

---

# Hw2 : Shell it! (level two)

## A homework of making your own shell

Due date: 5/7 (SUN) 23:59 am (好孩子請勿遲交、抄襲，本作業不接受補交)

目的：

在這個作業中，我們將繼續要求你完成 shell，功能上則以執行一個新程式以及管線處理為主。本次你將於作業中專注於使用 fork / exec / pipe 等相關家族指令來完成工作。

\*此作業僅可使用 C/C++、執行環境為 ubuntu / mac OS X (請註明)。

在這個作業中，你必須接續你上次的成果，並根據上次切割出的許多指令，依據需要把他們進行輸入與輸出的串接。我們預期你將會需要使用 fork(), execvp(), wait(), waitpid(), pipe(), open(), close(), dup\*()... 等來完成工作(可以使用之函式不限於以上)。

你可以參考一下文件以獲取更多資訊：

<http://linux.die.net/man/>

<http://man7.org/linux/man-pages/index.html>

基本上我們也允許你使用任意的 open source 函示庫(比方說 GNU Readline Library)，但**如果直接運用其他 shell 來完成你的作業一律零分**

\*在 Level2 中，本 Shell 應具備 Level1 的基礎，並將 Level1 的第 2,3 兩項改寫成程式的執行。

1. 可以接收並執行任意不帶參數的指令。舉例：

icheyneh@/home/OS>ls << 使用者輸入 ls 並按下 Enter

接著你便透過你的 shell 來建立一個新的 ls 程序，並將輸出導向標準輸出（於文字介面上輸出）

2. 可以接收並任何帶有參數的指令。舉例：

icheyneh@/home/OS>ls -al << 使用者輸入 ls 並加上附加參數 "-al" 最後按下 Enter

接著你便透過你的 shell 來建立一個新的 ls 程序，執行時給與-al 這個參數，並將最終輸出導向標準輸出

3. 可以將多個不同的指令串接起來，提供 pipe '|' 的功能，讓使用者可以將一個程式的輸出 (standard output) 接到另一個程式的輸入，需要支援多個程式串接（最多四個）。

---

舉例：`ls -al | more` ( 將 `ls` 的輸出導向程式 `more` 的輸入 )

舉例：`ps -ef | grep svn | more`

(簡易 pipe 實作) <http://hwchiu.logdown.com/posts/1733-c-pipe>

4. 提供 redirect '`<`, '`>`, '`1>`, '`2>` ...' 的功能讓程式將標準輸出、標準錯誤輸出等重新導向到新目的地 ( 作業中只測試從檔案導入或導出到檔案，但會跟 pipe 混合運用 )

舉例：`ls -al > ~/logfile.t` (將輸出導向一個檔案)

以上行為可參考這邊的說明：

(更多 pipe/redirect 使用說明) <http://ryanstutorials.net/linuxtutorial/piping.php>

5. ( 加分項目 ) 提供使用者可以用方向按鍵 ( 上下 ) 來瀏覽之前輸入過的指令，以及用 Tab 來進行檔名自動完成等功能...

提示：推薦使用 `readline` 這個函式庫來達成

。

#### \*作業繳交：

1. 請撰寫一份報告 ( 10% )，詳細描述上面的每一個功能如何完成，以及你在完成功能時的發現。同時包含詳細的功能測試截圖。( 報告中請務必提及 **如何編譯你的程式、是否需要安裝額外套件** )

命名範例：`OShw1_s123456789.pdf`

2. 請繳交你 **可編譯的程式碼**，不需要附上任何編譯好的檔案。程式碼請用以下形式命名：

命名範例：`OShw1_s123456789.cpp` (結尾可是 `.c` 或 `.cpp`)

若你有多個檔案，請壓縮成 `zip` 檔案並記得在報告中附上編譯方法說明