

# به نام خدا

تمرین کامپیوتری چهارم

استاد : دکتر یزدانی

اعضای گروه :

علی دارابی – 810100264

حسام رمضانیان – 810100248

## توضیح کدهای قسمت اول :

کدها شامل سه بخش اصلی هستند: کلاینت، سرور و پکت.

### 1. فایل Client.cpp و Client.hpp

`Client::Client(const char* ip, uint16_t port) :`

- یک سوکت ایجاد می‌کند. آدرس و پورت سرور را تنظیم می‌کند.

`void Client::sendTo(int client_socket, const char* message) :`

- داده‌ها را به فرم پکت تبدیل کرده و از طریق سوکت ارسال می‌کند.

`void Client::start() :`

- به سرور متصل می‌شود. تبادل اولیه با سرور را انجام می‌دهد. داده‌ها را ارسال می‌کند و سپس ارتباط را می‌بندد.

`void Client::perform_handshake() :`

- پیام‌های SYN، SYN-ACK و ACK را برای برقراری ارتباط با سرور ارسال و دریافت می‌کند.

`void Client::send_data() :`

- پیام DEFAULT MESSAGE را ارسال کرده و پاسخ سرور را دریافت می‌کند.

## 2. فایل Packet.cpp و Packet.hpp

Packet::Packet(int id, const char\* data) :

- شماره شناسه و داده‌های پکت را تنظیم می‌کند.

const char\* Packet::encode(const Packet& packet) :

- داده‌ها را به رشته‌ای با فرمت خاص تبدیل می‌کند.

Packet Packet::decode(const char\* str) :

- رشته ورودی را به یک پکت با شماره شناسه و داده‌ها تبدیل می‌کند.

Packet Packet::decode(const char str) :

- شماره شناسه و داده‌های پکت را چاپ می‌کند.

## 3. فایل Server.cpp و Server.hpp

Server::Server(const char\* ip, uint16\_t port) :

- یک سوکت ایجاد می‌کند. آدرس و پورت سرور را تنظیم و شروع به گوش دادن برای ارتباطات می‌کند.

void Server::sendTo(int client\_socket, const char message) :

- پیام را به فرم پکت تبدیل کرده و ارسال می‌کند.

`void Server::start() :`

- به انتظار ارتباطات کلاینت‌ها می‌ماند و هر کلاینت را در یک رشته جدید مدیریت می‌کند.

`void Server::handle_client(int client_socket) :`

- پیام‌های ورودی کلاینت را دریافت و پردازش می‌کند. تبادل اولیه با کلاینت را انجام داده و داده‌ها را مدیریت می‌کند.

#### 4. فایل `main.cpp`

- اگر حالت `s` انتخاب شود، سرور شروع به کار می‌کند. در غیر این صورت، سه کلاینت ایجاد شده و به سرور متصل می‌شوند تا داده‌ها را ارسال کنند.