```
 1: // $Id: dateserver.cpp,v 1.14 2014-05-23 13:26:36-07 - - $
 2:
 3: //
 4: //
 5: // NAME
 6: //    dateserver - accept connections and reply with current date
 7: //
 8: // SYNOPSIS
 9: //    dateserver [host] port
10: //
11: // DESCRIPTION
12: //    If port and host are given, operates as a client and gets
13: //    the date from the server.
14: //    if only port is given, operates as a server in an infinite
15: //    loop returning the date for each connection.
16: //    Should probably use UDP instead of TCP, but it is easier
17: //    if we avoid two different protocol discussions.
18: //    The server also accepts a telnet connection.
19: //
20:
21: #include <iostream>
22: #include <sstream>
23: #include <stdexcept>
24: #include <string>
25: #include <typeinfo>
26: #include <vector>
27: using namespace std;
28:
29: #include <libgen.h>
30: #include <time.h>
31:
32: #include "../simple-sockets/sockets.h"
33:
34: string execname;
35: int exit_status = EXIT_SUCCESS;
36:
37: template <typename Type>
38: Type from_string (const string &str) {
39:    stringstream stream;
40:    stream << str;
41:    Type result;
42:    if (not (stream >> result and stream.eof())) {
43:       throw domain_error (string (typeid (Type).name())
44:             + " from_string (" + str + ")");
45:    }
46:    return result;
47: }
48:
49: void usage() {
50:    cerr << "Usage: " << execname << " [host] port" << endl;
51:    exit_status = EXIT_FAILURE;
52: }
53:
```

```
 54:
 55: void date_client (vector<string> args) {
 56:    client_socket server (args[0], from_string<in_port_t> (args[1]));
 57:    for (;;) {
 58:       string buffer;
 59:       int nbytes = server.recv (buffer);
 60:       if (nbytes == 0) break;
 61:       cout << buffer;
 62:    }
 63: }
 64:
 65: void date_server (vector<string> args) {
 66:    server_socket listener (from_string<in_port_t> (args[0]));
 67:    for (;;) {
 68:       accepted_socket client;
 69:       listener.accept (client);
 70:       time_t now = time (NULL);
 71:       struct tm* localnow = localtime (&now);
 72:       char buffer[64];
 73:       size_t bufbytes = strftime (buffer, sizeof buffer,
 74:                           "%a %b %e %H:%M:%S %Z %Y\n", localnow);
 75:       for (char* bufp = buffer; bufbytes > 0; ){
 76:          int nbytes = client.send (bufp);
 77:          bufp += nbytes;
 78:          bufbytes -= nbytes;
 79:       }
 80:    }
 81: }
 82:
 83: int main (int argc, char** argv) {
 84:    execname = basename (argv[0]);
 85:    vector<string> args (&argv[1], &argv[argc]);
 86:    try {
 87:       switch (args.size()) {
 88:          case 1: date_server (args); break;
 89:          case 2: date_client (args); break;
 90:          default: usage(); break;
 91:       }
 92:    }catch (socket_error& error) {
 93:       cerr << execname << error.what() << endl;
 94:    }
 95:    return exit_status;
 96: }
 97:
 98: // This is not the correct way do #include.  We should have
 99: // a Makefile to do this, but mkc and mkt assume standalone
100: // programs.  But it's easier to have a standalone example
101: // for such a simple program.
102: #include "../simple-sockets/sockets.cpp"
103:
104: //TEST// mkpspdf dateserver.ps dateserver.cpp* dateserver.output
105:
```

```
    1: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: starting dateserver.cpp
    2: dateserver.cpp:
    3:       $Id: dateserver.cpp,v 1.14 2014-05-23 13:26:36-07 - - $
    4: g++ -g -O0 -Wall -Wextra -std=gnu++11 dateserver.cpp -o dateserver -lglu
t -lGLU -lGL -lX11 -lm -lrt
    5: rm -f dateserver.o
    6: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: finished dateserver.cpp
```

```
 1:
 2: bash-2$ ./dateserver 8888&
 3: [1] 21136
 4:
 5: bash-3$ telnet localhost 8888
 6: Trying ::1...
 7: telnet: connect to address ::1: Connection refused
 8: Trying 127.0.0.1...
 9: Connected to localhost.
10: Escape character is '^]'.
11: Thu May 22 18:07:59 PDT 2014
12: Connection closed by foreign host.
13:
14: bash-4$ ./dateserver localhost 8888
15: Thu May 22 18:08:07 PDT 2014
16:
17: bash-5$ ./dateserver localhost 8888
18: Thu May 22 18:08:14 PDT 2014
19:
20: bash-5$ ./dateserver localhost 8888
21: Thu May 22 18:08:15 PDT 2014
22:
23: bash-5$ kill %1
24:
25: bash-6$
26: [1]+  Terminated              ./dateserver 8888
27:
28: bash-6$ exit
29: exit
30:
31: Script done on Thu 22 May 2014 06:08:27 PM PDT
```