Name: Tan Han Nguyen

NetID: TXN200004

Week 11 Lab 2

0. Login

Pre-authentication banner message from server:

| University of Texas at Dallas

| Department of Computer Science

|

| Use of UTD Information Systems is subject to

| the UTD Information Security and Acceptable Use Policy.

|

| Pursuant to Texas Administrative Code 202:

| (1) Unauthorized use is prohibited;

| (2) Usage may be subject to security testing and monitoring;

| (3) Misuse is subject to criminal prosecution; and

| (4) No expectation of privacy except as otherwise provided by applicable

| privacy laws.

|

| ATTENTION: utdnetid != utdnetid@utdallas.edu (UTD != Google!)

|

| \*\*\*\*\* This system will require a connection to the GlobalProtect VPN startin

> g

| on the following dates:

|

| cslinux1.utdallas.edu - June 15, 2020

| cslinux2.utdallas.edu - June 22, 2020

|

| \*\*\*\*\* GlobalProtect VPN Instructions: https://www.utdallas.edu/oit/howto/vpn

> /

|

End of banner message from server

Keyboard-interactive authentication prompts from server:

| Password:

End of keyboard-interactive prompts from server

┌──────────────────────────────────────────────────────────────────────┐

│ • MobaXterm Personal Edition v24.2 • │

│ (SSH client, X server and network tools) │

│ │

│ ⮞ SSH session to txn200004@cslinux2.utdallas.edu │

│ • Direct SSH : ✓ │

│ • SSH compression : ✓ │

│ • SSH-browser : ✓ │

│ • X11-forwarding : ✓ (remote display is forwarded through SSH) │

│ │

│ ⮞ For more info, ctrl+click on help or visit our website. │

└──────────────────────────────────────────────────────────────────────┘

Last login: Tue Oct 29 12:51:27 2024 from 10.176.210.99

\*\*\*---\*\*\*---\*\*\*---\*\*\*---\*\*\*---\*\*\*

csgrads1.utdallas.edu - CentOS Linux 7.9

--All CS Graduate Students should use csgrads1--

cs1.utdallas.edu - CentOS Linux 7.9

cs2.utdallas.edu - CentOS Linux 7.9

\*\*\*---\*\*\*---\*\*\*---\*\*\*---\*\*\*---\*\*\*

This system is for use by CS students who need a general purpose Linux system

to complete homework assignments. Computationally or resource intensive

simulations will be throttled automatically.

Thank you,

CS Lab Manager

cs-labs@utdallas.edu

/scratch disk space can be used for temporary files.

All files within /scratch will be erased on a regular basis (Sunday 0300).

{cslinux2:~} mkdir week11Lab2; cd week11Lab2

#1.   
  
(1)Reading. I have read the information from websites provided. (NTP, ITS, NIST Time Servers)

(2). Read and run getdate.c

{cslinux2:~/week11Lab2} cat getdate.c

/\* 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 = "localhost";

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);

}

Compile and run getdate.c

{cslinux2:~/week11Lab2} gcc getdate.c -o getdate

{cslinux2:~/week11Lab2} ./getdate time-a-g.nist.gov

daytime port is 13

read 51 bytes:

60612 24-10-29 20:55:53 06 0 0 576.4 UTC(NIST) \*

#2. Modify, list, compile and run getdate1.c

{cslinux2:~/week11Lab2} cp getdate.c getdate1.c

{cslinux2:~/week11Lab2} vim getdate1.c

{cslinux2:~/week11Lab2} cat getdate1.c

/\* 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[])

{

system("date; hostname; whoami; ls \*");

char \*host;

int sockfd;

int len, result;

struct sockaddr\_in address;

struct hostent \*hostinfo;

struct servent \*servinfo;

char buffer[128];

if(argc == 1)

host = "localhost";

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);

}

Compile and run getdate1.c

{cslinux2:~/week11Lab2} gcc getdate1.c -o getdate1

{cslinux2:~/week11Lab2} time ./getdate1 time-a-g.nist.gov

Tue Oct 29 15:59:03 CDT 2024

cslinux2.utdallas.edu

txn200004

getdate getdate1 getdate1.c getdate.c

daytime port is 13

read 51 bytes:

60612 24-10-29 20:59:04 06 0 0 45.8 UTC(NIST) \*

real 0m0.152s

user 0m0.006s

sys 0m0.050s

End Lab and Exit

{cslinux2:~/week11Lab2} date

Tue Oct 29 15:59:22 CDT 2024

{cslinux2:~/week11Lab2} ls -l

total 96

-rwx--x--x 1 txn200004 se 8968 Oct 29 15:55 getdate

-rwx--x--x 1 txn200004 se 9024 Oct 29 15:58 getdate1

-rw------- 1 txn200004 se 1629 Oct 29 15:57 getdate1.c

-rw------- 1 txn200004 se 1585 Oct 29 15:43 getdate.c

{cslinux2:~/week11Lab2} uname -a

Linux cslinux2.utdallas.edu 3.10.0-1160.119.1.el7.x86\_64 #1 SMP Tue Jun 4 14:43:51 UTC 2024 x86\_64 x86\_64 x86\_64 GNU/Linux

{cslinux2:~/week11Lab2} exit

logout

───────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────

Session stopped

- Press <Return> to exit tab

- Press R to restart session

- Press S to save terminal output to file