

# CS305 作業系統概論 Prog. #2 Proc. Generation & Communication

2015/04/20

## 一、作業目的

這是 **Prog. 1 補交版**。鼓勵同學熟悉如何利用fork()系統呼叫產生新的process，以及process彼此之間如何互相傳遞資料。但本作業成績將以8折計算，且不接受任何理由補交。**Prog. 1 與 prog. 2 會選高分計算，兩者不會重複計分。**

## 二、作業內容

【探索新能源】面對世界暖化越來越嚴重的情況，科學家們無不四處尋找最環保的能源。經過一番搜尋噶瑪蘭宇宙能源總局終於在大約北緯23度28分東經120度57分左右的地區發現紫氦石（克利普頓石）礦源，能夠在激化能源之後，轉成普通的花崗石。然而在紫氦石激化能源之前，由於粒子輻射會嚴重影響人類腦細胞的神經元，使人易怒邪惡，因此必須派出機器人進行探勘與採礦作業。為了能夠迅速探勘到紫氦石礦源，探勘作業要平行化進行。噶瑪蘭宇宙總局長特別尋找海豚學院的志達人帶著他的高徒開發平行探勘程式。由於土質因素，有些土層不能挖掘，因此機器人在尋找礦源時，不能破壞這些土層。

為了在系統上線前，先行驗證程式正確，志達人要你設計一個程式，利用fork()的方式，產生新的process來模擬機器人探勘的過程。程式會先讀入一個地層圖，當進行探勘時，每遇到一個岔路就 fork 數個 children processes前往探索，parent process 則停留原地等候回報，等收到所有回報，parent process再往上一層parent process 回報，直到所有路徑都已經探索完畢。在回報的時候，需要印出一些訊息，規則如下（**請注意，這個流程就是助教要評分的主要項目，因此務必清楚顯示出尋找的流程**）：

1. 最開始的機器人要印出自己的 PID 與起始座標。
2. 分派的時候，每一個parent process 要印出children processes的PID，以及產生children process 的座標。
3. 如果process走到某條路的盡頭仍然找不到礦源，就立刻回報盡頭的座標給等候的parent process 同時直接結束。此時由parent process 送出 PID+座標+ “None!” 的訊息，此失敗的訊息就不再往上传送。Parent process 收到所有回報後，如果沒有發現礦源，也要回傳自己的PID+座標+ “None!” 到上一層。
4. 如果process找到礦源，也立即回報座標給等候的parent process 同時直接結束，此時由parent process 送出 PID+座標+ “Found!” 的訊息。並繼續回報自己的座標給上一層的parent process。上一層的parent process繼續印出PID+座標+ “Found!” 的訊息，一直到最上層為止。
5. 如果該地層圖有礦源，最上層的parent process最後就印出自己的 PID+當時座標+ “Found!” 訊息。如果沒有，則印出自己的 PID+當時座標+ “None!”。最後並印出整個程式所花費的CPU時間。

例如，最基本的地層圖將以矩陣顯示在一個檔案中。S表示起點，K表示礦源，左上角為（0,0）：

```
*****
*****
S      **  ** *  *****
***** **  ** *  ***** K
***** **  ** *  *****
***** **  ** *  *****
***** *****
**                ***
** *****
*****
```

假設第一個機器人的 PID為1200，因此程式執行的過程可能為：

```
> prog1 data.txt
```

```
[pid=1200]: (2,0)
```

```
[pid=1201]: (7,5)
```

```
[pid=1202]: (7,5)
```

```
1201 (8,2) None!
```

```
...
```

```
1203 (3,19) Found!
```

```
...
```

1202 (8,13) Found!

1200 (7,5) Found! Total: 1250 ms.

### 三、作業要點

1. **請注意，4.20日以後，不再接受Prog. 1 的上傳，請勿上傳到錯誤區域。**
2. 請注意，本作業系統使用的程式語言是C/C++，測試平台的作業系統： Ubuntu 14.04 LTS。使用的編譯程式為g++ 4.8.2。其他平台或程式語言不在本次作業考慮範圍之內。如無法在測試平台上編譯與執行，都不予給分。
3. 請注意，本作業一定要用fork()來進行。任何不用fork()所產生的行程來完成的程式，都不予給分。
4. 地層圖資料內容為：‘\*’代表不可挖掘的土層，‘S’代表機器人起點，‘K’表示礦源。S與K有可能在矩陣中任何一點。其他情況，將在不同分難易程度中考慮。
5. 本作業的評分方式如下：**（總成績將以8折計算）**
  - a. **【基礎功能】**在命令列讀入一個資料檔，並能順利產生一個child process者，可得20分。即看見 [pid=1200]: (0,2) 與 [pid=1201]: (7,5)。
  - b. **【基本功能】**假設地圖中**最多**只會出現一個礦源，地圖中沒有迴圈，且假設地圖外都是不可開挖的土層。地圖矩陣大小為10x20或20x20。左上角為 (0,0)。所有路徑的寬度都為 1。
    - i. 產生子行程（取下列兩項中的高者）
      1. 正確按照叉路數目，循序一次一條岔路地產生子行程來完成探索，可得10分。
      2. 正確按照叉路數目，平行產生子行程來完成探索，可得30分。
    - ii. 最上層的process正確取得自己的pid，可得5分。
    - iii. 子行程正確探索新的路徑，而不重複已走過的路徑，可得5分。
    - iv. 可正確處理走出地圖範圍外的情況，可得5分。
    - v. 最後結果可以正確判斷有無礦源者，可得5分。
    - vi. 傳遞資料給主行程的得分（取下列兩項中的高者）
      1. 能正確將child process的搜尋結果通知給主行程，且不使用檔案者，可得20分。
      2. 能正確將child process的搜尋結果通知給主行程，但需要使用檔案者，可得5分。
    - vii. 正確印出整個程式所用到的CPU時間，以 millisecond 為單位，可得10分。
    - viii. 報告的得分：將依照各位說明報告的優劣，給予0~10分。
  - c. **【進階功能】**完成以上基本功能者，才可按照以下項目，多得其他的分數。但請注意，**如何demo出你程式中的這些進階功能，必須在你的說明文件檔案中詳細說明，如果助教看不懂，可能反而會扣分。**
    - i. 可處理有2-5個礦源的地圖，找出所有礦源。parent process 要多印出礦源個數。（10分）
    - ii. 可處理有迴圈的地圖。但child只看有沒有遇到parent走過的路，如果有，就立刻回報None訊息並中止。（10分）
    - iii. 可處理迴圈。child連遇到其他child走過的路，也立刻回報None訊息並中止。（10分）
    - iv. 針對有多個路徑到某礦源的情況，可找出其中最短路徑，並印出所有最短路徑。（20分）
6. 本作業需繳交檔案：
  - a. 說明報告：檔案為doc或pdf格式。
    - i. 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**。
    - ii. 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。

- iii. 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，會以portal的資訊寄信（**最多兩次**）通知你來說明，但每說明一次，**助教會少給你10分**。
- a. 完整原始程式碼。**不可含執行檔。助教會重新編譯你們的程式。**
- 7. 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。**
- 8. 助教會用多個地圖來進行不同功能測試。

#### 四、繳交方式：

- 1. 最終繳交時間：
  - a. 電子檔在 2015.05.02以前，上傳至個人portal。如有多個檔案，將所有檔案壓縮成zip（rar,7z 亦可）格式，然後上傳。
  - b. 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. **本作業不接受任何理由補交。本作業評分將以8折計算。Prog. 1 與 prog. 2 會選高分計算，兩者不會重複計分。**
- 4. 老師不接受「門縫」方式繳交，助教也不接受任何portal以外任何方式所繳交之作業。

#### 五、如有未盡事宜，將在個人portal板面公告通知。

#### 六、If you need **an English version** of this assignment or **any assistance in English**, please contact Prof. Yang.

#### 七、參考資料

- 1. 參考課本圖3.17與3.18。
- 2. 參考課本programming problem 3.14, 3.15。
- 3. time 指令，會傳回三個時間
  - a. 實際時間(real time): 從command命令列開始執行到運行終止的消逝時間；
  - b. 用戶CPU時間(user CPU time): 命令執行完成花費的用戶CPU時間，即命令在用戶狀態中執行時間總和；
  - c. 系統CPU時間(system CPU time): 命令執行完成花費的系統CPU時間，即命令在核心狀態中執行時間總和。
- 4. CPU 時間 = user CPU time + system CPU time