Continuous Assessment

Socket Programming and Report

Marks: 100% of CA Marks[i]

1. There are a number of parts to be completed for this assignment:

   a. Programme a client and server using BSD sockets library in C/ C ++ employing the more advanced techniques explained and analysed in the Steven's text, Unix Network Programming Vol 1, which is seminal to you gaining the knowledge to apply these techniques to the problem domain. 30%

      i. This work will be completed weekly in labs and thereafter in your own time as you see fit. Cooperation and discussion is encouraged in the programming tasks and to this end I will assemble loose teams[1] that may adopt agile type methods to drive research, learning and problem solving forward during lab sessions. **However any form of plagiarism will incur penalties and may involve disciplinary proceedings, especially if discovered in the report element**.

      ii. The completed work will be an online, text based version of the popular game ***Hangman***.

   b. Write an individual report that reflects and analyses your learning and research, highlighting the knowledge and insights you gained in completing the programming tasks. 70% (20% and 50%)

      i. In earlier semesters on this stream, in modules delivered in year 2 and 3, you completed weekly reports on lab work which were reflective reports on the progress you were making in learning. While the focus of this year's report remains the same, i.e. the focus is on what you learn, the approach to completing the task is not as fine grained. Instead you will reflect in one report over the process as a whole somewhat like the project work you have completed to date, i.e. your 3rd year project. There will be a requirement to submit a draft of your report, i.e. a work in progress, during the semester and this submission will attract a portion of the mark allocated for the report.

[1] See table below for assigned teams.

2. The programming tasks are as follows:

   i. Redesign, comment and code the *client* to access data from the keyboard and the *server* correctly, i.e. the *client* will handle asynchronous data inputs from the network/ server and from the keyboard/ user correctly.

   ii. Alter the server to accept concurrent requests:

      (1) Using *fork()* and child processes;

      (2) Using *select()* within one process.

   iii. Design, code and comment:

      (1) The server to operate using Datagram sockets and UDP.

      (2) The client to operate using Datagram sockets and UDP.

   iv. All servers and clients should be able to handle a dual TCP/ IP stack, i.e. IPv4 and IPv6. Furthermore the client should handle naming and the DNS.

   v. Design documentation should include pseudo code which should act as a basis for liberal commentary in your listings. It should be included in the report document as an appendix.

   vi. Every effort should be made to facilitate code reuse and a modular design, e.g. the bundling of error messaging into a separate programme listing as Donahoo manages with his DieWithMessage.c.

   vii. You may consider using a *Git* repository for version control and code sharing with the lecturer.

3. The report must contain <u>**2500 words**</u> in its body. Code snippets, design documentation etc. should be included as required in appendices, outside the body of the report. **Appendices are NOT included in the calculation of the minimum word requirement**.

4. Upload all the teams programme listings tarred and zipped to Moodle, using your team name in the name of the compressed archive and thereafter the individual draft and completed reports to Turnitin as follows:

   a. Individual Draft report on or before 23.59, Sunday, 16th Oct 2016. 20%

   b. Individual Final report and team code on or before 23.59, Sunday, 27th Nov 2016. 80%

5. Hand in a typed and stapled, or bound, copy of you report along with all appendices and

programme listings at the first class after the submission date. Include also a signed declaration as to the originality of your submissions.

6. The following are the team assignments. Please organise and seat yourselves in class in the below configuration.

Team 1/ Row 1	Declan Cordial	James Walsh	John Fogarty
Team 2/ Row 2	Ciaran Clerkin	Caroline Cagney	Cian Fitzpatrick
Team 3/ Row 3	Mark Douglas	Conor Holmes	Shane Hayes

[i] CA/ exam ratio is 30/70, i.e. it is heavily weighted towards the exam. Therefore this is the only element of CA.