```
// $Id: inettime.cpp,v 1.11 2013-08-20 20:20:12-07 - - $

//
// Print time from internet time server.
//

#include <cerrno>
#include <cstdlib>
#include <cstring>
#include <ctime>
#include <iomanip>
#include <iostream>
#include <sstream>
#include <string>

#include <libgen.h>
#include <netdb.h>

using namespace std;

// Unix epoch is 1970-01-01 00:00:00 UTC
// Inet epoch is 1900-01-01 00:00:00 UTC
const long INET_UNIX_YEARS = 1970 - 1900;
const long INET_UNIX_DAYS = INET_UNIX_YEARS * 365 + INET_UNIX_YEARS / 4;
const long INET_UNIX_SECONDS = INET_UNIX_DAYS * 24 * 3600;
const string DEFAULT_HOST = "tick.ucla.edu";

string exec_name;

void crash (const string &object, const string &error) {
   cerr << exec_name << ": " << object << ": " << error << endl;
   exit (EXIT_FAILURE);
}

void print_time (const string &label, const time_t hosttime) {
   char timebuf[256];
   strftime (timebuf, sizeof timebuf, "%a %b %d %T %Y %Z",
             localtime (&hosttime));
   cout << setiosflags(ios::left) << setw(5) << label << " = "
        << setiosflags(ios::right) << setw(12) << hosttime
        << " = " << timebuf << endl;
}

template <typename type_t>
string to_string (const type_t &that) {
   ostringstream stream;
   stream << that;
   return stream.str ();
}

string host_string (char *host_addr) {
   string result;
   for (size_t itor = 0; itor < 4; ++itor) {
      if (itor > 0) result += ".";
      result += to_string (host_addr[itor] & 0xFF);
   }
   return result;
}

```

```

hostent *get_hostent (const string &hostname) {
   hostent *hostent = gethostbyname (hostname.c_str());
   if (hostent == NULL) crash (hostname, hstrerror (h_errno));
   cout << hostname << " = " << hostent->h_name << " = "
        << host_string (hostent->h_addr) << endl;
   return hostent;
}

int get_sock_fd (char *host_addr, int ipport) {
   int sockfd = socket (AF_INET, SOCK_STREAM, 0);
   if (sockfd < 0) crash ("socket()", strerror (errno));
   sockaddr_in sock;
   sock.sin_family = AF_INET;
   sock.sin_port = htons (ipport);
   sock.sin_addr = *reinterpret_cast<in_addr *> (host_addr);
   int connect_rc = connect (sockfd,
               reinterpret_cast<sockaddr *> (&sock), sizeof sock);
   if (connect_rc < 0) {
      crash ("connect(" + host_string(host_addr) + ":"
             + to_string (ipport) + ")", strerror (errno));
   }
   return sockfd;
}

string visible (const char *const daytime) {
   stringstream result;
   for (const char *itor = daytime; *itor != ’\0’; ++itor)
      if (isprint(*itor)) result << *itor;
         else result << "\\x" << uppercase << hex << int (*itor);
   return result.str();
}

```

```

int main (int argc, char **argv) {
   exec_name = basename (argv[0]);

   cout << "INET_UNIX_YEARS = " << INET_UNIX_YEARS << endl;
   cout << "INET_UNIX_DAYS = " << INET_UNIX_DAYS << endl;
   cout << "INET_UNIX_SECONDS = " << INET_UNIX_SECONDS << endl;
   cout << "IPPORT_TIMESERVER = " << IPPORT_TIMESERVER << endl;
   cout << "IPPORT_DAYTIME = " << IPPORT_DAYTIME << endl;

   // Get Inet host address and print it.
   string hostname = argc < 2 ? DEFAULT_HOST : argv[1];
   if (hostname == "-") hostname = "localhost";
   hostent *hostent = get_hostent (hostname);

   // Get information from IPPORT_TIMESERVER (port 37).
   int time_sock_fd = get_sock_fd (hostent->h_addr, IPPORT_TIMESERVER);
   uint32_t inettime;
   int rc = read (time_sock_fd, &inettime, sizeof inettime);
   if (rc != sizeof inettime) crash ("socket", "read failed");
   close (time_sock_fd);
   uint32_t hosttime = ntohl (inettime) - INET_UNIX_SECONDS;
   print_time ("inet", hosttime);
   print_time ("local", time (NULL));

   // Get information from IPPORT_DAYTIME (port 13).
   int date_sock_fd = get_sock_fd (hostent->h_addr, IPPORT_DAYTIME);
   char daytime_buf[1024];
   int nbytes = read (date_sock_fd, daytime_buf, sizeof daytime_buf);
   if (nbytes <= 0) crash ("socket", "read failed");
   close (date_sock_fd);
   daytime_buf[nbytes] = '\0';
   cout << "daytime = \"" << visible (daytime_buf) << "\"" << endl;

   return EXIT_SUCCESS;
}

//TEST// ./inettime >inettime.out 2>&1
//TEST// mkpspdf inettime.ps inettime.cpp* inettime.out

```

```
* @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: starting inettime.cpp
* inettime.cpp: $Id: inettime.cpp,v 1.11 2013-08-20 20:20:12-07 - - $
* g++ -g -O0 -Wall -Wextra -std=gnu++0x inettime.cpp -o inettime -lm
* rm -f inettime.o
* @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: finished inettime.cpp
```

```
INET_UNIX_YEARS = 70
INET_UNIX_DAYS = 25567
INET_UNIX_SECONDS = 2208988800
IPPORT_TIMESERVER = 37
IPPORT_DAYTIME = 13
tick.ucla.edu = tick.ucla.edu = 164.67.62.194
inettime: connect(164.67.62.194:37): Connection timed out
```