**ШИНЖЛЭХ УХААН ТЕХНОЛОГИЙН ИХ СУРГУУЛЬ**

**Мэдээлэл холбооны технологийн сургууль**

****

**ЛАБОРАТОРИЙН АЖЛЫН**

**ТАЙЛАН - 5**

**Сүлжээний програмчлал I (F.NS351)**

**2022-2023 оны хичээлийн жилийн хавар**

**Лабораторийн ажлын нэр: Си програмчлалын хэлийг судлах-2**

**Хичээл заасан багш:** Б.Мөнхбаяр /Ph.D/

**Лабораторийн ажил гүйцэтгэсэн:** Уламбаяр /В200970314/

**Лабораторийн цаг:** 4-2

Улаанбаатар хот

2023 он

**Гарчиг**

Лабораторийн ажлын зорилго............................................................................3

Даалгавар….........................................................................................................4

Дүгнэлт................................................................................................................9

Ашигласан материал..........................................................................................9

**ЛАБОРАТОРИЙН АЖЛЫН ЗОРИЛГО, ЗОРИЛТ**

Энэхүү лабораторийн зорилго нь дараах шаардлагатай мэдлэг чадварыг олж авах явдал юм. Үүнд:

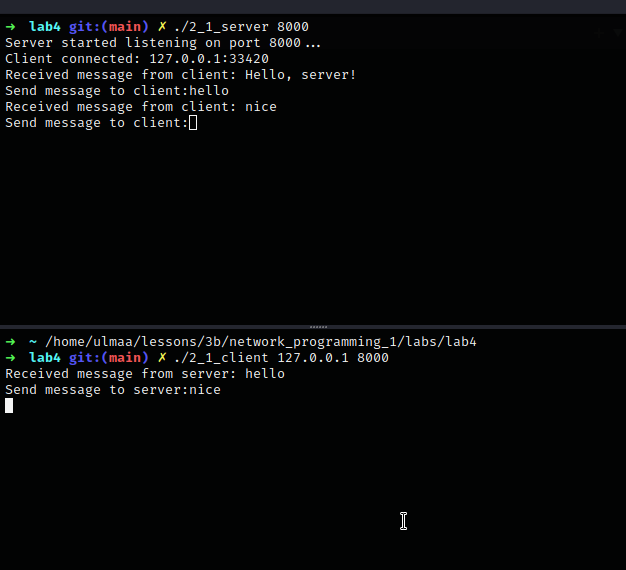
* socket programming
* client and server programming

**ЛАБОРАТОРИЙН АЖИЛ**

**Даалгавар-1:**

TCP Stream socket үүсгэнэ үү. Сервер, клент хоёр хоорондоо холболт

тогтоодог байна



**Даалгавар-2:**

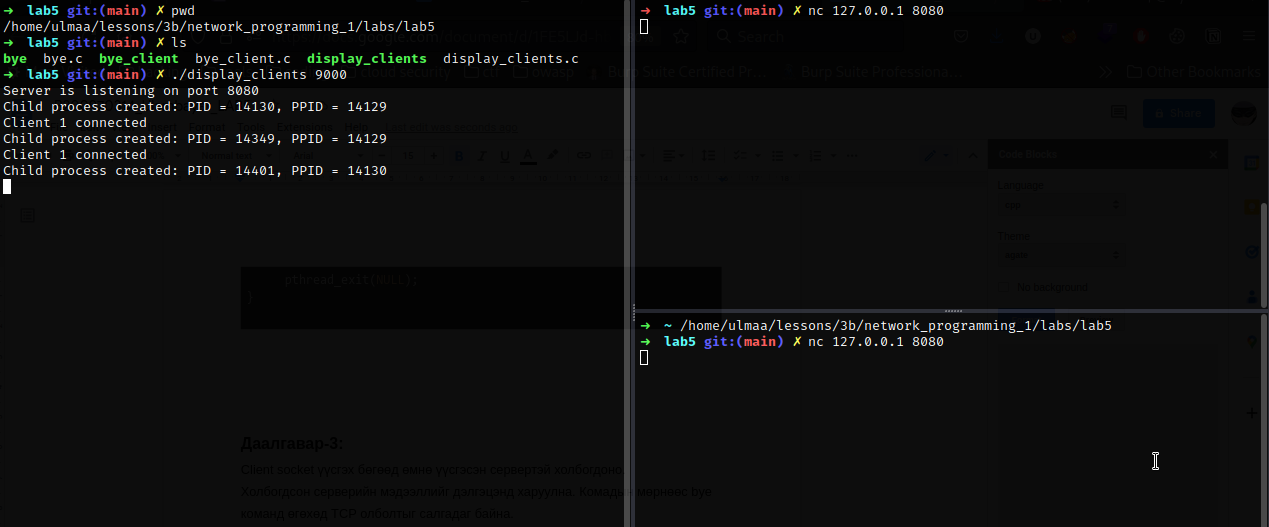
TCP сервер үүсгэнэ үү. Сервер нь ирсэн хүсэлтэнд шинэ процесс үүсгэж

хариулах ба холбогдсон хэрэглэгчийн жагсаалтыг дэлгэцэнд харуулна. Сервер

шеллийн горимд ажиллах бөгөөд холбогдсон хэрэглэгчийн жагсаалтыг дэлгэцэнд

хэвлэдэг байна. Сервер холболтыг салгахгүй клент холболтыг салгадаг байна

| #include <stdio.h> #include <stdlib.h> #include <unistd.h> #include <pthread.h> #include <sys/socket.h> #include <arpa/inet.h> #include <string.h> #include <netinet/in.h> #define MAX\_CLIENTS 10  void \*handle\_client(void \*arg);  int main(int argc, char \*argv[]) {  int server\_fd, client\_fd;  struct sockaddr\_in address;  int opt = 1;  int addrlen = sizeof(address);  pthread\_t threads[MAX\_CLIENTS];   // Create socket  if ((server\_fd = socket(AF\_INET, SOCK\_STREAM, 0)) == 0) {  perror("Socket creation error");  exit(EXIT\_FAILURE);  }   // Set socket options  if (setsockopt(server\_fd, SOL\_SOCKET, SO\_REUSEADDR | SO\_REUSEPORT, &opt, sizeof(opt))) {  perror("Setsockopt error");  exit(EXIT\_FAILURE);  }   // Bind socket  address.sin\_family = AF\_INET;  address.sin\_addr.s\_addr = INADDR\_ANY;  address.sin\_port = htons(8080);  if (bind(server\_fd, (struct sockaddr \*)&address, sizeof(address)) < 0) {  perror("Bind error");  exit(EXIT\_FAILURE);  }   // Listen for connections  if (listen(server\_fd, MAX\_CLIENTS) < 0) {  perror("Listen error");  exit(EXIT\_FAILURE);  }   printf("Server is listening on port 8080\n");   // Accept client connections and create a thread for each  for (int i = 0; i < MAX\_CLIENTS; i++) {  pid\_t pid = fork();  if (pid == 0) {  // This is the child process  printf("Child process created: PID = %d, PPID = %d\n", getpid(), getppid());  }  if ((client\_fd = accept(server\_fd, (struct sockaddr \*)&address, (socklen\_t \*)&addrlen)) < 0) {  perror("Accept error");  exit(EXIT\_FAILURE);  }  printf("Client %d connected\n", i+1);  pthread\_create(&threads[i], NULL, handle\_client, (void \*)&client\_fd);  }   // Join threads  for (int i = 0; i < MAX\_CLIENTS; i++) {  pthread\_join(threads[i], NULL);  }   return 0; }  void \*handle\_client(void \*arg) {  int client\_fd = \*(int \*)arg;  char buffer[1024] = {0};  int valread;   // Read from client  valread = read(client\_fd, buffer, 1024);  printf("Received: %s\n", buffer);   // Write to client  char \*message = "Hello from server";  send(client\_fd, message, strlen(message), 0);   printf("Client disconnected\n");   close(client\_fd);  pthread\_exit(NULL); } |
| --- |

****

**Даалгавар-3:**

Client socket үүсгэх бөгөөд өмнө үүсгэсэн сервертэй холбогдоно.

Холбогдсон серверийн мэдээллийг дэлгэцэнд харуулна. Комадын мөрнөөс bye

команд өгөхөд TCP олболтыг салгадаг байна.

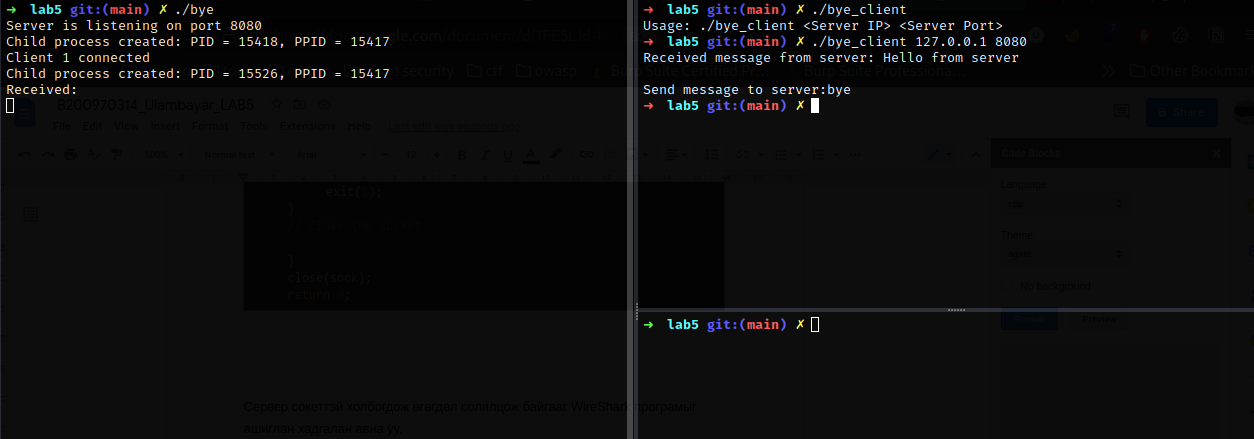
Server code

| #include <stdio.h> #include <stdlib.h> #include <unistd.h> #include <pthread.h> #include <sys/socket.h> #include <arpa/inet.h> #include <string.h> #include <netinet/in.h> #define MAX\_CLIENTS 10  void \*handle\_client(void \*arg);  int main(int argc, char \*argv[]) {  int server\_fd, client\_fd;  struct sockaddr\_in address;  int opt = 1;  int addrlen = sizeof(address);  pthread\_t threads[MAX\_CLIENTS];   // Create socket  if ((server\_fd = socket(AF\_INET, SOCK\_STREAM, 0)) == 0) {  perror("Socket creation error");  exit(EXIT\_FAILURE);  }   // Set socket options  if (setsockopt(server\_fd, SOL\_SOCKET, SO\_REUSEADDR | SO\_REUSEPORT, &opt, sizeof(opt))) {  perror("Setsockopt error");  exit(EXIT\_FAILURE);  }   // Bind socket  address.sin\_family = AF\_INET;  address.sin\_addr.s\_addr = INADDR\_ANY;  address.sin\_port = htons(8080);  if (bind(server\_fd, (struct sockaddr \*)&address, sizeof(address)) < 0) {  perror("Bind error");  exit(EXIT\_FAILURE);  }   // Listen for connections  if (listen(server\_fd, MAX\_CLIENTS) < 0) {  perror("Listen error");  exit(EXIT\_FAILURE);  }   printf("Server is listening on port 8080\n");   // Accept client connections and create a thread for each  for (int i = 0; i < MAX\_CLIENTS; i++) {  pid\_t pid = fork();  if (pid == 0) {  // This is the child process  printf("Child process created: PID = %d, PPID = %d\n", getpid(), getppid());  }  if ((client\_fd = accept(server\_fd, (struct sockaddr \*)&address, (socklen\_t \*)&addrlen)) < 0) {  perror("Accept error");  exit(EXIT\_FAILURE);  }  printf("Client %d connected\n", i+1);  pthread\_create(&threads[i], NULL, handle\_client, (void \*)&client\_fd);  }   // Join threads  for (int i = 0; i < MAX\_CLIENTS; i++) {  pthread\_join(threads[i], NULL);  }   return 0; }  void \*handle\_client(void \*arg) {  int client\_fd = \*(int \*)arg;  char buffer[1024] = {0};  int valread;  char \*message = "Hello from server\n";  if (send(client\_fd, message, strlen(message), 0) == -1) {  perror("Send failed");  exit(1);  }  // Read from client  valread = read(client\_fd, buffer, 1024);  char bye[3] = "bye";  printf("Received: %s\n", buffer);  if (strncmp(buffer,bye,3) == 0) {  printf("Client disconnected\n");  close(client\_fd);  pthread\_exit(NULL);  } else {  printf("wtf");  }  char message[1024];  scanf("%s",&message);    } |
| --- |

Client code

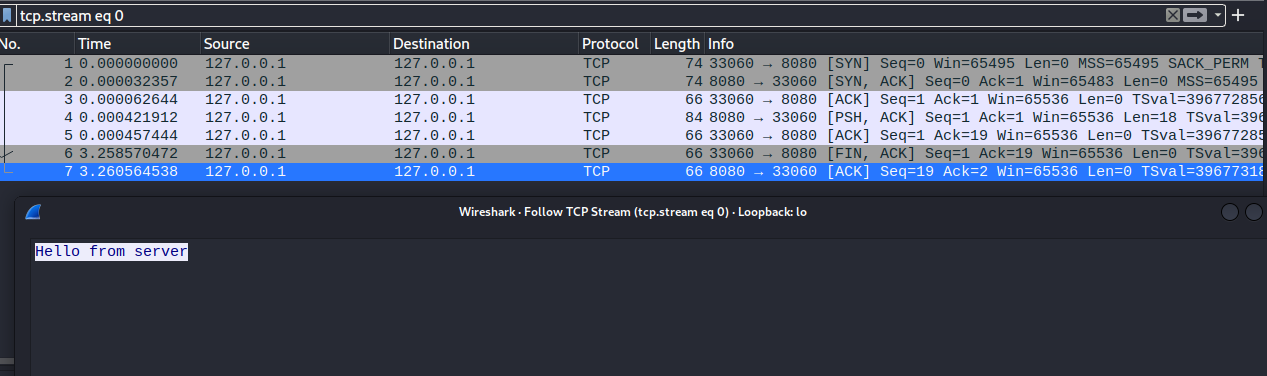
| #include <stdio.h> #include <stdlib.h> #include <string.h> #include <unistd.h> #include <sys/socket.h> #include <arpa/inet.h>  int main(int argc, char \*argv[]) {  // Check for correct number of arguments  if (argc != 3) {  printf("Usage: %s <Server IP> <Server Port>\n", argv[0]);  exit(1);  }   // Create a socket  int sock = socket(AF\_INET, SOCK\_STREAM, 0);  if (sock == -1) {  perror("Socket creation failed");  exit(1);  }   // Set up the server address  struct sockaddr\_in server\_addr;  memset(&server\_addr, 0, sizeof(server\_addr));  server\_addr.sin\_family = AF\_INET;  server\_addr.sin\_addr.s\_addr = inet\_addr(argv[1]);  server\_addr.sin\_port = htons(atoi(argv[2]));   // Connect to the server  if (connect(sock, (struct sockaddr \*)&server\_addr, sizeof(server\_addr)) == -1) {  perror("Connection failed");  exit(1);  }   // Receive a message from the server  while (1) {  char buffer[1024];  int num\_bytes = recv(sock, buffer, sizeof(buffer), 0);  if (num\_bytes == -1) {  perror("Receive failed");  exit(1);  }   // Print the message received from the server  buffer[num\_bytes] = '\0';  printf("Received message from server: %s\n", buffer);  printf("Send message to server:");  char message[1024];  scanf("%s",&message);  if (strncmp(message,"bye",3) == 0) {  close(sock);  exit(1);  }  if (send(sock, message, strlen(message), 0) == -1) {  perror("Send failed");  exit(1);  }  // Close the socket    }  close(sock);  return 0; |
| --- |

Result



Сервер сокеттэй холбогдож өгөгдөл солилцож байгааг WireShark програмыг

ашиглан хадгалан авна уу.

****

**Дүгнэлт**

Энэхүү лабораторийн ажлаар “Сүлжээний програмчлал 1” хичээлийн суурь ойлголт болох socket programming талаар судалж мэдэж авсан ба өгөгдсөн бодлогуудын программыг бичэж гүйцэтгэсэн билээ.

**Ашигласан материал**

[1] “F.NS351 Сүлжээний програмчлал 4 хичээлийн лекцийн агуулга” 2022 он, Улаанбаатар хот, Монгол Улс

[2] “laboratory 5.pdf ” - Лабораторийн ажлын заавар

[3] https://www.tutorialspoint.com/cprogramming/

[4] https://www.tutorialspoint.com/cprogramming/c\_basic\_syntax.htm