

CSCE 5580: Computer Networks

Programming Assignment 2

Due: 11:59 PM on Thursday, March 19, 2020

PROGRAM DESCRIPTION:

In this assignment, you will implement network port scanner in C that will report the status of services and their associated ports for the TCP and UDP protocols for a given port range. Your port scanner shall be invoked as follows:

```
portScan <hostname> <protocol> <portlow> <porthigh>
```

where

- <hostname> is the host name of the machine you want to run the port scanner on (e.g., cse05).
- <protocol> is the protocol, limited to "TCP" or "UDP".
- <portlow> is the lower range, inclusively, of ports to scan.
- <porthigh> is the upper range, inclusively, of ports to scan.

The basic idea behind a network port scanner is simple: given a host name of a machine and a list of ports to scan, the scanner will connect to each port using sockets with the specified protocol (i.e., TCP or UDP) to determine whether or not the port is open based on the success of a connection request and then close the socket before moving on to the next port to scan. Especially for TCP, you will not allow for "half-open" scans (i.e., connection requests without a corresponding close) as it can lead to network failures and availability issues. Where available, you will include the name of the service that is likely running on that port. Visit <https://www.iana.org/assignments/port-numbers> for information about the services and transport protocol associated with each of the ports. If no service is available or you are unable to determine the service for open ports, you will simply display "svc name unavail".

Your program shall also adhere to the following requirements:

- If the user does not enter all required command-line arguments, you will display a usage statement and terminate the program. You may assume the user enters the correct data type (e.g., string or integer), but you should check for errors and handle accordingly as appropriate.
- Your program should run on the INET domain using SOCK_STREAM and SOCK_DGRAM for TCP and UDP sockets, respectively so that the port scanner can operate over the network on a remote machine.

SAMPLE OUTPUT (user input shown in **bold**):

==> port scanner on cse05

```
$ ./portScan
```

```
usage: ./portScan <hostname> <protocol> <portlow> <porthigh>
```

```

$ ./portScan cse04
usage: ./portScan <hostname> <protocol> <portlow> <porthigh>
$ ./portScan cse08 tcp 1 120
scanning host=cse08, protocol=tcp, ports: 1 -> 120
error: host cse08 not exist
$ ./portScan cse05 xyz 1 120
scanning host=cse05, protocol=xyz, ports: 1 -> 120
invalid protocol: xyz. Specify "tcp" or "udp"
$ ./portScan cse04 tcp 1 120
scanning host=cse04, protocol=tcp, ports: 1 -> 120
port 22      open    : ssh
port 111     open    : sunrpc
$ ./portScan cse04 tcp 9000 9005
scanning host=cse04, protocol=tcp, ports: 9000 -> 9005

%% NOW PROGRAM RUN ON CSE04 USING TCP PORT 9001

$ ./portScan cse04 tcp 9000 9005
scanning host=cse04, protocol=tcp, ports: 9000 -> 9005
port 9001    open    : svc name unavail
$ ./portScan cse04 udp 1 30
scanning host=cse04, protocol=udp, ports: 1 -> 30
port 7       open    : echo
port 9       closed  : discard
port 13      open    : daytime
port 18      closed  : msp
port 19      open    : chargen
port 21      closed  : fsp
port 25      open    : svc name unavail
$ ./portScan cse04 udp 8000 8005
scanning host=cse04, protocol=udp, ports: 8000 -> 8005

%% NOW PROGRAM RUN ON CSE04 USING UDP PORT 8001

$ ./portScan cse04 udp 8000 8005
scanning host=cse04, protocol=udp, ports: 8000 -> 8005
port 8001    open    : svc name unavail

```

REQUIREMENTS:

- Your code should be well documented in terms of comments. For example, good comments in general consist of a header (with your name, course section, date, and brief description), comments for each variable, and commented blocks of code.
- Your program should be named “**project2.c**”, without the quotes.
- Your program will be graded based largely on whether it works correctly on the CSE machines (e.g., cse01, cse02, ..., cse06), so you should make sure that your program compiles and runs on a CSE machine.
- Please pay attention to the **SAMPLE OUTPUT** for how this program is expected to work. If you have any questions about this, please contact your instructor or TAs assigned to this course to ensure you understand these directions.

- This is an individual programming assignment that must be the sole work of the individual student. Any instance of academic dishonesty will result in a grade of “F” for the course, along with a report filed into the Academic Integrity Database.

SUBMISSION:

- You will electronically submit your C source code file `project2.c` to the **Project 2** dropbox in Canvas by the due date.