

***Continuous Assessment***  
***Socket Programming and Report***  
***Marks: 100% of CA Marks<sup>i</sup>***

1. There are a number of parts to be completed for this assignment:
  - a. Design and programme a variety of client/ servers using the BSD sockets library in C on Ubuntu Linux, employing the more advanced techniques explained and analysed in the Steven's text, Unix Network Programming Vol 1. This text is fundamental reading which will allow you to understand these advanced techniques and apply them to the problem domain. **30%**
    - i. These tasks maybe completed weekly in labs and thereafter in your own independent study time, as you see fit. Cooperation and discussion is encouraged in the design and programming tasks, and, to this end, the class is divided into teams<sup>1</sup>. The teams should adopt agile methodologies to research, design and develop their solutions.
    - ii. The required outcomes are a variety of online, text-based versions of the popular game ***Hangman***.
    - iii. The specific programming tasks are as follows:
      - (1) Redesign, comment and code the *client* to access data from the keyboard and the *server/ network* correctly, i.e. the *client* will handle asynchronous data inputs from the network/ server and from the keyboard/ user correctly.
      - (2) Alter the original TCP server to accept concurrent requests:
        - (a) Using *fork()* and child processes, ensuring that there are no memory leaks;
        - (b) Using *select()* within one process.
      - (3) Design, code and comment:
        - (a) The server to operate using Datagram sockets and UDP.
        - (b) The client to operate using Datagram sockets and UDP.

---

<sup>1</sup> See table below for assigned teams.

- (4) All servers and clients should be able to handle a dual TCP/ IP stack, i.e. IPv4 and IPv6. Furthermore the client must be designed and programmed to resolve DNS names.
- (5) Design documentation should include pseudo code, which will act as a basis for liberal commentary in the code listings. All such documentaion should be included in the report document, as an appendix.
- (6) Teams should make every effort to facilitate code reuse using a modular design and implementation of their functionality, e.g. the bundling of error messaging and socket setup into a separate programme listing as Donahoo manages with his DieWithMessage.c, TCPServerUtility.c and TCPClientUtility.c.
  - (a) Ultimately, the goal of such a project would be the creation of a network library designed more generally for games of this genre. However, this is not a requirement in this element of CA, as time will not permit such ambition.
- (7) You may consider using a *Git* repository for version control and code sharing among team members, and also with the lecturer.

**iv. Designate one team member to upload all the teams programme listings, tarred and gzipped, to Moodle by 23.59, Sun, 14 Apr 2019.**

The designated member must include the team name in the name of the compressed archive and in the names of all the code listings, e.g. TeamACodeSubmission.tar.gz and TeamAForkSigServer.c etc.

- b. Write an individual report that reflects on and analyses your research and learning, **70%**  
highlighting the knowledge and insights you gained in completing the programming tasks.
  - i. **Any form of plagiarism in the report element will incur penalties, up to and inclusive of an award of 0 marks, and if deemed to be of a serious nature it may involve the student in disciplinary proceedings.**
  - ii. In earlier semesters on this stream, i.e. in modules delivered in year 1, 2 and 4, you completed weekly reports on the lab work, which were reflective reports on the progress you were making in your learning. While the focus of this year's report remains the same, i.e. the focus is on research undertaken and the

knowledge you gain, the approach to completing the task is not as fine grained. Instead, you will reflect in a single report on the task as a whole, somewhat like the project report you completed in the 3<sup>rd</sup> year project module.

There is 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.

- iii. The final report must contain **at least 2500 words** in its body. Code snippets, design documentation etc. should be included as required in appendices, outside the body of the report.

If you wish to include some small snippet of code in the body, convert it to a *.jpeg/.png* and include it this form, and this form only.

**Appendices** are NOT included in the calculation of the minimum word requirement.

- iv. **Individual Draft report: upload as a MS Word doc, on or before 23.59, Sunday, 24 Feb 2019. (10%)**

- v. **Individual Final report: upload as a MS Word doc, on or before 23.59, Sunday, 14 April 2019. (60%)**

- 2. 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.

**The following are the team assignments. You may wish to reorganise the seating plan in the class to suit the team configurations.**

	<b>Student No.</b>	<b>Student Name</b>	<b>Status</b>	<b>Course</b>	<b>Team</b>
5	K00211819	Dunne, Nathan	EL	LC_KGDVM_KTH	<b>A</b>
10	K00204708	Mc Grath, Eoghain	EL	LC_KGDVM_KTH	<b>A</b>
4	K00200404	Dineen, Steven	RG	LC_KGDVM_KTH	<b>A</b>
2	K00198630	Barnett, Emily	RG	LC_KGDVM_KTH	<b>B</b>
8	K00204391	Garrahy, Conor	TR	LC_KGDVM_KTH	<b>B</b>
9	K00200042	Mc Allister, Alexander M	EL	LC_KGDVM_KTH	<b>B</b>
13	P11011180	O'Brien, Luke	EL	LC_KGDVM_KTH	<b>C</b>
17	K00209111	Walsh, Robert E	EL	LC_KGDVM_KTH	<b>C</b>
6	K00193187	Ferris, Tom	TR	LC_KGDVM_KTH	<b>C</b>
15	K00207923	Quirke, Joseph	TR	LC_KGDVM_KTH	<b>D</b>
7	K00213083	Fogarty, James	TR	LC_KGDVM_KTH	<b>D</b>
16	K00207682	Ross, Daniel	TR	LC_KGDVM_KTH	<b>D</b>
1	K00203841	Ansell, Charlie P	RG	LC_KGDVM_KTH	<b>E</b>
12	K00211758	Murphy, Niall	TR	LC_KGDVM_KTH	<b>E</b>
11	K00205588	Morrissey, Alan	TR	LC_KGDVM_KTH	<b>E</b>

---

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