

一、問題一(30%)

1. 問題描述

請寫一隻程式驗證 `fork()` 的機制。驗證的內容包含 (1) 父子程序的 `fork()` 的返回值以及 (2) 如何控制父子程序的執行順序。

(P.S. 只有 Unix-based 的作業系統才有 `fork()` 等函式，所以請直接或虛擬機安裝相關的作業系統，如 Ubuntu, Fedora 等等。請不要在 Windows 上做，會做不出來。)

2. 給分標準

(1) 10%，輸出父子程序各自的返回值。

(2) 20%，請找到一個方法去控制父子程序的執行順序，並且寫出一個會利用這個控制方法的功能。(P.S 只要有用到就會給分數，不需要寫的很複雜)

二、問題二(70%)

1. 問題描述

請寫出一個 shell。以下是 pseudocode，請參考。

```
#define TRUE 1

while (TRUE) {                                /* repeat forever */
    type_prompt( );                            /* display prompt on the screen */
    read_command(command, parameters);        /* read input from terminal */

    if (fork() != 0) {                         /* fork off child process */
        /* Parent code. */
        waitpid(-1, &status, 0);             /* wait for child to exit */
    } else {
        /* Child code. */
        execve(command, parameters, 0);      /* execute command */
    }
}
```

根據以上 pseudocode 可以寫出最基本的 shell。

另外還有三個功能分別是 Pipe (|), Redirection to File (>), Redirection from File (<)。

2. 給分標準

(1) 30%，寫出基本 shell，能夠根據你的程式呼叫並執行各種 Linux 的指令，包含各項指令的參數。

(2) 10%，以你的 shell 為基礎寫出 pipe 的功能。如果你的程式只能跑一個 pipe 只會得到 5%，必須要能夠 pipe 無數個指令才能得到完整的分數。

如以下範例：

I. `ls | head -3`，5%

II. `ls | head -3 | tail -1 | <other command> | ...`，10%

(3) 10 %，以你的 shell 為基礎寫出 redirection to file 的功能。同 (2)，必須要能夠連續重導入才有完整的分數。

(4) 10 %，以你的 shell 為基礎寫出 redirection from file 的功能。同 (2)，必須要能夠連續重導出才有完整的分數。

(5) 10%，能夠組合 (2)、(3)、(4) 的功能。如範例：`ls -al | head -3 > output`

三、其他資訊

1. 不限程式語言，可以用 C、C++、Java、Python 等等。
2. 需要一份報告，報告內容必須要有 (1) 開發環境、(2) 程式邏輯 (可用流程圖表示)、(3) 函式功能說明，如有其他想寫可以在後面補充。注意，如果沒有寫出 (1)、(2)、(3) 的內容會斟酌扣分。
3. 請在期限內 new ee-class 上傳一個壓縮檔，包含程式碼和報告。檔案名稱格式為「Program1_學號_姓名」，如果沒有按照格式命名也會斟酌扣分。
4. 繳交期限是 4/7 (一)。
5. Demo 時間是 4/13 (二) 至 4/16 (五)，會開表單給各位填時間。