# Project #1: Threads and Sockets

Due: April 14, 2006 11:59pm

## Overview

For this assignment you will implement three variations of a simplified `wget` program. These programs will allow a user to non-interactively download a file from a website. The variations will extend the basic functionality to allow for simultaneous retrieval of multiple files and to allow for retrieval of the same file from multiple sites.

## Part 1: `SimpleWebGet`

The `SimpleWebGet` program takes a URL as its single command-line argument. This URL is meant to represent a resource accessible via the http protocol. This program must contact the specified server (at port 80, unless an alternate port is given) and download the specified file. To simplify things, you may use the `java.net.URL` class to break the URL into its components (if a file name is not given, then use `index.html`). You must, however, write all networking, using sockets, yourself.

To retrieve the file, as in lab #1, your program must send a `GET` request. The server will then send a reply. Each reply begins with a header. The header contains information identifying the type of response, the type of any attached content, and the length of that content (among other possible options). The header ends with a blank line. Each line of the header ends with `\r\n` (though your program should still work if only a `\n` is present). The following is an example of a reply to a valid request (with the carriage-return, line-feed explicitly shown):

```
HTTP/1.0 200 OK\r\n
Content-Type: text/html\r\n
Content-Length: 5686\r\n
\r\n
<---- contents here ---->
```

Your program must behave as follows:

1. Parse the URL.
2. Connect to the specified server.
3. Send a `GET` request.
4. Read the reply header.
5. If the reply is of type `200`, then read the contents (which are just bytes) of the file and store them locally in a file of the same name (e.g., if the URL was `http://www.csc.calpoly.edu/ akeen/index.html`, then the local copy should be in a file named `index.html`).
6. If the reply is an error, then report the error and exit.

Be certain that your program can properly download binary (i.e., non-text) files. For example, retrieve this handout and check that it perfectly matches that retrieved by a web browser.

## Part 2: MultiWebGet

The `MultiWebGet` program works much like the `SimpleWebGet` program, but allows for the simultaneous retrieval of multiple files. To this end, `MultiWebGet` accepts multiple URLs on the command-line and spawns a separate thread to download each file.

Since the program will continue as long as any file is being downloaded, it must report when each file finishes (or if an error occurs). Specifically, an execution of your program might look as follows.

```
% java MultiWebGet http://www.csc.calpoly.edu/index.html http://www.csc.calpoly.edu/monkey.html
http://www.csc.calpoly.edu/monkey.html ==> 404 Not Found
http://www.csc.calpoly.edu/index.html ==> Complete
%
```

## Part 3: ParallelWebGet

The `ParallelWebGet` program downloads the same file from multiple sites, stopping when the first download completes. This program is intended to be used when a file is available on many sites, but when the time to download that file varies between the sites. The program takes multiple URLs as command-line arguments, starts individual threads to download the file from each site, and terminates as soon as one download successfully completes.

Again, your program should report progress as in Part 2. Moreover, since each thread is assumed to be downloading the same file (though the program is not technically enforcing this), each thread will need to write to a separate file.

The first thread that *successfully* completes its download must terminate the other threads and then rename its temporary file to the expected file name (that of the file being downloaded).

## Handin

Submit your source files using handin on hornet/falcon. To do this, execute

**handin akeen proj1 <your files here>**

Grading will be divided as follows.

| Part | Percentage |
|---|---|
| 1 | 25 |
| 2 | 35 |
| 3 | 40 |