

Programming Assignment 2: SNS (cont'd)

Assigned: February 19, 2004

Due: February 26, 2004, 11:59pm

1 Introduction

The real name service in the Internet is used to resolve the mapping between a hostname and an IP address. In this assignment there are two entities that comprise the name service system — a name server and a resolver (client). You will implement a simplified version of *the server only*. It is a continuation from Programming Assignment# 1 and the protocol details are the same as before.

The database file used by the server will be an ASCII file with format as follows. The first line will contain the total number of entries in the file. Each subsequent line of this file will contain a hostname and corresponding IP address separated by white space. For example,

```
2
machine1.cs.wisc.edu 128.45.6.3
machine2.yahoo.com 12.4.3.7
```

(You can assume that there will be no errors in the data format for this file.)

Also, please remember to implement message type 101, which is sent by the client to the server and indicates that the server should terminate. Note that unless the server receives this message, the server should not terminate. The server does not send any response to the client, but terminates silently.

2 Executables

The following is the expected executable format for the server (along with the necessary arguments):

- `server -p <port> -f <database-file-name>`
port: Port number of the host at which the server
database-file-name: Name of the database file to use which maps hostnames to IP addresses.

The server is not required to output any text to screen in this programming assignment.

You should take care that your code is robust and handles possible erroneous messages from the client.

3 Submission

You will need to submit the source code along with a Makefile (located in a directory called `p2/`). in a single tar.gz file (name it `p2.tar.gz`). Do not submit object files, or compiled executables. The Makefile should have two rules: `clean` and `all`. `clean` will delete previous `.o` files, executables, etc. `all` should produce a single executable called `server`. The TA will run the following sequence of operations to execute and test your code:

```
tar xvfz p2.tar.gz
cd p2/
make clean
make all
./server -p <port> -f <database-file>
```

Please test to make sure that these commands can execute in sequence without any intervention. Further submission instructions will be posted in the class webpage.