|  |  |  |
| --- | --- | --- |
|  | **FACULTY of COMPUTING, ENGINEERING & SCIENCE** | Final mark awarded:\_\_\_\_\_ |

**Assessment Cover Sheet and Feedback Form 2016/17**

|  |  |  |  |
| --- | --- | --- | --- |
| Module Code:CS3S601 | Module Title: Concurrent and Parallel Programming | | Module Lecturer: C. W. Morris |
| Assessment Title and Tasks: Threads | | | Assessment No. 1 of 2 |
| No. of pages submitted in total including this page:  Completed by student | | | Word Count of submission  (if applicable): Completed by student |
| Date Set:  25/11/16 | | Submission Date: 13/01/17 | Return Date: Usually within 20 working days of submission |

|  |  |
| --- | --- |
| ***Part A: Record of Submission (to be completed by Student)*** | |
| **Extenuating Circumstances**  If there are any exceptional circumstances that may have affected your ability to undertake or submit this assignment, make sure you contact the Advice Centre on your campus prior to your submission deadline. | |
| **Fit to sit policy**:  The University operates a fit to sit policy whereby you, in submitting or presenting yourself for an assessment, are declaring that you are fit to sit the assessment. You cannot subsequently claim that your performance in this assessment was affected by extenuating factors. | |
| **Plagiarism and Unfair Practice Declaration:**  By submitting this assessment, you declare that it is your own work and that the sources of information and material you have used (including the internet) have been fully identified and properly acknowledged as required[[1]](#footnote-1). Additionally, the work presented has not been submitted for any other assessment. You also understand that the Faculty reserves the right to investigate allegations of plagiarism or unfair practice which, if proven, could result in a fail in this assessment and may affect your progress. | |
| **Intellectual Property and Retention of Student Work:**  You understand that the University will retain a copy of any assessments submitted electronically for evidence and quality assurance purposes; requests for the removal of assessments will only be considered if the work contains information that is either politically and/or commercially sensitive (as determined by the University) and where requests are made by the relevant module leader or dissertation supervisor. | |
| **Details of Submission:**  Note that all work handed in after the submission date and within 5 working days will be capped at 40%[[2]](#footnote-2). No marks will be awarded if the assessment is submitted after the late submission date unless extenuating circumstances are applied for and accepted (Advice Centre to be consulted). | |
| You are required to acknowledge that you have read the above statements by writing your student number(s) in the box: | Student Number(s): |

**IT IS YOUR RESPONSIBILITY TO KEEP RECORDS OF ALL WORK SUBMITTED**

|  |
| --- |
| **Part B: Marking and Assessment**  **(to be completed by Module Lecturer)** |
| This assignment will be marked out of 100%  This assignment contributes to 25% of the total module marks.  This assignment is bonded. |
| **Assessment Task:**  This coursework will allow you to investigate some of the issues involved in writing threaded programs.  You are to write a series of simple programs in C/C++ that will run under Linux, to illustrate how multithreading may be used to partition a processing task. **Designs should be presented and all code must be fully commented and handed in as part of the deliverable.**  Consider a 2 D *data array* whose contents need to be processed.  The array consists of values representing the height of a terrain at that point.  Create two new arrays to hold the results of calculations.  *Result array one* is to hold the Euclidean distance between each point and the next point in the row.  *Result array two* is to hold the angle of the slope between the point and its neighbour in the row calculated using arcsin.   1. Set up the first array and write a sequential program that accesses the data array and populates the result arrays. **Provide evidence that the calculations used are correct.** 2. Create a second program that it is still sequential but accesses the data array by a series of calls to a single function. Each call to that function will process a number of rows of the array as defined by a parameter passed to the function. 3. Write a third version of the program that will allow the processing to be carried out in one or more threads. Each thread should access a different set of rows of the array as in 2. above so that locking etc. is not an issue. Your program must be able to run with different numbers of threads. 4. Investigate how you might obtain run time information for the implementations and **report on run times of each of your implementations. Experiment with your implementations on machines with different numbers of processors/cores and explain all results.** 5. **Write a conclusion to your investigations. This should include**  * **an overview of threads and processes** * **expected gains in threading in a typical application** * **expected link between numbers of cores/processes and performance** * **narrative regarding your results and development include short discussion on how code would need to change if locking were required**  1. You may need to do a short demonstration of your code in a lab session or at some mutually agreed time. **This is a mandatory part of the assessment**.   **Submission Instructions**  The report should be a single document, word processed in Word format, and it should present your findings clearly and concisely. Take care that code is neatly laid out in the report – correctly formatted and with all code in Courier New font.  You must submit **only** an electronic version of the coursework (via Blackboard) which may be submitted to a plagiarism detection system.  All sources of information **must** be fully referenced and full URLs of all Internet sources must be given. Use Harvard referencing style. |
| **Learning Outcomes to be assessed** (as specified in the validated module descriptor <http://icis.southwales.ac.uk>):  To critically analyse the advantages and problems inherent in concurrent and parallel programming.  To be able to implement and evaluate the performance of an application using concurrent and parallel programming techniques. |
| **Grading Criteria:**   |  |  | | --- | --- | | **Performance Level** | **Criteria** | | Fail  (< 40%) | Unacceptable documentation has been handed in. Poor presentation of material. Little working code is available. Understanding of threading has not been demonstrated. | | 3rd Class  (40% - 49%) | A basic level of understanding of the application of threading. Basic sequential code presented and some effort to produce threaded version. Code mainly working but there may be some issues with efficiency of the code and your solutions. Documentation is minimal and little experimentation is presented. | | Lower 2nd Class  (50% - 59%) | A good report showing evidence of a good understanding of the issues in multithreading. Basic working multithreaded code is presented. Some effort at experimentation is evident. Documentation of the code is adequate and in general explains what has been done. There may be some areas where code is not as efficient as it could be. | | Upper 2nd Class  (60% - 69%) | A well-presented report that shows an excellent understanding of most of the issues. Working code is presented and good performance data has been obtained. Designs are presented and code is generally quite efficient. | | 1st Class  (70% +) | Excellent understanding of all issues is evident. Good quality code presented. Code is efficient and documentation explains the operation of the code to a high level. Performance data has been gathered and explained. Presentation of the results is well thought out and the results obtained are explained clearly. | |

|  |  |  |
| --- | --- | --- |
| **Indicative Marking Scheme** | **Marks Available** | **Marks Awarded** |
| Basic sequential code | 10 |  |
| Modified sequential code | 10 |  |
| Threaded version | 40 |  |
| Performance issues investigated | 20 |  |
| Report, discussion and conclusions. | 20 |  |

|  |
| --- |
| **Part B: Marking and Assessment**  **(to be completed by Module Lecturer)** |
| **Assessment Task:** |
| **Learning Outcomes to be assessed** (as specified in the validated module descriptor <https://icis.southwales.ac.uk/> ): |
| **Grading Criteria:** |

|  |  |  |
| --- | --- | --- |
| **Feedback/feed-forward** (linked to assessment criteria):   * Areas where you have done well: * Feedback from this assessment to help you to improve future assessments: * Other comments | | |
| **Mark:** | **Marker’s Signature:** | **Date:** |
| **Work on this module has been marked, double marked/moderated in**  **line with USW procedures.** | | |
| *Provisional mark only: subject to change and/or confirmation by the Assessment Board* | | |

|  |  |
| --- | --- |
| **Part C: Reflections on Assessment**  **(to be completed by student – optional)** | |
| **Use of previous feedback:**  In this assessment, I have taken/took note of the following points in feedback on previous work: | |
| **Please indicate which of the following you feel/felt applies/applied to your submitted work**   * A reasonable attempt. I could have developed some of the   sections further.   * A good attempt, displaying my understanding and learning, with   analysis in some parts.   * A very good attempt. The work demonstrates my clear   understanding of the learning supported by relevant literature and scholarly work with good analysis and evaluation.   * An excellent attempt, with clear application of literature and   scholarly work, demonstrating significant analysis and evaluation. | |
| **What I found most difficult about this assessment:** |  |
| **The areas where I would value/would have valued feedback:** |  |

1. University Academic Misconduct Regulations [↑](#footnote-ref-1)
2. Information on exclusions to this rule is available from the Advice Centre at each Campus [↑](#footnote-ref-2)