

```
1: // $Id: addrinfo.cpp,v 1.12 2014-05-22 14:35:13-07 - - $
2:
3: #include <iomanip>
4: #include <iostream>
5: #include <sstream>
6: #include <string>
7: #include <unordered_map>
8: #include <vector>
9: using namespace std;
10:
11: #include <arpa/inet.h>
12: #include <netdb.h>
13: #include <netinet/in.h>
14: #include <string.h>
15: #include <sys/socket.h>
16: #include <sys/types.h>
17:
18: string socktype (int socktype) {
19:     static unordered_map<int,string> socktype_map {
20:         {SOCK_DGRAM , "dgram" },
21:         {SOCK_RAW   , "raw"    },
22:         {SOCK_STREAM, "stream"},
23:     };
24:     const auto& result = socktype_map.find (socktype);
25:     return result == socktype_map.end() ? "(\\?\\?\\?\\?)" : result->second;
26: }
27:
28: string af_family (int family) {
29:     static unordered_map<int,string> family_map {
30:         {AF_FILE , "File" },
31:         {AF_INET , "IPv4" },
32:         {AF_INET6 , "IPv6" },
33:         {AF_UNIX , "Unix" },
34:         {AF_UNSPEC, "Unspec"},
35:     };
36:     const auto& result = family_map.find (family);
37:     return result == family_map.end() ? "(\\?\\?\\?\\?)" : result->second;
38: }
39:
40: string inet_address (addrinfo* info) {
41:     sockaddr* addr = nullptr;
42:     switch (info->ai_family) {
43:         case AF_INET:
44:             addr = (sockaddr*) &((sockaddr_in*) info->ai_addr)->sin_addr;
45:             break;
46:         case AF_INET6:
47:             addr = (sockaddr*) &((sockaddr_in6*) info->ai_addr)->sin6_addr;
48:             break;
49:     }
50:     char textstr[INET6_ADDRSTRLEN];
51:     const char* text = inet_ntop (info->ai_family, addr,
52:                                   textstr, sizeof textstr);
53:     if (text == nullptr) text = strerror (errno);
54:     return textstr;
55: }
56:
```

```
57:
58: void print_addrinfo (const string& hostname) {
59:     addrinfo hints;
60:     memset (&hints, 0, sizeof hints);
61:     hints.ai_flags = AI_CANONNAME;
62:     hints.ai_family = AF_UNSPEC;
63:     addrinfo* results;
64:     cout << "IP addresses for " << hostname << ":" << endl;
65:     int rc = getaddrinfo (hostname.c_str(), NULL, &hints, &results);
66:     if (rc != 0) {
67:         cout << "      " << gai_strerror (rc) << endl;
68:         return;
69:     }
70:     for (auto info = results; info != nullptr; info = info->ai_next) {
71:         cout << "      " << setw(6) << socktype (info->ai_socktype)
72:             << " " << setw(6) << af_family (info->ai_family)
73:             << " " << inet_address (info);
74:         if (info->ai_canonname) cout << " " << info->ai_canonname;
75:         cout << endl;
76:     }
77:     freeaddrinfo (results);
78: }
79:
80: int main (int argc, char** argv) {
81:     string progname (argv[0]);
82:     vector<string> args (&argv[1], &argv[argc]);
83:     for (const auto& hostname: args) {
84:         print_addrinfo (hostname);
85:     }
86:     return 0;
87: }
88:
89: /*
90: //TEST// addrinfo www.google.com www.nohost.zone www.soe.ucsc.edu \
91: //TEST//          unix.ucsc.edu unix1.lt.ucsc.edu unix2.lt.ucsc.edu \
92: //TEST//          localhost >addrinfo.output 2>&1
93: //TEST// mkpspdf addrinfo.ps addrinfo.cpp* addrinfo.output
94: */
95:
```

GLU

```
1: IP addresses for www.google.com:
2:   stream  IPv4 74.125.239.52 www.google.com
3:   dgram   IPv4 74.125.239.52
4:   raw     IPv4 74.125.239.52
5:   stream  IPv4 74.125.239.48
6:   dgram   IPv4 74.125.239.48
7:   raw     IPv4 74.125.239.48
8:   stream  IPv4 74.125.239.49
9:   dgram   IPv4 74.125.239.49
10:  raw     IPv4 74.125.239.49
11:  stream  IPv4 74.125.239.50
12:  dgram   IPv4 74.125.239.50
13:  raw     IPv4 74.125.239.50
14:  stream  IPv4 74.125.239.51
15:  dgram   IPv4 74.125.239.51
16:  raw     IPv4 74.125.239.51
17:  stream  IPv6 2607:f8b0:4010:801::1010
18:  dgram   IPv6 2607:f8b0:4010:801::1010
19:  raw     IPv6 2607:f8b0:4010:801::1010
20: IP addresses for www.nohost.zone:
21:   Name or service not known
22: IP addresses for www.soe.ucsc.edu:
23:   stream  IPv4 128.114.50.76 www-01.soe.ucsc.edu
24:   dgram   IPv4 128.114.50.76
25:   raw     IPv4 128.114.50.76
26: IP addresses for unix.ucsc.edu:
27:   stream  IPv4 128.114.108.152 unix.lt.ucsc.edu
28:   dgram   IPv4 128.114.108.152
29:   raw     IPv4 128.114.108.152
30:   stream  IPv4 128.114.104.230
31:   dgram   IPv4 128.114.104.230
32:   raw     IPv4 128.114.104.230
33: IP addresses for unix1.lt.ucsc.edu:
34:   stream  IPv4 128.114.108.152 unix1.lt.ucsc.edu
35:   dgram   IPv4 128.114.108.152
36:   raw     IPv4 128.114.108.152
37: IP addresses for unix2.lt.ucsc.edu:
38:   stream  IPv4 128.114.104.230 unix2.lt.ucsc.edu
39:   dgram   IPv4 128.114.104.230
40:   raw     IPv4 128.114.104.230
41: IP addresses for localhost:
42:   stream  IPv6 ::1 localhost
43:   dgram   IPv6 ::1
44:   raw     IPv6 ::1
45:   stream  IPv4 127.0.0.1
46:   dgram   IPv4 127.0.0.1
47:   raw     IPv4 127.0.0.1
```