

# Exercise 7

---

## UDP-Client/Server

In this exercise we develop a UDP Server that can be used to send data read from different files to a UDP client that asks for this. The programming language used in this exercise is C++.

## Server app

---

A UDP server needs to have a socket which listens to incoming connections. Client will be sending telegrams towards the server requesting something, and then the server will send back a response.

In the code below you see the "main" action of the server program.

```
int udp_serv::run_serv(){
    bool done = false;
    int recvBytes;

    cout << "Starting server\n";
    while(!done){
        recvBytes = recvfrom(udp_socket, &recvBuf, sizeof(recvBuf), 0, (sockaddr*)&sother, &sother_len);
        if(recvBytes < 0){
            perror("error reading recv\n");
            done = true;
        }
        else{
            // Do something
            if(checkCmd()){
                std::cout << "Sent data\n";
                clearBufs();
            }
            else{
                std::cout << "Unrecognized command.\n";
            }
        }
    }
}
return 0;
```

In this snippet below, you see the functionality for handling and responding to commands read from the incoming socket.

```
bool udp_serv::checkCmd(){
    std::cout << "cmd : " << recvBuf << "\n";
    for(int i = 0; i < 4; i++){
```

```

        if(strcoll(accCmd[i], recvBuf) == 0){
            if(accCmd[i] == "U" || accCmd[i] == "u"){
                cpyUptime();
                sendto(udp_socket, &sendBuf, sizeof(sendBuf), 0, (sockaddr*)
&sother, sother_len);
                return true;
            }
            else if(accCmd[i] == "L" || accCmd[i] == "l"){
                cpyLoadAvg();
                sendto(udp_socket, &sendBuf, sizeof(sendBuf), 0, (sockaddr*)
&sother, sother_len);
                return true;
            }
        }
    }
    return false;
}

```

Below is a code snippet showing one of the file reading functions used to complete the forementioned functionality

```

int udp_serv::cpyLoadAvg(){
    int fd = open("/proc/loadavg", O_RDONLY);
    int readBytes = read(fd, &sendBuf, sizeof(sendBuf));
    if(readBytes < 0){
        perror("Couldn't read from loadavg\n");
        return -1;
    }
    close(fd);
    return 0;
}

```

## Client App

---

The client app is also very similar in terms of the setup stuff, where again the socket is set to SOCK\_DGRAM. Furthermore there is now no need to do the binding manually, since the use of sendto() automatically binds the socket.

## Results

---

the results of this exercise can be watched in the attached UDPtest.wav video.