Unit: COMP2006

Operating Systems Assignment

1/05/2017

Author: Jason Gilbert
Student ID: XXXXXXXX

Overview

There is a total of 4 shared resources that exist in the mssv program:

- Sudoku Grid (Buffer1)
- Task Log (Buffer2)
- Task Counter (Counter)
- Log File (The file that logs invalid sub-grids)

Each resource is stored in the heap (if in a threaded solution) or in an unnamed shared memory segment (if in a multi-process solution).

Child Tasks

As each of these four resources are shared by the parent, they are accessible to all child tasks, whether it be a thread or child process.

Mutual Exclusion

Out of the four shared resources, only the 'Task Counter' and 'Log File' resources have to deal with multiple incoming writes. The 'Sudoku Grid' (after being initialised) is only read from. And the 'Task Log' (Buffer2), while technically having to sustain multiple writes, each write is only targeted at a specific subsection of the array, and that specific write only occurs once.

As such, the 'Task Counter' and 'Log File' resources have an associated lock stored alongside them. A semaphore lock is used in the multi-process solution and a mutex lock is used in the threaded solution. And since the lock is stored alongside the resource, it too will be shared.

When a child process or thread needs to write to the 'Task Counter' or 'Log File' resources, it will use it's associated 'write method' that will write to it in a safe manner:

```
writeLocal_LogFile()    // uses mutex lock
writeShared_LogFile()   // uses semaphore lock

writeLocal_TaskCtr()    // uses mutex lock
writeShared_TaskCtr()   // uses semaphore lock
```

The general format for what each of these methods to is as follows:

- Acquire the lock for this resource
- Write to the resource
- Release the lock for this resource

As these writes are only triggered by the child tasks, and all reads from these resources take place after all the child tasks terminate. No locks are needed to be acquired to read from the resources.

Jason Gilbert (XXXXXXXX)

Testing

Testing was performed by iteratively inputting a variety of Sudoku grids with known invalid sub-grids and checking if the output matches. Mutual exclusion was tested by seeing if not releasing locks would induce a dead lock, (which was the case).

Sample inputs & outputs

See attached files:

Test_threaded_01.txt
Test_threaded_02.txt
Test_forked_01.txt
Test_forked_02.txt

A Note on terminology and output format

In the code; Rows, Columns & sub-grids are referred to as 'Shapes'. And the positions (or index) of these shapes within the grid are referred to as 'zones'.

'zones' are a base 0 index system, but are converted to base 1 on output. To clarify;

- Rows are labelled from 1 to 9 (left to right).
- Columns are labelled from 1 to 9 (top to bottom).
- Sub-grid boxes are labelled from 1 to 9 (left to right, top to bottom).