**Component 1: Getting to Grips with Git (10 marks)**

Successful completion of all sub-components will be given 10 marks of the overall module marks.

**NB: This is a pass/fail component. Failure to complete any and/or all of the sub-components will result in a zero for this component.**

Please complete the following table with proofs (screenshots must show a form of identification to you or your project) and a short description of how you’ve accomplished the task.

The proofs MUST BE clear and readable without forcing a marker to zoom in to verify them. If you do not have space for a screenshot in a cell of the table simply reference it (ex: “see figure X”) and put it below the table.

Make sure to push your code to your PRIVATE repository at <https://gitlab.uwe.ac.uk/> and put myself (ty-win) and the OS delivery team as “Reporter” of your project.

The completed worksheet must be submitted under your repository.

|  |  |  |
| --- | --- | --- |
| **Operating Systems**  **Component 1** | **Student ID: 22009561**  **Name: OMAR HALASA**  **GITLAB REPOSITORY: https://gitlab.uwe.ac.uk/o2-halasa/omar-halasa** | |
|  | **Proofs and notes** | **Date** |
| **Create a project on your Gitlab account** |  | **17/10/2023** |
| **Clone your project to your local VM** | **A screenshot of a computer  Description automatically generated** | **17/10/2023** |
| **Create a file in your repository and modify it** | **A screenshot of a computer  Description automatically generated** | **17/10/2023** |
| **Use Git command to add the modified file to your commit** | **A screenshot of a computer  Description automatically generated** | **17/10/2023** |
| **Push changes to your remote repository** | **A screenshot of a computer  Description automatically generated** | **17/10/2023** |
| **Fork a classmate project (make sure to indicate your classmate repository URL)** |  |  |

**Component 2: C programming refresher (10 marks)**

Please complete the following table with proofs (screenshots must show a form of identification to you or your project) and a short description of how you’ve accomplished the task.

The proofs MUST BE clear and readable without forcing a marker to zoom in to verify them. If you do not have space for a screenshot in a cell of the table simply reference it (ex: “see figure X”) and put it below the table.

Make sure to push your code to your PRIVATE repository at <https://gitlab.uwe.ac.uk/> and put myself (ty-win) and the OS delivery team as “Reporter” of your project.

The completed worksheet along with the source code for each sub-component must be submitted under your repository.

**NB: This is a pass/fail component. Failure to complete any and/or all of the sub-components will result in a zero for this component.**

|  |  |  |
| --- | --- | --- |
| **Operating Systems**  **Component 2** | **Student ID: 22009561**  **Name: OMAR HALASA**  **C Programming Refresher GitLab repo: https://gitlab.uwe.ac.uk/o2-halasa/omar-halasa** | |
|  | **Proofs and notes** | **Date** |
| **Using only loops write a C program which produces the following pattern:**  **\***  **\* \***  **\* \* \***  **\* \***  **\***  **The user should be able to specify the number of rows he/she wants** | **For even numbers I thought of 3 ways. To print all rows except the last star or, to duplicate the middle row, or to print 1 less row. I did the third way because duplicating the middle didn’t look pleasant or professional in a sense, and these are the outputs.A screenshot of a computer  Description automatically generated**  **A screenshot of a computer  Description automatically generated** | **25/10/2023** |
| **Write a small C program which does the following:**   1. **Create a text file;** 2. **Write some text into it;**   **Opens the file and outputs its content on the console.** | **This picture shows that the Desktop doesn’t have any text files in it.**    **This picture shows that after running the code in the Linux terminal a file of type txt was created and appeared on the Desktop, Hello world was written in the file and that was outputted in the Linux terminal.**  **A screenshot of a computer  Description automatically generated** | **02/11/2023** |
| **Write a small number guessing game (between 1 and 5000) using C. It needs to have the following features:**   1. **Console-based;** 2. **Needs to have a “scoreboard”**   **The user should be able to determine when he/she wants to end the game** | **The first and second game I (user) played until the number was guessed, each game has a different number as proved in the screenshots.**  **A screenshot of a computer  Description automatically generated**  **In the third game I (user) guessed a few guesses and decided to end the game so zero was guessed and then played another game (4th game) but exited straight away with the first guess being zero as shown on the scoreboard both games show no guessed and show that the game was ended, as for the first 2 games it shows on which try the user guessed correctly.**  **A screenshot of a computer  Description automatically generated** | **04/11/2023** |