port; // e.g. htons(3490)

struct in\_addr sin\_addr; // see struct in\_addr, below

char sin\_zero[8]; // zero this

};

* struct in\_addr {

unsigned long s\_addr; // load with inet\_addr()

};

* struct sockaddr {

unsigned short sa\_family; // address family, AF\_xxx

char sa\_data[14]; // 14 bytes of protocol address

};

* int listen(int socket, int backlog)
* int accept(int socket, struct sockaddr \*addr, int \*addrlen)
* int connect(int socket, struct sockaddr \*addr, int addrlen)
* int send(int socket, const void \*buf, int buflen, int flags);
* int recv(int socket, void \*buf, int buflen, int flags);
* int sendto(int socket, const void \*buf, int buflen, int flags, struct sockaddr\* to, int tolen);
* int recvfrom(int socket, void \*buf, int buflen, int flags, struct sockaddr\* from, int \*fromlen);
* int close(int socket)

**Multithreading:**

* int pthread\_create(pthread\_t \*thread, pthread\_attr\_t \*attr, void \*(\*start\_routine)(void \*), void \*arg);
* void pthread\_exit(void \*value\_ptr);
* int pthread\_join(pthread\_t thread, void \*\*value\_ptr);

**Part 2 Wireshark Packet Analyzer (Marks: 10)**

**\*\*\*\*Submission: You have to submit your (Roll-No.docx) word file in a folder named Question 2. You should provide proper explanation along with screen shots. \*\*\*\***

**Part 1: \*\*\*Wireshark Session: Capture 1\*\*\* [4]**

1. Filter out all TCP packets which are either going to or coming from IP: 128.119.245.12 on Port: 80**.** Write your filter as answer
2. If we take the first filtered packet as Packet No. 1, with each successive packet increasing one in number, then what is the request made in Packet No. 20 and what is the response from the server?
3. How many data containing TCP Segments are present between the request made and response received for the packet in question 2**?**
4. What is the content (data) length in bits returned by the server in response to the request made in Packet No. 28 taking first filtered packet as Packet No. 1?

**Part 2: \*\*\* Wireshark Session: Capture 2\*\*\* [4]**

1. FTP is an application protocol and uses TCP as a Transport Layer Protocol. Filter out all the FTP packets by applying the filter on specific ports used by File Transfer Protocol to make a connection or transfer the data. Write your filter as the answer
2. In the packets filtered, client asks for data two times from the server on a specific port and IP Address because of which server opens a data connection on the corresponding socket and sends the data to client on that newly opened socket on the client side. Keeping the sequence of first two client requests with regards to IP and Port Number, what will be the request made by client to get the data for the third time from the server?
3. In regard to the third request made by client in question 6, server will send the data to the client on which Port Number (client side), Port Number (server side), IP address (client side) and IP address (server side)?
4. Apply a filter to get all the UDP packets which are directed towards Port Number 53 either by client or by server. What filter did you apply? If we take the first filtered packet as Packet No. 1 with each successive packet increasing one in number, then what will be the acknowledgement number of the DNS request made in Packet No. 12? Also tell the sequence number of the packet in which the request is made?

**Part 3: \*\*\* Wireshark Session: Capture 3\*\*\* [2]**

1. In which packet (write packet number) we have received the acknowledgement for the TCP Segment being carried in the packet number 13? What is the acknowledgement number and sequence number in the packet received? Show clear calculations how the server has sent the corresponding acknowledgement number?
2. In Packet No. 170, what is the value of Sequence Number? How will you explain the value of Sequence Number?