本群組計畫執行的注意事項：

1. 各組請在助教所設定的時間內完成抽籤來選定學期計畫實做的題目。

2. 抽籤後所選定的實做題目不可改變與交換。

3. 請運用上課所學的Multi-Thread Synchronization概念與技術來完成群組計畫，本群組

計畫不限定所使用的程式語言如C, C++, Java, Python, Scala等皆可，但是必須要用

Multi-Thread 的技術並透過 Semaphore, Lock, or Monitor 等 Synchronization

Primitives來完成。

4. 群組計畫分為兩部分實做部分佔85％，學期報告佔15％，測試時請各組提供。

5. 在群組計畫測試時，請明確說明你需要多少Threads以及這些Threads所扮演的角色與功

能為何？哪一些是共有資源與共享變數(Shared Variables)是需要來進行Multi-Thread

Synchronization。

6. 請將Multi-thread Synchronization的過程透過各Thread以文字顯示運作狀態與結果，

以確認Multi-Thread Synchronization的運作過程正確無誤，如此則可以得到實做部分

80％，配上述文字的說明完成正確圖形使用者介面(GUI)顯示可以得到額外實做部分20％

。另外提供額外實做與分析功能則可以再加分，例如進行Multi-Thread Multi-Core實做

與觀察並提出相關心得與評論。

7. 群組成績和個人成績的配分比是總成績的20％中群組成績佔14％，個人成績佔6％。

8. 請在本課程期末考（2016/01/12）完畢後的一星期之內（2016/01/19）完成各組的分

組測試，測試的流程是先經過助教的初步測試再由老師確認最後評量結果。

1 Berztiss, Alfs T., Synchronization of processes, Department of Computer Science, University of

Wollongong, Working Paper 82-11, 1982, 66p., http://ro.uow.edu.au/compsciwp/28

1. Readers and Writers (RW) Problem

Here one has a system of r readers and w writers that all access a common database

(or some other resource). A reader may share the resource with an unlimited number

of other readers, but a writer must be in exclusive control of the resource. We call

this the RW problem. Two additional constraints characterize variants of the problem.

Problem RW1. As soon as a writer is ready to write, no new reader should get

permission to run. Starvation of readers is a possibility here.

Problem RW2. No writer is permitted to start running if there are any waiting

readers. Here it is possible to starve the writers.