Muhammad Abdullah

SE(6A) | 19F-0916

Computer Networks Lab

Lab 5 Concurrent Socket Programming

**CONCURRENT TCP CLIENT SERVER COMMUNICATION**

**Server-Side Code:**

**#include <iostream>**

**#include <sys/socket.h>**

**#include <netinet/in.h>**

**#include <stdlib.h>**

**#include <stdio.h>**

**#include <unistd.h>**

**#include <string.h>**

**#include <sys/types.h>**

**#include <sys/wait.h>**

**#include <sys/user.h>**

**int client\_count = 0;**

**using namespace std;**

**int main ()**

**{**

**struct sockaddr\_in address , cliaddress;**

**int obj\_server, sock, reader, binder;**

**int selected = 1;**

**int address\_length = sizeof(address);**

**char buffer[1024] = {0};**

**char \*message = "Hey Client! I'm Fine. What about You?";**

**obj\_server = socket ( AF\_INET, SOCK\_STREAM, 0);**

**setsockopt(obj\_server, SOL\_SOCKET, SO\_REUSEADDR, &selected, sizeof (selected));**

**address.sin\_family = AF\_INET;**

**address.sin\_addr.s\_addr = INADDR\_ANY;**

**address.sin\_port = htons(8080);**

**binder = bind(obj\_server, ( struct sockaddr \* )&address,sizeof(address));**

**pid\_t pid;**

**listen ( obj\_server, 8);**

**while (true)**

**{**

**sock = accept(obj\_server, (struct sockaddr \*)&address, (socklen\_t\*)&address\_length);**

**if ((pid = fork()) == 0)**

**{**

**client\_count++;**

**reader = read(sock, buffer, 1024);**

**cout << (buffer) << endl;**

**send(sock , message, strlen(message) , 0 );**

**cout << "Server : Message has been sent to Client " << client\_count << " !" << endl;**

**close(sock);**

**break;**

**}**

**else if (pid > 0)**

**{**

**client\_count++;**

**close(sock);**

**continue;**

**}**

**else**

**{**

**cout<<"Unsuccesfull Child Creation!"<<endl;**

**}**

**}**

**return 0;**

**}**

**Client1-Side Code:**

**#include <iostream>**

**#include <stdio.h>**

**#include <sys/socket.h>**

**#include <arpa/inet.h>**

**#include <unistd.h>**

**#include <string.h>**

**using namespace std;**

**int main ()**

**{**

**struct sockaddr\_in sAddress;**

**int socketInstance = 0, reader;**

**char \*message = "Hey Server! How Are You?";**

**char buffer[1024] = {0};**

**socketInstance = socket (AF\_INET, SOCK\_STREAM, 0);**

**sAddress.sin\_family = AF\_INET;**

**sAddress.sin\_port = htons(8080);**

**inet\_pton (AF\_INET, "127.0.0.1", &sAddress.sin\_addr);**

**connect(socketInstance, (struct sockaddr \*)&sAddress, sizeof(sAddress));**

**send (socketInstance , message , strlen(message) , 0);**

**cout << endl << ( "Client 1: Message has been sent !") << endl;**

**reader = read ( socketInstance, buffer, 1024 );**

**cout << ( buffer ) << endl;**

**close(socketInstance);**

**return 0;**

**}**

**Client2-Side Code:**

**#include <iostream>**

**#include <stdio.h>**

**#include <sys/socket.h>**

**#include <arpa/inet.h>**

**#include <unistd.h>**

**#include <string.h>**

**using namespace std;**

**int main ()**

**{**

**struct sockaddr\_in sAddress;**

**int socketInstance = 0, reader;**

**char \*message = "Hey Server! How Are You?";**

**char buffer[1024] = {0};**

**socketInstance = socket (AF\_INET, SOCK\_STREAM, 0);**

**sAddress.sin\_family = AF\_INET;**

**sAddress.sin\_port = htons(8080);**

**inet\_pton (AF\_INET, "127.0.0.1", &sAddress.sin\_addr);**

**connect(socketInstance, (struct sockaddr \*)&sAddress, sizeof(sAddress));**

**send (socketInstance , message , strlen(message) , 0);**

**cout << endl << ( "Client 2: Message has been sent !") << endl;**

**reader = read ( socketInstance, buffer, 1024 );**

**cout << ( buffer ) << endl;**

**close(socketInstance);**

**return 0;**

**}**

Concurrent TCP/IP Communication Screenshot:

**Graphical user interface

Description automatically generated**

**CONCURRENT UDP CLIENT SERVER COMMUNICATION**

**Note: UDP cannot express the concurrent behavior because it is connection less protocol. It only sends message, and that message is received by a server. It can or cannot be receive by server because UDP doesn’t requires to form a connection.**

**Apart from that, I’ve tried to use fork() call for this task but clearly it doesn’t have any advantage over here.**

**Server-Side Code:**

**#include <iostream>**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <unistd.h>**

**#include <string.h>**

**#include <sys/types.h>**

**#include <sys/socket.h>**

**#include <arpa/inet.h>**

**#include <netinet/in.h>**

**#include <sys/wait.h>**

**#include <sys/user.h>**

**int client\_count = 0;**

**using namespace std;**

**int main ()**

**{**

**struct sockaddr\_in sAddress, cAddress;**

**int obj\_server, sock, reader;**

**sock = sizeof(cAddress);**

**char buffer[1024] = {0};**

**char \*message = "Hey UDP Client! I'm Fine. What about You?";**

**pid\_t pid;**

**obj\_server = socket(AF\_INET, SOCK\_DGRAM, 0);**

**memset(&sAddress, 0, sizeof(sAddress));**

**memset(&cAddress, 0, sizeof(cAddress));**

**sAddress.sin\_family = AF\_INET;**

**sAddress.sin\_addr.s\_addr = INADDR\_ANY;**

**sAddress.sin\_port = htons(8080);**

**bind(obj\_server, (struct sockaddr \*)&sAddress, sizeof(sAddress));**

**while (true)**

**{**

**reader = recvfrom(obj\_server,(char \*)buffer,1024,MSG\_WAITALL, (struct sockaddr \*) &cAddress,(socklen\_t\*)&sock);**

**if ((pid = fork()) == 0)**

**{**

**cout << (buffer) << endl;**

**sendto(obj\_server, (char \*)message, strlen(message),MSG\_CONFIRM,(struct sockaddr \*)&cAddress,sock);**

**cout << ("Server : Message has been sent !") << endl;**

**break;**

**}**

**else if (pid > 0)**

**{**

**client\_count++;**

**continue;**

**}**

**else**

**{**

**cout<<"Unsuccesfull Child Creation!"<<endl;**

**}**

**return 0;**

**}**

**}**

**Client1-Side Code:**

**#include <iostream>**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <unistd.h>**

**#include <string.h>**

**#include <sys/types.h>**

**#include <sys/socket.h>**

**#include <arpa/inet.h>**

**#include <netinet/in.h>**

**using namespace std;**

**int main ()**

**{**

**struct sockaddr\_in sAddress;**

**int socketInstance, sock, reader;**

**sock = sizeof(sAddress);**

**char buffer[1024] = {0};**

**char \*message = "Hey UDP Server! How Are You? Client 1";**

**socketInstance = socket(AF\_INET, SOCK\_DGRAM, 0);**

**memset(&sAddress, 0, sizeof(sAddress));**

**sAddress.sin\_family = AF\_INET;**

**sAddress.sin\_addr.s\_addr = INADDR\_ANY;**

**sAddress.sin\_port = htons(8080);**

**sendto(socketInstance, (char \*)message, strlen(message),MSG\_CONFIRM, ( struct sockaddr \*) &sAddress,sock);**

**reader = recvfrom(socketInstance,(char \*)buffer,1024, MSG\_WAITALL, (struct sockaddr \*) &sAddress,(socklen\_t\*)&sock);**

**cout << (buffer) << endl;**

**cout << ("Client 1: Message has been sent !") << endl;**

**close(socketInstance);**

**return 0;**

**}**

**Client2-Side Code:**

**#include <iostream>**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <unistd.h>**

**#include <string.h>**

**#include <sys/types.h>**

**#include <sys/socket.h>**

**#include <arpa/inet.h>**

**#include <netinet/in.h>**

**using namespace std;**

**int main ()**

**{**

**struct sockaddr\_in sAddress;**

**int socketInstance, sock, reader;**

**sock = sizeof(sAddress);**

**char buffer[1024] = {0};**

**char \*message = "Hey UDP Server! How Are You? Client 2";**

**socketInstance = socket(AF\_INET, SOCK\_DGRAM, 0);**

**memset(&sAddress, 0, sizeof(sAddress));**

**sAddress.sin\_family = AF\_INET;**

**sAddress.sin\_addr.s\_addr = INADDR\_ANY;**

**sAddress.sin\_port = htons(8080);**

**sendto(socketInstance, (char \*)message, strlen(message),MSG\_CONFIRM, ( struct sockaddr \*) &sAddress,sock);**

**reader = recvfrom(socketInstance,(char \*)buffer,1024, MSG\_WAITALL, (struct sockaddr \*) &sAddress,(socklen\_t\*)&sock);**

**cout << (buffer) << endl;**

**cout << ("Client 2: Message has been sent !") << endl;**

**close(socketInstance);**

**return 0;**

**}**

Communication Screenshot:

**Graphical user interface

Description automatically generated**