UNIVERSITY OF SOUTHAMPTON

SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

System Administration Tools and Techniques COMP1005
2010/11 Coursework

Lecturers: Prof. Bob Damper and Dr. Kirk Martinez

Due Date and Time: 1600, Friday 13 May 2011

This coursework contributes 30% of the marks for this module. Coursework is the continuously assessed part of the module and is a required part of the degree assessment. Assessed questions must be handed in by the due date to the C-BASS online system. The use of the online system is part of the assessment.

Aim: To gain practical experience of a range of UNIX tools discussed in the module.

Objectives: To explore some of the UNIX scripting facilities, gain a greater understanding of html and ensure you can use a UNIX system and LaTeX document preparation system effectively.

Requirements: Your task involves the development of scripts for the analysis of web pages. First, you must ensure you have some web pages to work with. There are some marks for this; if you are not familiar with html and creating links in web pages then becoming familiar is part of the coursework. However, the majority of the marks are for the tasks below. For testing purposes you will see that your web pages will need to contain a variety of links. Use small purpose-designed web pages.

Write an awk script (called `aa1`) which takes a single web page filename or path as an argument and counts the links that are on that page. Your script should write to standard output the filename of the web page source, followed by a list of each unique link on the page and the number of times it occurs on that page. So if the url `http://www.bloggs.ac.uk/index.html` appears three times on the page, it should appear in some suitable form in your list with the number 3 after it, separated by spaces. (See also the NB below in the section on `ss1` as you need to be able to re-detect the links in your output.) The links should be listed in three groups. The first group should be links to other web pages (file extensions `.htm` and `.html` or similar), the second should include links to images (file extensions `.jpg` or `.jpeg` etc.) and the third group should include links to other things (like pdfs or other downloadable files). Put a blank line between each group. There are a wide variety of ways in which links can be expressed in web pages and you are only expected to cope with the most common ones. Also, you only need to consider links within `<a>` and `<img>` tags.

NB: Your awk script should be in a shell script file (with a first line `#!/bin/gawk -f`). It should be possible to run it on a web page `X.html` with the command `aa1 X.html` or on a web page in `/tmp` with the command `aa1 /tmp/X.html`.

Next, write a C shell script (called `ss1`) with a synopsis:

```
ss1 [-r] directory
```

The ss1 command takes a directory path name as its argument and applies your awk script to all the web page files (those with .html and .htm extensions etc.) in the directory. The output from the script should be to standard output. If the -r option is set, the command should run recursively over all subdirectories in the hierarchy below the directory. To run ss1 on your own file hierarchy, it should be possible to do this with the command:

```
aa1 -r $HOME
```

NB: you should choose a form for listing your url's in the output of aa1so that you can run aa1 again on your output from ss1 (eg by piping the output from ss1 into aa1) to produce a summary of all the links in all the web pages in the hierarchy. The number of times each link occurs will now be the number of FILES in which each occurs.

Finally, using LaTeX, prepare a one-two page document describing your scripts, explaining how you designed and tested them and detailing any problems encountered.

Important note: this coursework will be tested on a Linux platform like those in the UG labs. It is your responsibility to ensure your scripts run on that platform and that the script names are correct. Use a relative pathname to refer to your awk script in the shell script and assume they will be in the same directory when tested. If you develop them in a Windows environment, you will produce DOS files, which you must convert to UNIX files and CHECK THAT THEY RUN correctly on the Linux machines before hand in.

What to hand-in: You should use the C-BASS system to hand in a tar archive called comp1005.tar containing the following files:

1. Three of your web pages named page1.html, page2.htm and page3.html.
2. Copies of the two scripts aa1 and ss1.
3. A file called aa1_output containing an example of the output from ss1 when run on the web pages you hand in.
4. A file called report.tex containing the LaTeX source of your report.
5. A file called report.pdf containing a pdf version of your report.

Assessment is based on your use of the tools, rather than the particular content of the web pages. Do not hand in more than three web pages and do not hand in any large web pages. Use small purpose-designed web pages to test the functionality.

For hand-in via the C-BASS system, you will need to find out for yourself how to use tar, and this is part of the assignment for which marks will be awarded. You might find it sensible before you submit it, to check your tar file just unpacks just your files (and does not, for example, insert an extra directory containing them).

Originality of Work: You are required to be familiar with the University's policy on plagiarism as stated in the Academic Integrity Regulations and Statement at http://www.calendar.soton.ac.uk/sectionIV/.