

網路程式設計實務專題題目

多路鏈結平行傳輸 – easy FCFS + backfilling 排班

國立高雄科技大學資訊工程學系 王志強老師

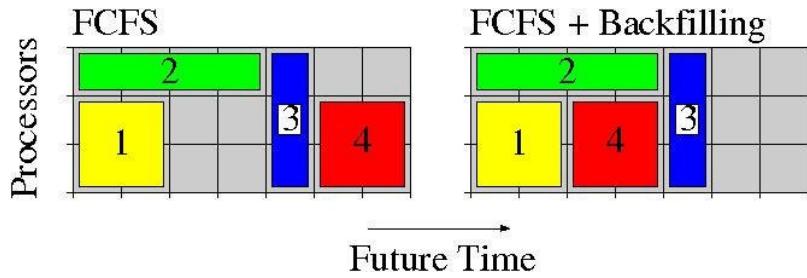
1. 導言

這個作業是讓學生撰寫一個 C++ 程式：設計一個無線網路系統的多路鏈結平行傳輸的排班模組。生活中常見的 Wi-Fi (IEEE 802.11 標準) 提供多個無線頻段供通訊使用，並將這些頻段劃分為較小的頻道。舉例來說，Wi-Fi 的 2.4GHz 頻段的頻率範圍為 2.4-2.4835 GHz，共約 83.5 MHz，並被劃分為 14 個頻道（頻道編號 1-14）。為了讓 Wi-Fi 網路傳輸更有效率，Wi-Fi 7 開發了很多創新的新功能，其中最關鍵的為：多路鏈結運作 (Multi-Link Operation, MLO)。

早期市面上所使用的 Wi-Fi 技術可以允許一個 Wi-Fi 裝置利用 2.4GHz、5GHz 或 6GHz 的頻段來傳輸資料，但是在同一個時間內只能使用一種頻段，而切換不同的頻段需要耗費一定的切換時間，因此對網路系統造成延遲。對於延遲相對敏感的應用如無線 VR 裝置、即時的多人高解析度線上連線遊戲、元宇宙等，如何降低延遲時間即為一個急須解決的課題，而 Wi-Fi 7 的 MLO 技術是可以降低延遲時間的解決方案。

本次專題之目的是：

- (1) 讓學生練習如何使用 2d 陣列來記錄多種資源(頻道)的配置狀況：以 rows 來表示各個資源、以 columns 來表示該不同時段，所以陣列中的元素 (row A, column T) 便記錄著資源 A 在時段 T 的配置狀況。
- (2) 當某一筆資料傳輸的工作來到時，學生寫的排班模組便要依據前述 2d 陣列排班表，替這筆資料傳輸來排班。這筆資料傳輸工作會記錄著它的 id 編號、抵達時間、它的傳輸需要使用多少個資源、以及它的傳輸時間需要多少時段。
- (3) 為了替資料傳輸配置資源，排班模組會採用 easy FCFS + backfilling 排班法。FCFS 就是先到的資料傳輸排前面，後來的資料傳輸排後面，可是加上了 backfilling 就有例外：在不延遲已經排好的資料傳輸的情況下，後到的資料傳輸是可以先做的。下圖的範例就說明了 backfilling 可以把 4 號排在 3 號前面。



2. 模組的設計思維

這個軟體需要(1)一個類別做成的排班表來記錄已經到達的資料傳輸工作的排班現況，以及(2)一個類別依照排班表現況，替目前抵達的工作來執行(esay FCFS + backfilling)排班，其設計大綱如下：

(1) Class Schedule

[1] 主要用途

提供建構函數來建立和初始化 2d 陣列，提供函數來設定、檢查 2d 陣列 內存的數值。

[2] 實作方法

(a) 資料成員：

甲、一或多個 int 類型指標的指標，用來動態配置 2d 陣列。

乙、陣列的 row 數量和 column 數量。

(b) 成員函數：

甲、一個建構函數，用來初始化資料成員。

乙、一或多個 set 函數，用來設定陣列內存的數值。

丙、一或多個 get 函數，用來讀取陣列內存的數值。

丁、一或多個 test 函數，用來檢查陣列內的元素是否可以被預約排班。

(2) Class MLO

[1] 主要用途

提供 FCFS 和(esay FCFS + backfilling)兩種方法來排班 Schedule 物件。

[2] 實作方法

(a) 資料成員：

甲、一個供 FCFS 使用的 Schedule 物件。

乙、一個供(esay FCFS + backfilling) (BF)使用的 Schedule 物件。

(b) 成員函數：

- 甲、一個建構函數，用來初始化資料成員。
- 乙、一個 FCFS 排班函數，其參數來自剛到的資料。
- 丙、一個 BF 排班函數，其參數來自剛到的資料。

(3) Main function 的運作

[1] 資料傳輸工作之產生

老師會提供一個可以不斷產生新資料的程式碼。該程式碼被呼叫之後，便會產生出一個 struct 變數，其 struct 定義如下：

```
struct DX_INFO {  
    long long id;  
    double arrivalTime;  
    int NoOfResourcesNeeded;  
    int NoOfTimeSlotsNeeded;  
};
```

[2] Main function 的任務

學生寫的 main function 必須執行 while 迴圈不斷地呼叫(3)-[1]的函數來產生新的資料傳輸工作，然後呼叫 MLO 去執行 FCFS 和 BF 排班。工作可以排班的最早時間是計算方式是將它抵達時間(arrivalTime)無條件進入。舉例來說，某工作的抵達時間是 12.345，那麼它的最早排班時間是 13。

[3] 結果展示

學生寫的 main function 內會設定一個 While 迴圈執行次數的上限值。當 While 迴圈跑完之後，程式必須計算 FCFS 和 BF 排班的 throughput 並且把結果列印出來； $\text{throughput} = (\text{已傳輸的總數量}) / (\text{最後一筆資料傳輸的時段編號} + 1)$ ，假設時段是從 0 開始編號(此乃按照 C++ 陣列語法)。

為了證明程式的正確性，請學生自行設計一個測試用的案例，然後用這個案例來跑你的程式。學生必須把 Schedule 排班的內容列印出來，然後根據 Schedule 排班內容來做口頭報告，其目的是說服評審相信學生寫的程式正確無誤。