

CS306 – Introduction to Perl

Fall 2005

Homework Assignment #4

Due: November 7th, 2005 (start of class)

The homework should be submitted in the form of a zip file which contains one program for each question. Your zip file should be named *firstname-lastname-hw4.zip* and should contain a file *hw4q1.pl*. Email this zip file to cs306@cis.uab.edu.

1. (20 pts) Write a web server logfile analyzer. This program should take a standard Apache logfile and produce the following statistics:

- The total number of requests handled
- The total number of bytes transferred
- A list in descending order of the number of requests made for each page
- A list in descending order of the number of requests made from each address
- A list in descending order of the number of times each result code appears
- Average number of bytes / request
- A chart showing how many requests were served each month
- A chart showing how many requests were served in each hour of the day
- A chart showing how many times each protocol was used (e.g. HTTP/1.0)

In addition, your program must practice good Perl design principles. **Specifically, you should have thorough commenting, well-aligned and well-indented code, subroutines as organizational blocks, no global variables, sensible data structures, meaningful variable names, appropriate error handling and be operating under 'use strict' and 'use warnings'.**

A sample logfile entry looks like (this is all one line – the only \n is at the very end):

```
lj2566.inktomisearch.com - - [27/Oct/2005:13:07:33 -0500] "GET /info/grads/mmf/edpage/edpage2.html HTTP/1.0" 404 11924
```

The fields, in order, are:

- address making the request
- client ident (the – means no info available) (not used in this assignment)
- username (again, not used in this assignment)
- date and time including offset from universal time appear between []
- the request line appears within “ “
- the result code
- the number of bytes transferred

A sample logfile will be available for download on the course website. However, a different file will

be used to test your script. The output should be neat and easy to read. Please see the sample output below for an example. Your output does not need to match the example, but it must be neat, well-aligned, easy to read and include all of the elements outlined above.

4 POINT BONUS. Add ASCII bar charts to your output for the chart items (time of day, day of week). They can be horizontal (bars grow from left to right) but must fit within 60 columns total width of output no matter how big the value is (so scale all the values accordingly). You should still indicate absolute values somewhere on your chart.

Hints

- Since the log entries themselves have / in them, you may want to use the feature of pattern matching that allows you to use other delimiters besides // for the pattern. For example, you could use `m|pattern goes here|` instead. Note that when you use an alternate delimiter, the leading `m` (for (m)atch) is no longer optional.
- Some pages take parameters. These appear after the ? in the request string. For example, `/faculty/view.php?id=2`. In order to accurately record page requests, you need to ignore the ? and everything after it. This request should be considered a request to `/faculty/view.php`.
- The ? modifier to the .* pattern might be helpful. .* normally means “match zero or more anythings”. However, .*? changes this to be “match zero or more anythings non-greedily”. To illustrate the difference, given the string “hey now|brown|cow”, in the pattern `/hey.*|/`, the .* part matches “now|brown” due to the Leftmost, *Longest* Rule. However, if the pattern were `/hey.*?|/`, the .*? portion only matches “now”. The ? in the pattern alters the behavior to make the shortest match rather than the longest match.

Sample Output

```
File Analyzed: /home/fran/sample-logfile
```

```
-----  
| Summary Statistics |  
-----
```

```
Total Number of Requests: 401  
Total Number of Bytes:    12156803
```

```
Requests Per Page
```

```
-----  
/                               35  
/favicon.ico                    17  
/images/banner750.gif          11  
/courses/                       8
```

```
etc....
```