CS256 Lab 7 - Popularity contest

Value: 30 points, plus 5 or 8 points extra credit
Due: Friday 25 October @ 23:50 (design document due for class on Wednesday 23 October)
10% bonus for submitting it before Friday @ 18:00!
Submission Format: Design document (image) and a Python script, lab_07_sl.py, optionally lab_07_sl_ec.py.

As the book's material on linked lists describes they are often used to keep track of access frequencies of one sort or another, along with many other common tasks. This lab will give you practice designing and building single linked structures, the context is parsing and analyzing access logs from a web server.

Setup

The CS department supports a number of web servers (Apache), one of them hosts a bunch of virtual domains such as http://cs.earlham.edu, http://fieldscience.cs.earlham.edu, and others. The department would like to know what the most popular virtual domain is and how many times it was referenced during the period the logfile covers.

Tasks

Design, write and test:

- A class for a list node to be used in a single linked list, it will need fields for a domain name, a reference count, and a pointer to the next node.
- A single linked list class, it will need a field for the head pointer and depending on your design possibly others.
- A method that given an input log file populates a linked list with the domain names and references found in the log file, returning the linked list to the caller (your main()). Your algorithm will need to read one line of the file at a time, parse-out the domain name (the first column, space separated), remove the port number from it, and then search your list for it. If you don't find it, add it to the list with a reference count of 1, if you find it in your list (i.e. you have seen it before) then increment the reference count for that node/domain name by one.
- Your main() function should read the log file name from the command line, check to see that it exists, call your populate method, and with the returned, populated list determine the following from it:
    - the total number of visits to all domains
    - the number of unique domain names
    - the most popular domain and how many visits it had
    - the % of total visits represented by that domain
- Your solution should print the results to stdout in this format (these are example values, the correct answers will different):

```
25445
14
alices-pentest-data.cs.earlham.edu 4440
11%
```

Implementation Notes

- Design your solution on paper first, there is more of your design in this lab than in previous ones. Most people would start by reviewing the whole, and then designing the low-level elements (the class/method definitions) and then move to the higher level tasks. Update it as you develop and refine your thinking. Make sure your design covers all of the assignment, you will be turning-in an image of it on Wednesday and Friday.
- Your script should have one command line argument, -i for the input logfile.
- You may need to refresh your memory of the builtins that do splitting, stripping, etc. of strings.
- Your script should use the main() technique, it helps organize the code and it makes reuse much easier.
- Your algorithm for determining the four values should run in O(n) time, that is you should only make one pass through the list to determine all four values. N in this case is the number of log records.
- Read the input file one line at a time rather than slurping it up in one go.
- We will provide both small and large input files to test with, you should use all of them, volume matters and you need to show that your solution works across the range of reasonable input sizes.
- Note that the percentage has been correctly rounded to two places.

Extra Credit

- (5 points) Make a copy of the base solution and modify it so that it lists the top 3 domains in order with their visit numbers and the fraction of the total visits those three represent, like so:

  ```
  25445
  14
  alices-pentest-data.cs.earlham.edu 4440
  carols-reindeer-data.cs.earlham.edu 3992
  teds-fish-data.cs.earlham.edu 2783
  44%
  ```

  This solution must also run in O(n) time.

- (3 points) Add a command line parameter -t which controls the top N domains to display, for example this command line would list the top 5 (and calculate the percentage appropriately) in addition to the other information:

  ```
  $ python3 lab_07_ec.py -i logfile.dat -t 5
  ```