OS Project 2

Share Memory Management -Number Calculation

Deadline: 2014/12/10 23:59, No late submission accepted

Description

The goal of this project is to practice how to use multi-thread/multi-process mechanism to speed up your program, and to manage the share memory to protect critical section. You will be given 4 files containing lots of numbers, you need to calculate the sum of integers before "**wait**" term in each file, and output the added sum in "Result.txt". More details will be given in the following sections.

Implement Details

Your program should create 4 threads (or 4 processes). Each of them will read one of the 4 input files (*testdata1.txt* ~ *testdata4.txt*) given by TA. The input file will contain lots of integer numbers and "wait" terms, please refer to the example we gave for more details.

Firstly, 4 threads (processes) need to read their corresponding input file and calculate sum before "wait" term in parallel. And then four sums before "wait" in each file should be added to share memory.

Secondly, there is a fifth entity (called *Writer Entity*) which gets the sum from share memory, and writes to an output file (named "*Result.txt*").

Make an example, if four sums calculated by each file are *90, 80, 70,* and *60*, then you need write the added sum *300* to the output file "*Result.txt.*"

Output File: Result.txt

Every time the *Writer Entity* gets added sum from the share memory, it should write it to the "*Result.txt*" file and specify the number of times it has outputted. For example, when it is second time *Writer Entity* getting added sum "*300*" from share memory, it should output "*The 2 output : 300*" to "*Result.txt*".

Environment

The environment of this project is limited to the CSIE workstation. All the other environment is forbidden.

Requirements:

1. Your program should accomplish share memory usage. That means each thread/process is restricted to do operation (*Read/Write*) in a segment of memory. DO NOT let every thread/process do its stuff without communication.
2. The output file format is restricted. You will get zero, if your output file format is different from TA's.
3. You need to write annotations（註解）in your code.
4. In your report, you should write your thoughts and talk about how you accomplish share memory part in your code.

Hand in

You should upload your code, execution file and makefile to e3. Please make sure the correctness of your code and the makefile. You may get **zero** if we cannot **compile** your code.

A report of this project is also needed. **Do not copy and paste the code directly.** You can include your thoughts and the way you accomplish share memory part in your code.

All the files should be put into one single compressed folder and the folder name should be the same as your student ID (ex: 00XXXXX.rar or 00XXXXX.zip).

Scoring

The scoring of this project is as follows:

- Program correctness and time consumed (85%)
- Annotation(5%)
- Report (10%)

Reference:

Pthread API :

http://cursuri.cs.pub.ro/~apc/2003/resources/pthreads/uguide/usersgu.htm#301162

Share Memory API :

http://pubs.opengroup.org/onlinepubs/009695399/basedefs/sys/shm.h.html