COMP3310/6331 Assignment 2 – Crawling the Web

Intro:

- This assignment is worth 10% of the final mark
- It is due by 17:00 Mon 23 April AEDT
- Late submissions will not be accepted, except in special circumstances
    - Extensions must be requested well before the due date, via the course convenor, with appropriate evidence.

Assignment 2

This is a coding assignment, to enhance and check your network programming skills.

A web-crawler, or 'spider' is a tool that scans/indexes/searches a website by opening the first page, parsing it for further links, and then recursively working its way down the set of links it finds to download some (usually bounded) number of pages. Many search engines and archives run their own spiders.

For this assignment, you need to write a web crawler in C or Java, without the use of any external (especially web/http/html-related) libraries. Your code should compile standalone on the CSIT Lab machines. It must open sockets in the standard way (Lab 2), make appropriate and correctly-formed HTTP/1.0 (RFC1945) requests on its own, and capture the results.

Your code will be tested against a friendly assessment server. There will be less than 100 pages on the site.

Your code needs to report:

1. The number of pages
2. The largest page (and size)
3. The most-recently modified page (and its date/time)
4. A list of invalid pages (not) found (404)
5. A list of redirected pages found (30x) and where they redirect to

The HTML on the friendly assessment server is guaranteed to be minimalist, no JS, no CSS, no external images, etc. Links will use a standard html *label* form, with absolute (http://host/path/file) URL forms to keep parsing simple.

Your code also needs to behave nicely – it must not make more than 1 request per 2 seconds. The server may generate a 503 error if your code exceeds that rate, resulting in lost marks.

It is your choice how you want to crawl the site and undertake the above analysis, there are a few ways you can approach it.

You need to submit your code, together with a Makefile (or Java equivalent) that has a 'make' and 'make run' targets to execute your code. Your submission must be a zip file, packaging everything as needed.

There are many existing web-crawling tools and libraries out there, many of them with source. While educational for you, the assessors know they exist and will be checking your code against them.

Your code will be assessed on

- correctness (what it reports),
- performance (with the pacing constraint above),
- code clarity and style, and
- documentation (comments).

You should be able to test your code against any website you like, though most sites today have complex html/css/js pages that make parsing harder. We are setting up a friendly test server at http://3310exp.hopto.org:9780/ with a small number of simple pages for you that will mimic the assessment server. An announcement will be posted to wattle when that is completed in the next few days.