CS305 作業系統概論

Prog. #1 Proc. Generation & Communication 說明報告

汪文豪(學號：1071710)

➢ 如何編譯與測試程式：

1. 編譯：gcc 檔名.cpp –o 檔名 –lrt
2. 執行：./檔名
3. 輸入整數 n：1-100，若非此範圍必須重新輸入整數
4. 輸出結果

➢ 設計理念：

本次使用到的特殊函式庫

1. unistd.h
   - C/C++提供其 POSIX 的 API 功能函式庫
   - 可使用 fork、getpid、getppid 等這次作業需要用到的功能
2. fcntl.h
   - for O_constants
   - 使用 shm_open()時，對於其文件的處理方式
3. sys/mman.h
   - 可使用 shm_open()、mmap()等 API，幫助建立共享記憶體位置
4. sys/wait.h
   - parent process 需要使用到 wait(NULL)，因此引入此標頭檔

本次程式說明：

- 2 個 function：
  1. int collatz(int n)

     負責計算作業題目之 Collatz conjecture，並將其計算結果做回傳，

倘若傳入整數為 1 則是其終止式，直接回傳 1

2. void error_and_die(const char *msg)

倘若分配空間產生錯誤，將其錯誤印出並退出程式

- 程式流程：

一、宣告整數 n 與 pid 以利後續使用

二、輸入 1 – 100 整數給 n，倘若超出範圍，重新輸入直到在 1 – 100 範圍內

三、開啟共享記憶體空間，並開啟讀寫權限。本次開啟五個整數空間，空間分配用途如下：

1. ptr[0]:第一個位置紀錄 pid，以利辨識目前是需要 parent process 做行動還是 child process 做行動
2. ptr[1]:紀錄 collatz 計算結果
3. ptr[2]:紀錄最大值
4. ptr[3]:紀錄最大值位置
5. ptr[4]:紀錄其計算 collatz 結果的次數

四、使用 fork，將 parent process 複製一份並建立 child process

- fork 執行後會傳回整數值，依此整數值寫下判斷式，並在其 parent、child process 寫下各自行程需做的事情
  - 0 代表目前在新的行程裡，也就是程式正在執行 child process
  - 大於 0 則代表目前在原本的 parent process 中，其紀錄的 pid 即為 child process 的 pid
  - 小於 0 就代表創建 child process 失敗，print error 並退出程式

五、建立其不同 process 所需做的事情

- Parent process

- 作業指定由 Parent process 先攻，因此在 parent process 進入迴圈前先做記憶體初始化，將 ptr[0]和 ptr[1]分別傳入其 child process id (即先前大於 0 的 pid)以及輸入的 n，其餘初始化成 0
- 進入 while(true)迴圈
  - 每次讀共享記憶體內第一個位置，倘若 ptr[0] == getpid()，意即第一個位置的數字轉換為 parent process 的 pid，即開始做 parent process 所要處理的事情
  - 計算 collatz 結果並將結果放置 ptr[1]
  - 將 ptr[4] +1，即紀錄其 collatz 計算次數
  - print 出結果
  - 倘若目前計算結果大於 ptr[2] (紀錄 max 位置)即將計算結果放入 ptr[2]並將目前 ptr[4]計算的次數放入 ptr[3]內
  - ptr[0]改成 pid ，將主控權還給 child process
  - 終止條件為最後算出的 collatz 為 1，為了告知 child process，parent process 已經結束，所以 ptr[1] = 0 並終止迴圈
  - 迴圈出來後執行 wait(NULL)，等待子行程結束，避免 zombie 發生

➢ Child process

- 進入 while(true)迴圈
  - 因最後必須由 parent process 做結束計算過程，因此迴圈必須一進入就先確認其 ptr[0]是否為 0，是就代表 parent process 已經結束，等待子行程的回應，所以終止迴圈

- 每次讀共享記憶體內第一個位置，倘若 ptr[0] == getpid()，意即第一個位置的數字轉換為 child process 的 pid，即開始做 child process 所要處理的事情
- 計算 collatz 結果並將結果放置 ptr[1]
- 將 ptr[4] +1，即紀錄其 collatz 計算次數
- print 出結果
- 倘若目前計算結果大於 ptr[2] (紀錄 max 位置)即將計算結果放入 ptr[2]並將目前 ptr[4]計算的次數放入 ptr[3]內
- ptr[0] = getppid() ，將第一個位置換成 parent process 的 pid，主控權還給 parent process
- 退出迴圈後，exit(0)，child process 任務結束

六、print 出最大值與是第幾次算出的結果

七、結束程式