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：**

**本次使用到的函式庫**

1. stdio.h

* 標準函式庫

1. stdlib.h
   * 標準函式庫
2. 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

1. fcntl.h
   * for O\_constants
   * 使用shm\_open()時，對於其文件的處理方式
2. sys/mman.h
   * 可使用shm\_open()、mmap()等API，幫助建立共享記憶體位置
3. 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