

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <sys/types.h>
5  #include <sys/socket.h>
6  #include <netinet/in.h>
7  #include <arpa/inet.h>
8  #include "Practical.h"
9  #include <unistd.h>
10 #include <time.h>
11
12 static const int MAXPENDING = 5; // Maximum outstanding connection requests
13
14 int main(int argc, char *argv[]) {
15     time_t ticks;
16     char sendbuffer[BUFSIZE]; // Buffer for sending data to the client
17
18     if (argc != 2) // Test for correct number of arguments
19         DieWithUserMessage("Parameter(s)", "<Server Port>");
20
21     in_port_t servPort = atoi(argv[1]); // First arg: local port
22
23     // Create socket for incoming connections
24     int servSock; // Socket descriptor for server
25     if ((servSock = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP)) < 0)
26         DieWithSystemMessage("socket() failed");
27
28     // Construct local address structure
29     struct sockaddr_in servAddr; // Local address
30     memset(&servAddr, 0, sizeof(servAddr)); // Zero out structure
31     servAddr.sin_family = AF_INET; // IPv4 address family
32     servAddr.sin_addr.s_addr = htonl(INADDR_ANY); // Any incoming interface
33     servAddr.sin_port = htons(servPort); // Local port
34
35     // Bind to the local address
36     if (bind(servSock, (struct sockaddr*) &servAddr, sizeof(servAddr)) < 0)
37         DieWithSystemMessage("bind() failed");
38
39     // Mark the socket so it will listen for incoming connections
40     if (listen(servSock, MAXPENDING) < 0)
41         DieWithSystemMessage("listen() failed");
42
43     for (;;) { // Run forever
44
45         // Wait for a client to connect
46         int clntSock = accept(servSock, (struct sockaddr *) NULL, NULL);
47         if (clntSock < 0)
48             DieWithSystemMessage("accept() failed");
49
50         // clntSock is connected to a client!
51         snprintf(sendbuffer, sizeof(sendbuffer), "%.24s\r\n", ctime(&ticks)); //Create data and time string in outgoing buffer
52         ssize_t numBytesSent = send(clntSock, sendbuffer, strlen(sendbuffer), 0); //Send date and time string to the client
53         if (numBytesSent < 0)
54             DieWithSystemMessage("send() failed");
55
56         close(clntSock); // Close client socket
57
58     }
59     // NOT REACHED
60 }

```