CS305 作業系統概論 Prog. #4 Deadlock Avoidance

2021.05.21

1. 作業目的

在本作業中，實作Banker’s algorithm 來瞭解deadlock avoidance的觀念。

1. 作業內容

**【醫療資源調度】**最近疫情不太平靜，於是最高醫療總署運籌帷幄，開始調兵遣將，派出許多特別任務小組緊急動員。每個特別小組會調動的醫療資源會有A~E共5種。在進行過程中，他們會向最高醫療總署請求若干資源來執行任務。他們也可能還回去一些資源。現在最高醫療總署擔心如果每個任務小組所要的醫療資源數量過多，將會造成死結，造成醫療處理上的困擾，因此現在需要有一個管理程式，利用Banker’s 演算法，來確保整體醫療處理上的順暢。

程式自檔案中讀入Available[]，Max[]，Allocation[]等陣列，然後依照演算法，判斷此時系統是否在safe state，並針對後續的requests，一一判斷它們是否可以被granted。如果request可以被許可，便將陣列內容隨之調整，再判斷後續的requests是否會使系統狀態在safe state。如果request不能被許可，就判斷這個request是否Valid（滿足Need與Available的檢查）。如果是Valid request，就將這個request放入一個等待的佇列，當有resource release時，對這個佇列當中正在等待的request按照FCFS的方式來檢查，看看是否可以被granted，如果可被granted，則加以處理，如無法granted，則處理佇列當中下一個等待的合法request。如果request不合法，就予以捨棄。這些運作的過程，都需要詳細印出。以便檢查。

例如，假設資料檔案如下， “#”表示相關的陣列或矩陣， “//”表示註解，gid是第一個數字表示任務小組ID，所有gid與資源都是0～99的整數，在request的最後一個欄位 “a” 表示需要新的allocation， “r” 表示release 資源。所有數字用一個空格隔開。資料檔中的request個數沒有限制。

#AVAILABLE

// A, B, C, D, E

0 12 5 3 10

#MAX

//gid, A, B, C, D, E

0 2 2 2 1 1

1 3 2 2 0 0

2 2 0 2 0 0

3 2 2 2 2 0

4 1 3 3 1 0

#ALLOCATION

//gid, A, B, C, D, E

0 2 2 0 1 1

1 0 2 2 0 0

2 2 0 1 0 0

3 2 0 0 0 0

4 1 1 1 1 0

#REQUEST

//gid, A, B, C, D, E, allocate/release

0 0 0 2 0 0 a

4 3 3 0 1 0 a

0 2 0 0 0 0 r

可能的輸出為

Initial state: safe, safe sequence = [0,2,1,3,4]

(0, 0, 0, 2, 0, 0): NEED OK

(0, 0, 0, 2, 0, 0): AVAILABLE OK

(0, 0, 0, 2, 0, 0): WORK= (0,12,3,3,10)

(0, 0, 0, 2, 0, 0): gid 0 finish, WORK= (2,14,5,4,11)

…

(0, 0, 0, 2, 0, 0): request granted, safe sequence = [0,2,1,3,4]

(4, 3, 3, 0, 1, 0): NEED error, request aborted

(0, 2, 0, 0, 0, 0): RELEASE granted, AVAILABLE=(2,12,3,3,10)

1. 作業要點
   1. 請注意，本作業使用的程式語言是C/C++，測試平台的作業系統： Ubuntu 20.04 64-bit。使用的編譯程式為gcc/g++ 編譯器： 9.3。其他平台或程式語言不在本次作業考慮範圍之內。如在測試平台上無法編譯與執行，都不予給分。
   2. 本作業的評分方式如下：
      1. Safety algorithm 的正確性，此部份最多可得60分。
      2. Banker’s algorithm 的正確性，此部份最多可得40分。
   3. 本作業需繳交檔案：
      1. 說明報告：檔案為docx或pdf格式。
         1. 報告中必須說明程式的設計理念、程式如何編譯，以及**如何操作**。
         2. 報告中同時必須詳細說明你完成哪些部份。如有用到特殊程式庫，請務必說明。
         3. 請務必讓助教明白如何編譯及測試你的程式。助教如果無法編譯或測試，會寄信（最多兩次）通知你說明，但每說明一次，**助教會少給你10分**。
      2. 完整原始程式碼檔案（.c 或 . cpp）。**不可含執行檔**。**助教會重新編譯你們的程式**。**請注意**：不可用 .txt檔或是 .docx檔等非正常方式繳交程式碼，如有類似情形，**助教也會扣10分**。
      3. **不可以含有病毒，如果含有病毒等惡意程式，本作業0分**。
   4. 所有相關檔案，例如報告檔、程式檔、參考資料等，請壓縮成一個壓縮檔（不可超過2MB）後上傳至portal。**請注意，不可抄襲。助教不會區分何者為原始版本，被判定抄襲者，一律0分。**
2. 繳交方式：
   1. 最終繳交時間：
      1. 電子檔在 2021.06.10以前，上傳至個人portal。如有多個檔案，將所有檔案壓縮成zip（rar 亦可）格式，然後上傳。
      2. 上傳檔名格式：「學號\_作業號碼.doc」或「學號\_作業號碼.rar」。例如：912233\_01.doc 或 912233\_01.rar。
   2. **如有違規事項者，依照課程規定處理**。
   3. 如需請假，請上portal請假，並持相關證明文件，在請假結束後的第一次上課時完成請假手續，並在一週內完成補交。補交作業將以8折計算。
   4. 老師不接受「門縫」方式繳交，助教也不接受任何作業。
3. 如有未盡事宜，將在個人portal板面公告通知。
4. If you need **any assistance in English**, please contact Prof. Yang.
5. 參考資料
   1. 課本第8章