University of Lincoln
School of Computer Science
CMP9133M – Advanced Programming
Workshop 16

Task (assessed): Implement a program to process a large dataset concurrently using multiple threads and a shared resource, and then merge the results.

Description:

1. Create a C++ program that reads a large dataset from a file.
2. Implement a class, let's say DataProcessor, that processes the dataset (e.g., performs some computations or transformations on the data) in parallel using multiple threads.
3. Divide the dataset into equal-sized chunks and distribute them among threads for processing.
4. Use locks or other synchronization mechanisms to manage shared resources, such as the output or intermediate result storage.
5. Each thread should process its assigned chunk independently and store the results in a shared data structure.
6. Once all threads have completed their processing, merge the results from each thread into the final result.
7. Display or save the final result.

Example Usage:

```
./data_processing_program dataset.txt output.txt 4
```

This example command runs the program data_processing_program, processes the data in dataset.txt using 4 threads, and saves the final result in output.txt.

By completing this task, students will gain experience in:

- Reading input and managing large datasets in a multithreaded environment.
- Synchronizing access to shared resources using locks or synchronization primitives.
- Efficiently partitioning the dataset for parallel processing.
- Merging results from multiple threads into a final output.
- Handling I/O operations and managing file output.