

OS Project Report

9962173 林泓璋

Architecture

- 開一個檔案指標，和一個結構來存資料
- 兩個Thread競爭讀同一檔案指標
- 避免競爭的時候讀錯資料，用Mutex解決
- QuickSort資料
- 比較ID大小，最多允許10000個相同Rank
- 寫檔案

Makefile

- `CC = gcc` #use gcc as compiler
- `CFLAGS = -lpthread -Wall` #parameters
- #compile to object file and executable run
- `all:read.o`
- `$(CC) -o run read.o $(CFLAGS)`
- `read.o: read.c`

Performance

- 大部份時候Single Thread跑的時間較短，但有時候Two Thread較快，應該是誤差
- 因為我的Two Thread使用了Mutex鎖住檔案的讀取權，一次只允許一個Thread讀資料，雖是以多執行緒來寫，但是實際上是做一個Thread的動作

Performance

- Testcase with 100000 players
- `time ./a.out testcase SingleThread`
- `real 0m3.532s`
- `user 0m3.523s`
- `sys 0m0.007s`

Performance

- Testcase with 100000 players
- `time ./run testcase MultiThread`
- `real 0m3.995s`
- `user 0m3.601s`
- `sys 0m0.225s`

What I have learned?

- 第一次接觸Makefile檔案，現在比較瞭解
- 在編寫多執行緒的程式時，要注意競爭問題，才不會發生錯誤
- Mutex的用法
- 產生大的測資(In otherFiles folder: testcaseGen.c)
- Use time ./run to measure the performance
- Use time() to measure the performance