COMP 2406

Assignment 2

Muhammad Mustafa

100823576

2nd January 2015

……....

Questions:

1. [4] Explain how **and why** the behavior of tinywebserver changes when you delete the following lines:
   - Line 50: 'html': 'text/html',
   - Line 64: docroot: '.'
   - Line 95: return response.end();
   - Line 126: requestpath += options.index;
2. [1] Who calls exists_callback() defined on lines 115-121? When is this call made?
3. [1] What does the path.join() call do on line 139? Why call this method rather than implenting this functionality inline?
4. [1] When serve_file() returns, what work has it accomplished? Specifically what data (if any) has been returned to the requesting web client? Why?
5. [1] What is the purpose of the code on lines 150-156 (the else clause of the "err != null" if test)?
6. [1] How can you access the incoming HTTP request headers in tinywebserver?
7. [1] Which part of tinywebserver would you change to add a new HTTP response header?
8. **[BONUS 1]** What do you find most confusing about the material covered so far in COMP 2406? Note you do NOT get a bonus mark for saying "nothing"!

……

Answers:

1. i. Line 50 is a MIME type which basically guides the web browser in how to process and display the content. Hence removing it will not specify weither the page is to be served as a text/html or text/plain.

   ii. Line 64 specifies where the website files are located e.g index.html. Hence removing that line will not indicate where the resource files are located and we will get a 404 error.

iii. If Line 95 is removed the page will be served infinitely as line 95 tells the client to break after it is done being served.

iv. Removing Line 126 will not add index.html to the requested path. As a result the cline will keep waiting to be served a page which will not be served.

2. Exists_callback() is called by return_index. This call is made when the request_handler needs to check if the page exists, if not then a 404 error is passed.
3. Path.join() will bind together the index file and then request url. It is used rather than implementing this functionality in line because it is much more robust and automatically corrects the format based on the operating system manual.
4. Serve_file() reads the request and checks if any errors had occurred. If so it will return to the client sending header information with status code 500 and error messages otherwise header information will be sent with a 200 status code.
5. Within the else clause, stat represents the existence of a file, thus the else is checking for if the stat exists and is a directory then it passes to return_index to serve the file with the directory. If not then the serve_file is called directly to serve the particular file as stat exists as a file itself.
6. res.headers in http.request(options(options, [callback]) allows access to incoming HTTP request headers.
7. The “writeHead” function in var respond function. We will add conditions to the request.method and writeHead will allow adding new HTTP Response header.
8. I still not used to the java script and node syntax and am having a hard time understanding reading and understanding just the code aspect of the course.

……...- END -……