

```

Command Prompt

Line#: 350 Count#: 11      GetExitCodeProcess(processes[i].hProcess, &exitCode);
Line#: 353 Count#: 22      string kernelTimeString = to_string(kTime.wHour) + ":" + to_string(kTime.wMinute) + ":" + to_string(kTime.wSecond) + ":" + to_string(u
(kTime.wMilliseconds);
Line#: 354 Count#: 27      string userTimeString = to_string(uTime.wHour) + ":" + to_string(uTime.wMinute) + ":" + to_string(uTime.wSecond) + ":" + to_string(u
Time.wMilliseconds);
Line#: 356 Count#: 2       if (exitCode != 0)
Line#: 358 Count#: 6       {
Line#: 359 Count#: 12       BatchInputError error;
Line#: 360 Count#: 6       error.launchGroup = processesInfo[i].launchGroup;
Line#: 361 Count#: 6       error.parameters = processesInfo[i].parameters;
Line#: 362 Count#: 2       error.kernelTime = kernelTimeString;
Line#: 363 Count#: 8       error.exitCode = to_string(exitCode);
Line#: 364 Count#: 4       error.userTime = userTimeString;
Line#: 365 Count#: 4       errors.push_back(error);
Line#: 370 Count#: 14       cout << left << setw(15) << processesInfo[i].launchGroup << setfill(' ');
Line#: 371 Count#: 8       cout << left << setw(30) << kernelTimeString << setfill(' ');
Line#: 372 Count#: 8       cout << setw(30) << userTimeString << setfill(' ');
Line#: 373 Count#: 8       cout << setw(15) << exitCode << setfill(' ');
Line#: 374 Count#: 10      cout << setw(35) << processesInfo[i].programName << setfill(' ');
Line#: 375 Count#: 9       cout << setw(15) << processesInfo[i].parameters << endl;
Line#: 378 Count#: 5       CloseHandle(processes[i].hProcess);
Line#: 379 Count#: 3       CloseHandle(processes[i].hThread);

./Week 1/MinimalBoostTest/MinimalBoostTest/MinimalBoostTest.cpp
Line#: 1 Count#: 1 #define BOOST_TEST_MODULE TestDemo
Line#: 2 Count#: 11 #include <boost/test/unit_test.hpp>
Line#: 4 Count#: 2 BOOST_AUTO_TEST_CASE(intro)
Line#: 8 Count#: 13 // Comment out the following line to get a clean run

Total found: 10198

C:\Users\tiber\Desktop\Files\School\Semester 6\C++>ultragrep . [count] .cpp

```

```

Command Prompt - ultragrep -v . [count] .cpp

Matched [1] : ./Sockets/TCPServer/TCPServer.cpp [13] // initialize WSA
Matched [2] : ./Sockets/TCPServer/TCPServer.cpp [14] WSADATA wsaData;
Matched [7] : ./Sockets/TCPServer/TCPServer.cpp [15] int iResult = WSAStartup(MAKEWORD(2, 2), &wsaData);
Matched [2] : ./Sockets/TCPServer/TCPServer.cpp [16] if (iResult != 0) {
Matched [6] : ./Sockets/TCPServer/TCPServer.cpp [17] cerr << "WSAStartup failed: " << iResult << endl;
Matched [2] : ./Sockets/TCPServer/TCPServer.cpp [18] return EXIT_FAILURE;
Matched [5] : ./Sockets/TCPServer/TCPServer.cpp [21] // Create the TCP socket
Matched [6] : ./Sockets/TCPServer/TCPServer.cpp [22] SOCKET hSocket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
Matched [2] : ./Sockets/TCPServer/TCPServer.cpp [24] // Create the server address
Matched [2] : ./Sockets/TCPServer/TCPServer.cpp [25] sockaddr_in serverAddress = { 0 };
Matched [4] : ./Sockets/TCPServer/TCPServer.cpp [27] serverAddress.sin_port = htons(PORT);
Matched [1] : ./Sockets/TCPServer/TCPServer.cpp [28] serverAddress.sin_addr.s_addr = inet_addr("127.0.0.1");
Matched [4] : ./Sockets/TCPServer/TCPServer.cpp [30] // bind the socket
Matched [4] : ./Sockets/TCPServer/TCPServer.cpp [31] if (bind(hSocket, (SOCKADDR*)&serverAddress, sizeof(serverAddress)) == SOCKET_ERROR) {
Matched [11] : ./Sockets/TCPServer/TCPServer.cpp [32] cerr << "bind() failed\n.";
Matched [19] : ./Sockets/TCPServer/TCPServer.cpp [33] goto close; // yes a goto, best practice = only goto forward in code.
Matched [9] : ./Sockets/TCPServer/TCPServer.cpp [35] cout << "TCP/IP socket bound.\n";
Matched [4] : ./Sockets/TCPServer/TCPServer.cpp [38] if (listen(hSocket, 1) == SOCKET_ERROR) {
Matched [7] : ./Sockets/TCPServer/TCPServer.cpp [39] cerr << "Error listening on socket.\n";
Matched [5] : ./Sockets/TCPServer/TCPServer.cpp [40] goto close;
Matched [5] : ./Sockets/TCPServer/TCPServer.cpp [43] // make a connection
Matched [11] : ./Sockets/TCPServer/TCPServer.cpp [44] {
Matched [3] : ./Sockets/TCPServer/TCPServer.cpp [45] cout << "Waiting for connection\n";
Matched [3] : ./Sockets/TCPServer/TCPServer.cpp [46] SOCKET hAccepted = SOCKET_ERROR;
Matched [6] : ./Sockets/TCPServer/TCPServer.cpp [47] while (hAccepted == SOCKET_ERROR)
Matched [5] : ./Sockets/TCPServer/TCPServer.cpp [49] cout << "Client connected\n";
Matched [3] : ./Sockets/TCPServer/TCPServer.cpp [51] int bytesSent;
Matched [4] : ./Sockets/TCPServer/TCPServer.cpp [52] char sendbuf[32] = "Goodbye";
Matched [3] : ./Sockets/TCPServer/TCPServer.cpp [53] char recvbuf[32] = "";
Matched [9] : ./Sockets/TCPServer/TCPServer.cpp [55] int bytesRecv = recv(hAccepted, recvbuf, 32, 0);
Matched [9] : ./Sockets/TCPServer/TCPServer.cpp [56] cout << "Recv = " << bytesRecv << ", " << recvbuf << endl;
Matched [8] : ./Sockets/TCPServer/TCPServer.cpp [57] bytesSent = send(hAccepted, sendbuf, strlen(sendbuf) + 1, 0);
Matched [8] : ./Sockets/TCPServer/TCPServer.cpp [58] cout << "Sent = " << bytesSent << " bytes" << endl;
Matched [

```