

ICS 53, Lab 2, Due April 16, 2014 11:59PM

This week's exercise is designed to give you a quick introduction to the Internet's **Domain Name System** and **Unix File System**. This is an example of a naming system which all of you use on a daily basis --- in fact you use it to get to web pages on browsers. http://en.wikipedia.org/wiki/Domain_Name_System

(Please also read the related materials in your book for URL/DNS)

This lab assignment is very simple and you only need to work with "**dig**" command in linux with some options. <http://linux.die.net/man/1/dig>

We expect that you will do this assignment on the ICS openlab servers. If you have your own linux machine and it and you want to install dig on it, you can find some sources on the web for its installation like

(for ubuntu write in terminal : **sudo apt-get install dnsutils**):
<http://askubuntu.com/questions/25098/how-do-i-install-dig>

Upload your solution, as either a pdf file or plain text file, to the appropriate dropbox on eee.

Problem #1:

What is the IP address of www.ics.uci.edu ? What is the Time To Live (TTL) value for this record? Paste the related section of the dig output in your solution and identify the TTL value.

IP: 128.195.1.83

TTL: 86400

www.ics.uci.edu. 86400 IN CNAME www-main.ics.uci.edu.

www-main.ics.uci.edu. 86400 IN A 128.195.1.83

TTL is after the domain name

Problem #2:

For this problem, you will go through the steps of resolving a particular hostname, by iterating through a series of servers, just like a regular server. Assuming it knows nothing else about a name, a DNS resolver will ask a well-known root server.

One way to get a list of them is with the command:

```
dig . ns
```

Perform a DNS lookup step by step (without auto-recursion) and show your command sequence.

```
dig +norecurse @198.41.04 www.ics.uci.edu
```

```
dig +norecurse @192.5.6.30 www.ics.uci.edu
```

```
dig +norecurse @128.48.101.53 www.ics.uci.edu
```

Problem #3:

Usually DNS has cache that contains records of recently accessed domains. If the cache has the answer for a DNS query, the answer can be returned without doing extra look-up. Submit the dig commands you use and related output section, and briefly explain how you can determine whether the returned result is a cached result or not.

```
dig www.ics.uci.edu
```

```
;; Query time: 1 msec
;; SERVER: 128.195.25.66#53(128.195.25.66)
;; WHEN: Wed Apr 16 11:56:54 2014
;; MSG SIZE rcvd: 216
```

```
dig +trace www.ics.uci.edu
```

```
;; Received 512 bytes from 128.195.25.66#53(128.195.25.66) in 2 ms
;; Received 268 bytes from 192.112.36.4#53(192.112.36.4) in 57 ms
;; Received 121 bytes from 192.41.162.30#53(192.41.162.30) in 70 ms
;; Received 216 bytes from 128.200.59.190#53(128.200.59.190) in 1 ms
```

The first dig command returned a cache result since the query time is 1 msec or less, while doing the trace results in multiple queries from name servers to find the domain with the total time being 130 ms.

Problem #4:

Please read the case study in Chapter 2 of the course textbook: **Case study: Unix® file system layering and naming (Section 2.5)**. Then answer questions in the **Bigger Files Problem** set in the back of the textbook, Q 1.1 and Q 1.2 on pages 427-428.

- Q 1.1 B. Increase just the number of bytes per block from 512 to 2048 bytes.
 D. Replace one of the direct block numbers in each i-node with an additional triple-indirect block number.
- Q 1.2 B. Decreasing the size of a block number from 4 bytes to 3 bytes.