Homework4

Object:

Implement Banker’s algorithm

Deadline :

2024.01.18 23:59 (No extension)

繳交方式:

http://dslab.csie.org/course/1121DS/index.html

Programming language:

C or C++ (gcc version 11.4.0)

作業說明:

銀行家演算法，由荷蘭計算機科學家艾茲赫尔·迪科斯彻於 1965 年設計，是一種用於操作系統中避免死結的演算法。它基於銀行的貸款系統，銀行對客戶的貸款有一定限額，客戶必須宣告他們的最大貸款需求。在計算中，這轉化為資源分配，進程必須宣告它們的最大資源需求。該演算法通過維持一種每個進程都能在現有資源下完成的狀態，以確保安全執行。它通過不進入不安全狀態來避免死結，這些狀態由每個進程是否能在可用資源下完成來確定。沒有修過作業系統的同學可以參考偉大印度人的影片

https://www.youtube.com/watch?v=2V2FfP_olaA&t=147s&ab_channel=ScholarlyThings

Input:

Line 1 : 系統中各個資源最大總量

Line 2 – Line N : process 向系統發出的資源請求

請求類型:

1. Pid in “max need of each resource” [ex P0 in 1,2,3]
   指的是一個 process（用 process_ID 標識，例如 P0）可能需要的每種資源的最大數量。例如，P0 in 1,2,3 表明進程 P0 可能分別最多需要 1 個第一種資源、2 個第二種、3 個第三種資源
2. Pid want out // mean the process request all of resources what it needs. [ex P0 want out]
   表示一個進程請求它需要的所有資源。如果系統能夠在不導致死結的情況下滿足這些請求，進程將繼續並最終退出，並釋放所有的資源。如果滿足請求會導致可能的不安全狀態，則拒絕請求。
3. Pid request “each resource” [ex P0 request 0,1,0]
   是指一個進程請求各種資源的具體數量。例如，“P0 請求 0,1,0” 意味著進程 P0 正在請求 0 個第一種資源、1 個第二種、0 個第三種。如果安全的話，系統將滿足這些資源請求；否則，拒絕請求。

Output <STDOUT>:

最後剩下的 process 以及他們所當前擁有的資源

注意事項:

- 所有 process 必須先聲明所需最大資源(先 in 才可以 request 或 went out)，否則拒絕請求
- process 預先聲明所需最大資源不可以超過系統所用有的最大資源量(範例中的 10,5,7)，否則拒絕請求
- process 不會重新宣告所需要的最大資源
- 系統必須保證分配資源後系統是否依然是 safe state，否則不會分配資源(safe state 的意思如前述影片所說，存在一個順序可以使所有 process 結束並釋放資源)
- process 當前所擁有的資源+請求的資源不可以超過一開始宣告所需的資源
- 本次作業沒有包含大數

Example:

Input:

```
10,5,7
P0 in 7,5,3
P0 request 0,1,0
P1 in 3,2,2
P2 in 9,0,2
P1 request 2,0,0
P2 request 3,0,2
P3 in 2,2,2
P4 in 4,3,3
P3 request 2,1,1
P4 request 0,0,2
P1 request 1,0,2
P0 request 0,2,0
P0 want out
P1 want out
```

計算過程:

10,5,7

P0 in 7,5,3

Process_ID	Allocation	Max	Available	Need
P0	[0, 0, 0]	[7, 5, 3]	[10, 5, 7]	[7, 5, 3]

P0 request 0,1,0

Process_ID	Allocation	Max	Available	Need
P0	[0, 1, 0]	[7, 5, 3]	[10, 4, 7]	[7, 4, 3]

P1 in 3,2,2

P2 in 9,0,2

Process_ID	Allocation	Max	Available	Need
P0	[0, 1, 0]	[7, 5, 3]	[10, 4, 7]	[7, 4, 3]
P1	[0, 0, 0]	[3, 2, 2]	[10, 4, 7]	[3, 2, 2]
P2	[0, 0, 0]	[9, 0, 2]	[10, 4, 7]	[9, 0, 2]

P1 request 2,0,0

Process_ID	Allocation	Max	Available	Need
P0	[0, 1, 0]	[7, 5, 3]	[8, 4, 7]	[7, 4, 3]
P1	[2, 0, 0]	[3, 2, 2]	[8, 4, 7]	[1, 2, 2]
P2	[0, 0, 0]	[9, 0, 2]	[8, 4, 7]	[9, 0, 2]

P2 request 3,0,2

Process_ID	Allocation	Max	Available	Need
P0	[0, 1, 0]	[7, 5, 3]	[5, 4, 5]	[7, 4, 3]
P1	[2, 0, 0]	[3, 2, 2]	[5, 4, 5]	[1, 2, 2]
P2	[3, 0, 2]	[9, 0, 2]	[5, 4, 5]	[6, 0, 0]

P3 in 2,2,2

P4 in 4,3,3

Process_ID	Allocation	Max	Available	Need
P0	[0, 1, 0]	[7, 5, 3]	[5, 4, 5]	[7, 4, 3]
P1	[2, 0, 0]	[3, 2, 2]	[5, 4, 5]	[1, 2, 2]
P2	[3, 0, 2]	[9, 0, 2]	[5, 4, 5]	[6, 0, 0]
P3	[0, 0, 0]	[2, 2, 2]	[5, 4, 5]	[2, 2, 2]
P4	[0, 0, 0]	[4, 3, 3]	[5, 4, 5]	[4, 3, 3]

P3 request 2,1,1

Process_ID	Allocation	Max	Available	Need
P0	[0, 1, 0]	[7, 5, 3]	[3, 3, 4]	[7, 4, 3]
P1	[2, 0, 0]	[3, 2, 2]	[3, 3, 4]	[1, 2, 2]
P2	[3, 0, 2]	[9, 0, 2]	[3, 3, 4]	[6, 0, 0]
P3	[2, 1, 1]	[2, 2, 2]	[3, 3, 4]	[0, 1, 1]
P4	[0, 0, 0]	[4, 3, 3]	[3, 3, 4]	[4, 3, 3]

P4 request 0,0,2

Process_ID	Allocation	Max	Available	Need
P0	[0, 1, 0]	[7, 5, 3]	[3, 3, 2]	[7, 4, 3]
P1	[2, 0, 0]	[3, 2, 2]	[3, 3, 2]	[1, 2, 2]
P2	[3, 0, 2]	[9, 0, 2]	[3, 3, 2]	[6, 0, 0]
P3	[2, 1, 1]	[2, 2, 2]	[3, 3, 2]	[0, 1, 1]

P4	[0, 0, 2]	[4, 3, 3]	[3, 3, 2]	[4, 3, 1]

P1 request 1,0,2

Process_ID	Allocation	Max	Available	Need
P0	[0, 1, 0]	[7, 5, 3]	[2, 3, 0]	[7, 4, 3]
P1	[3, 0, 2]	[3, 2, 2]	[2, 3, 0]	[0, 2, 0]
P2	[3, 0, 2]	[9, 0, 2]	[2, 3, 0]	[6, 0, 0]
P3	[2, 1, 1]	[2, 2, 2]	[2, 3, 0]	[0, 1, 1]
P4	[0, 0, 2]	[4, 3, 3]	[2, 3, 0]	[4, 3, 1]

(其中一個 Safe sequence: P1 P3 P0 P2 P4)

P0 request 0,2,0

REJECT 'P0 request 0,2,0', system will be unsafe

Process_ID	Allocation	Max	Available	Need
P0	[0, 1, 0]	[7, 5, 3]	[2, 3, 0]	[7, 4, 3]
P1	[3, 0, 2]	[3, 2, 2]	[2, 3, 0]	[0, 2, 0]
P2	[3, 0, 2]	[9, 0, 2]	[2, 3, 0]	[6, 0, 0]
P3	[2, 1, 1]	[2, 2, 2]	[2, 3, 0]	[0, 1, 1]
P4	[0, 0, 2]	[4, 3, 3]	[2, 3, 0]	[4, 3, 1]

P0 want out

REJECT 'P0 want out', system will be unsafe

Process_ID	Allocation	Max	Available	Need
P0	[0, 1, 0]	[7, 5, 3]	[2, 3, 0]	[7, 4, 3]
P1	[3, 0, 2]	[3, 2, 2]	[2, 3, 0]	[0, 2, 0]
P2	[3, 0, 2]	[9, 0, 2]	[2, 3, 0]	[6, 0, 0]
P3	[2, 1, 1]	[2, 2, 2]	[2, 3, 0]	[0, 1, 1]
P4	[0, 0, 2]	[4, 3, 3]	[2, 3, 0]	[4, 3, 1]

P1 want out

Process_ID	Allocation	Max	Available	Need
P0	[0, 1, 0]	[7, 5, 3]	[5, 3, 2]	[7, 4, 3]
P2	[3, 0, 2]	[9, 0, 2]	[5, 3, 2]	[6, 0, 0]
P3	[2, 1, 1]	[2, 2, 2]	[5, 3, 2]	[0, 1, 1]
P4	[0, 0, 2]	[4, 3, 3]	[5, 3, 2]	[4, 3, 1]

Output:

P0 0,1,0

P2 3,0,2

P3 2,1,1

P4 0,0,2

Command:

```
hw4.exe input.file
```

Provide data:

mailto: tinin@saturn.yzu.edu.tw
title:[DSHW4] SID 測資提供

作業相關問題:

mailto: s1091515@mail.yzu.edu.tw
title:[DSHW4] SID 作業問題