# Unix for Telecommunications

Portfolio Task – P-Socket-Programming
**Pass Level Task**

## I. Introduction

In this tutorial you will develop a simple C program using the Sockets API to download and parse the contents of a web page from a remote web server.

## II. Purpose

To gain and/or enhance the following practical skills:

- Basic programming skills
- Use of existing network programming tools to perform simple online tasks
- Understand the principles of communicating with remote applications

## III. Preparation

You can prepare for this exercise by reviewing the Sockets API and understanding the process of establishing TCP communications with a remote host.

## IV. Supporting Materials

There are two files available for download on Doubtfire for this task.

### A. *wwwstat.c*

A small C program. When compiled and executed, the program will take one command line parameter consisting of a URL. The program will extract the name of the server and the name of the web page from the provided parameter. The program will call a function to perform a DNS lookup of the server and determine the IP address which is returned in a sockaddr_in structure.

The functions in this program are:

*`void FillAddress(char *pcURL, struct sockaddr_in *psAddress):`*

The function takes two parameters, a string containing the name of the server and a pointer a sockaddr_in structure to populate with the IP address of the server.

A DNS lookup is performed on `pcURL`, the resultant IP address is printed to screen and used to populate `psAddress`

*`int main(int iArgC, char **ppcArgV):`*

The main program. The function parses the command line and returns an error if there is not exactly one command line parameter.

The parameter is parsed, stripping `http://` (if it exists at the start of the string) and splitting the remained into a server name and page name. If no page name is provided (eg. `http://swin.edu.au`), then a page name of '`/`' is set.

The program calls `FillAddress()` to get the IP address of the server name.

### B. *Makefile*

A file to allow you to execute the make command, compiling `wwwstat.c` to create the executable `wwwstat`

## V. Tutorial Task

You will be extending the provided `wwwstat.c` program to perform the following tasks.

*A. Connect to the web server*

Use the Sockets API to establish a TCP connection to the IP address returned by the `FillAddress()` function on `WWW_PORT` (port 80)

*B. Issue a HTTP request to download the specified page*

Send the HTTP command to request the nominated page (`pcPage`) from the web server

*C. Download the web page contents*

Read from the TCP socket in a loop until the entire page contents are retrieved
*Hint: Print the downloaded string to screen while it is being fetched*

*D. Save the downloaded page to disk*

Modify the program to open a file on disk. Within the read loop save the contents of the page to the file
*Hint: After the program is complete, view the saved file to confirm your program functions*

## VI. Bonus tasks

Finished early. Extend yourself by attempting the following tasks

*A. Store entire web page in a buffer in memory*

Append the downloaded text into one large memory buffer rather than save to file
*Hint: Print the entire memory buffer to screen to confirm functionality*

*B. Parse the buffer and extract all links from the HTML source*

Search the downloaded string buffer for all strings `<a href="link_text">`. When the search string is found, print the contents of `link_text` to the screen
*Hint: The `strcasestr()` C function is useful for searching strings*
*Hint: You need to search for the initial string, then search for the end string, then print the text between the two search strings*

*C. Print a list of images that need to be downloaded to display the web page*

Repeat the task above but this time search for the `<img src="image_tag">` string

## VII. Assessment

You are expected to complete your task within the allocated tutorial time. If you are unable to complete, you will need to demonstrate partial completion by the end of the class and arrange a time with your supervisor to demonstrate task completion.

*A. Assessment of Completion*

Upload your completed `wwwstat.c` file to Doubtfire as evidence within your portfolio.
Your tutor will review the functionality of your code in complete downloading of a test web page. You will also be required to discuss your code functionality with the tutor to ensure your understanding.

*B. Completion of task in Doubtfire*

When your tutor assesses you as having both completed the understood the task, the task will be marked as complete within Doubtfire