Computer Networks I Fall 2013

Programing Project #2 - DNS Resolution 2013-09-26

In this project students will develop there own version version of the DNS tool ***dig***, which we will call ***dug***. ***Dug*** wi provide many of the same features of dig, however unlike ***dig*** you will not be allowed to take advantage of any of the libraries built-in to the operating system for DNS support. The assignment will be graded in four parts

1. The most basic functionality, the ability to query an arbitrary nameserver for an A record while specifying that recursion is not desired, is worth 70% of the grade.
2. Adding the ability to perform recursive queries within you code is adds an additional 20%.
3. Daemonizing your program so that it can respond to queries from programs like ***dig*** is worth an additional 10%

Details

Students are encouraged to approach this project one step at a time, confirming that each phase of the development works correctly before trying to move on to the next stage. Note that the precise behavior of the DNS system is best documented in the RFCs 1034 and 1035. Should there be question about how your program should behave you should refer to those documents. I have also found it useful to use the Wireshark tool to examine the packets sent to/from DNS servers to confirm that my programs format the packages the same way as the "official" programs.

Part 1 - Peform a simple query

Your first version of your program should take two command line arguments - the name of the host you want to ask about and the IP address of the nameserver you want to query. Your program will need to build the query packet and send it to the nameserver via a UDP packet. At this point you should only make **A** type queries and should not request recursion. If you receive an A record back you should display the IP address along with an indication of the status of the answer (was it authoritative or not). If no information is found print an appropriate message ("unknown host"). Any other information returned may be ignored at this stage.

Part 2 - Implement recursion

Once you can reliably make **A** queries you need to expand your code so that it understands **NS** and **CNAME** types. This will allow your program to make recursive queries to discover the answers to questions the local nameserver does not have. At this stage your program should be able to take the same two command line parameters (hostname and nameserver) but if the nameserver you query does not know the answer to your question it will include other helpful information that you can use to find the nameserver that does know the answer.

Part 3 - Daemonizing your program

To this point your program should be able to answer most A record queries, with the information supplied on the command line. Modify your program so that it will run as daemon, accepting and answering queries from other programs (e.g., dig)

Extra Credit

Additional functionality will be allowed if you would like to try for extra credit. In particular:

1. Add caching to your program so that it will remember answers already given.
2. Add support for other query types (HINFO, PTR, SOA etc.)

Submission

This assignment is due by the end of the day on Sunday, October 20th. There will be a 5% per day penalty for late submissions. Students should submit a single tarball of a directory containing a Makefile and the required source files. If needed please include a README.txt file with instructions. The grader should not need to do anything other than untar your directory, cd to the directory, type make and begin testing.The directory should be named with your username.

In addition to functionality you will be graded on the quality of the code, including readability, comments and the use of proper programing practices.

Usage

At a minimum we will expect the following behavior:

- Basic, non recursive test:

```
$> dug imagine.mines.edu 138.67.1.2
   Authoritative answer: 138.67.132.196
```

- Test requiring recursion:

```
$> dug cse.unl.edu 138.67.1.2
   Authoritative answer: 129.93.165.2
```

- Test requiring recursion with debugging.

```
$> dug -d cse.unl.edu 138.67.1.2
          :
    List each successive answer
          :
    Authoritative answer: 129.93.165.2
```

- Daemon testing

```
$> dug -f 138.67.1.2 &
   Port WXYZ selected
$> dig -p WXYZ illusion.mines.edu @138.67.1.2
```