Week 12 Lab 2

1. Readings

From <https://en.wikipedia.org/wiki/Network_Time_Protocol>:

Network Time Protocol is a networking protocol for clock synchronization between computer systems over data networks. NTP synchronizes all participating computers to a few milliseconds of coordinated universal time.

From <https://www.nist.gov/time-distribution/internet-time-service-its>:

Using the internet to set your computer time, and this time is directly linked to UTC(NIST), the official NIST time.

From <https://tf.nist.gov/tf-cgi/servers.cgi>:

Some of the available NIST internet time servers:

* time-a-g.nist.gov 129.6.15.28 NIST, Gaithersburg, Maryland All services available
* time-b-g.nist.gov 129.6.15.29 NIST, Gaithersburg, Maryland All services available
* time-c-g.nist.gov 129.6.15.30 NIST, Gaithersburg, Maryland All services available
* time-d-g.nist.gov 129.6.15.27 NIST, Gaithersburg, Maryland All services available
* time-d-g.nist.gov 2610:20:6f15:15::27 NIST, Gaithersburg, Maryland All services via IPV6

1. Getdate.c

Text

Description automatically generated

1. Getdate.c modified

Text

Description automatically generated

Listing:

/\* Start with the usual includes and declarations. \*/

#include <sys/socket.h>

#include <netinet/in.h>

#include <netdb.h>

#include <stdio.h>

#include <unistd.h>

#include <stdlib.h>

**int** main(**int** argc, **char** \*argv[])

{

**char** \*host;

**int** sockfd;

**int** len, result;

**struct** sockaddr\_in address;

**struct** hostent \*hostinfo;

**struct** servent \*servinfo;

**char** buffer[128];

**if**(argc == 1)

host = "time-a-g.nist.gov";

**else**

host = argv[1];

/\* Find the host address and report an error if none is found. \*/

hostinfo = gethostbyname(host);

**if**(!hostinfo) {

fprintf(stderr, "no host: %s\n", host);

exit(1);

}

/\* Check that the daytime service exists on the host. \*/

servinfo = getservbyname("daytime", "tcp");

**if**(!servinfo) {

fprintf(stderr,"no daytime service\n");

exit(1);

}

printf("daytime port is %d\n", ntohs(servinfo -> s\_port));

/\* Create a socket. \*/

sockfd = socket(AF\_INET, SOCK\_STREAM, 0);

/\* Construct the address for use with connect... \*/

address.sin\_family = AF\_INET;

address.sin\_port = servinfo -> s\_port;

address.sin\_addr = \*(**struct** in\_addr \*)\*hostinfo -> h\_addr\_list;

len = **sizeof**(address);

/\* ...then connect and get the information. \*/

result = connect(sockfd, (**struct** sockaddr \*)&address, len);

**if**(result == -1) {

perror("oops: getdate");

exit(1);

}

result = read(sockfd, buffer, **sizeof**(buffer));

buffer[result] = '\0';

printf("read %d bytes: %s", result, buffer);

close(sockfd);

exit(0);

}