SIT323 Practical Software Development, Trimester 2, 2019

Week 9 – Practical 8

Importing Configuration Files and 2D Array of Run-time Data

Introduction

This week you will attempt to import a remote configuration file from Azure into your C# or Excel software. The URLs for configuration files are:

- https://sit323sa.blob.core.windows.net/at2/TestA.txt
- https://sit323sa.blob.core.windows.net/at2/TestB.txt
- https://sit323sa.blob.core.windows.net/at2/TestC.txt

Task 1

Ensure that your software can import/read remote configuration files.

- For Excel, slide 8 of last week's lecture (Class 8) will help you import a remote configuration file using a URL.
- For C#, slides 10 and 13 of last week's lecture (Class 8) will help you get the URL from the user and read a remote configuration file using a URL.

Task 2

Set up some data structures to compute the total run-time of 0 or more tasks on each processor. For each processor, ensure that the total run-time of task on that processor is not greater than the overall program run-time.

- For Excel, the video called "**Creating Allocations and HTML files"** will help you organise your worksheets. This video can be found in:
  Resources > Assessment > Programming Task 2 > Videos Excel

- For C#, although there are many ways to design and develop your software for AT2, you might consider starting by creating a 2D array of doubles to store run-time values.
  - Initially, each element in this array of N rows and M columns can have a 0 value, where N is the number of processors and M is the number of tasks.

    For example, a 2D array for 3 processors and 8 tasks is depicted below.

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

  - When your allocation algorithm places a task onto a processor, you can update that 2D array by placing a task's run-time value into the appropriate array element.

For example, the following table depicts that task 1 has been allocated to processor 1, and the run-time of T1 on P1 is 1.23 seconds.

1.23	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

- After placing a run-time value into the 2D array, you might prefer to update the total run-time for that processor too. Take care not to breach the program duration. You might prefer to create a 1D array of double for processor run-times.

  For example, the following depicts that T1 and T2 have been allocated to P1, T3 has been allocated to P2, T4 to T8 have not yet been allocated, and the total run-times for the processors are presented on the right hand side.

								total
1.23	1.5	0	0	0	0	0	0	2.73
0	0	1.75	0	0	0	0	0	1.75
0	0	0	0	0	0	0	0	0

Task 3

Continue working on designing your allocation algorithm, and asking questions during the practical session.

Remember your allocation algorithm can be a **mixture of many algorithms**, not just a pure Greedy algorithm.