

# ***Computer Science 159.335 - Operating Systems and Concurrent Programming***

## ***Assignment 1 - Due 17<sup>th</sup> August 2015***

Lecturer: Dr. Martin Johnson

Telephone: 4140800 ext 43142

Office: INMS 3.24

Email: [M.J.Johnson@massey.ac.nz](mailto:M.J.Johnson@massey.ac.nz)

---

The purpose of this assignment is to write a multithreaded web server.

An incomplete C program is available (for windows and unix) on the stream site, the program listens on port 3490 for web page requests and sends a header but does not send the requested page. The main loop is as follows:

```
while(1) {
    fd=accept(s,NULL,NULL);           // wait for a request
    n=recv(fd,data,512,0);            // receive the request using fd
    data[n]=0;                        // NUL terminate it
    sscanf(data,"GET /%s ",filename); // get the name of the file
    f=fopen(filename,"rb");           // open the file (might be binary)
    send(fd,header,strlen(header),0); // send the header
    //
    // send the file
    //
    closesocket(fd);                  // close the socket
}
```

Don't worry about how this works, all you need to use is the send function and C file input (use fread or fgetc, NOT fgets)

```
int send(int s, const void *msg, int len, unsigned int flags);
s      - a socket(you will use fd)
msg    - data to send
len    - number of bytes of data
flags  - should be 0.
```

Finish the program so that it sends the requested file from the current directory through the socket- fd.

You can test your program using any html or text file. Start a web browser and go to <http://127.0.0.1:3490/index.html>.

You may have to disable proxies for this to work in the labs.

If you want, you can add extra error checking to the program to deal with bad requests.

Test the program by requesting a large file. While the file is being sent, use a different browser window to make another request, the second request should have to wait for the first one to finish.

Rewrite the program using threads so that it can reply to multiple requests at the same time. Do not use any global variables.

Submit your source code (the multithreaded version) electronically via stream. Your submission must be a single source file written in ANSI C or C++. Your program should include comments as documentation.

No group submissions are allowed. This assignment will be worth 10% of the complete paper.

Marks will be subtracted for obvious copying and for late submissions.

*M Johnson 2015*