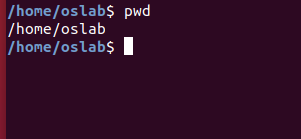
4108056018 資工三 王旻玄

**基本功能:**

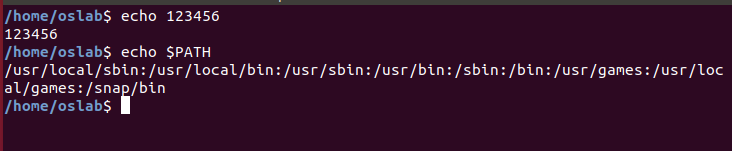
1. **印出prompt，例如“目前目錄$ ”(可以自行定義)，接受command**



1. **接受command並執行command**

**Build-in commands:**

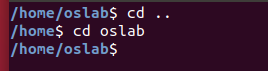
echo:



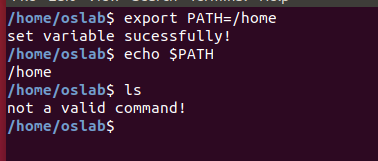
pwd:



cd:

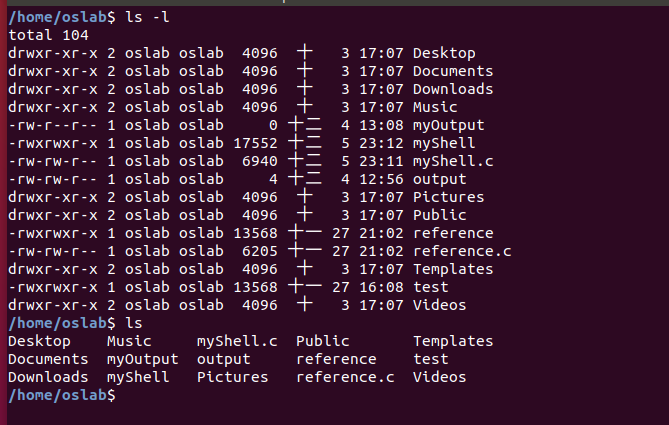


export:



設定PATH後ls變得不能使用

**外部commands：**

****

以ls 與 ls -l 來示範

1. **command執行結束後，再回到1.**

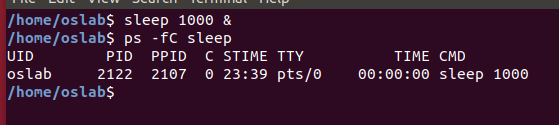
以上面的圖可看出，指令結束之後都會回到1.

**進階功能:**

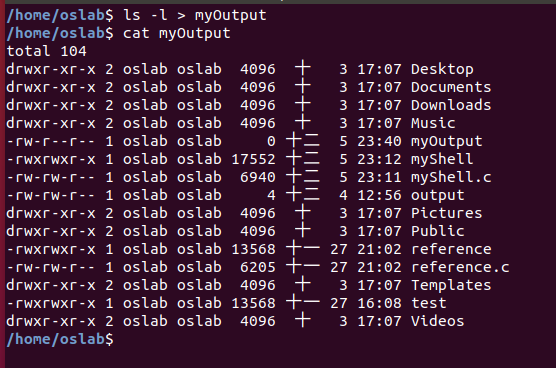
**查詢歷史command:**

由影片展示: <https://www.youtube.com/watch?v=UIdXDHwo82M>

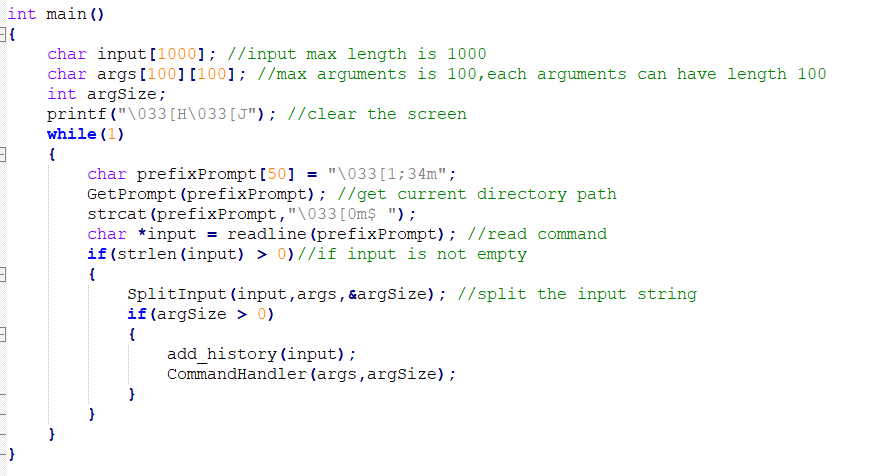
**支援background execution (&):**

****

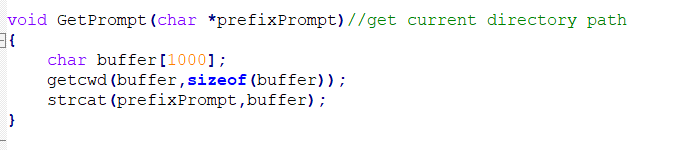
**支援Output redirection (>):**

****

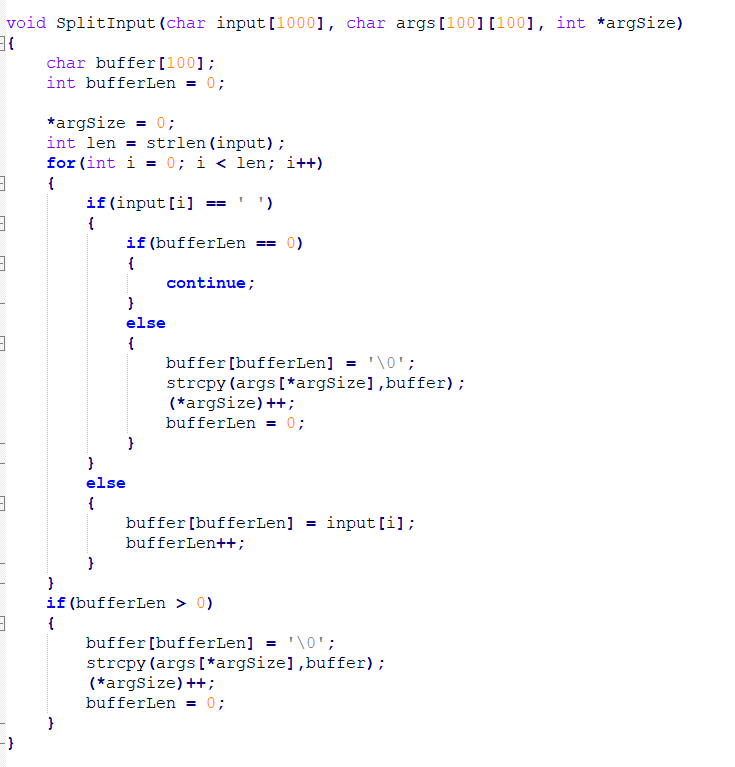
**程式碼說明:**

****

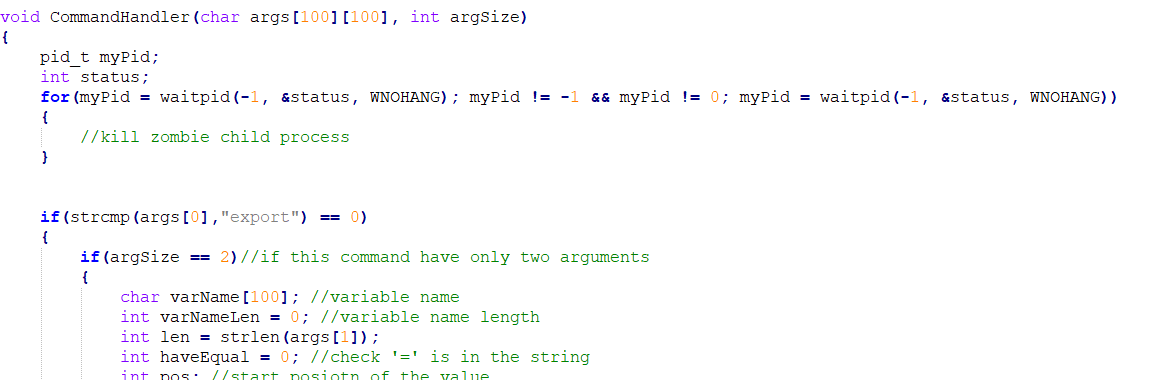
進入程式後會先把螢幕clear，每次都會呼叫GetPrompt取得現在的路徑並輸出，之後會在呼叫readline讀進terminal的command，如果讀進來的command不是空的，那麼會呼叫SplitInput切割字串，如果arguments的數量大於0就會把這個command加進history並執行。



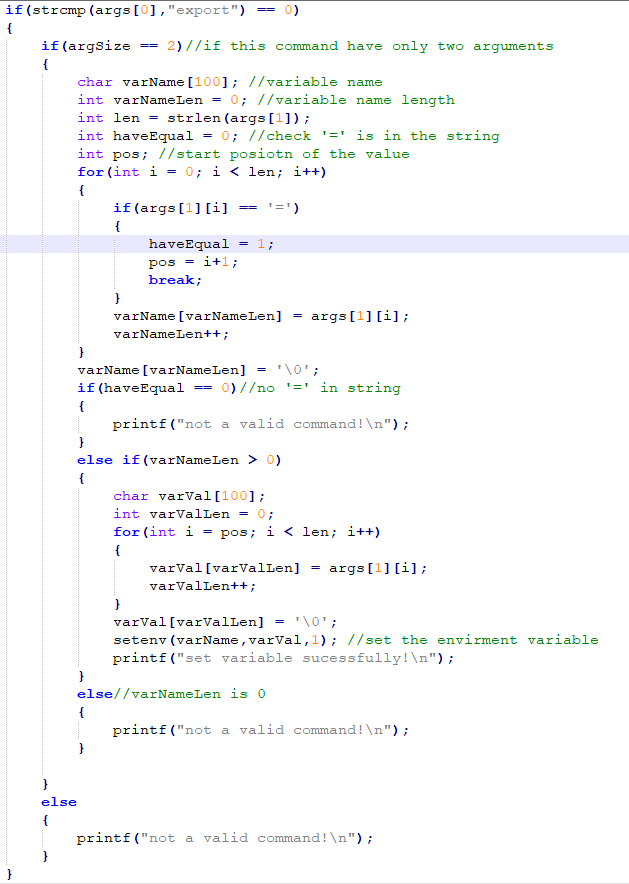
GetPrompt會取得現在所在的路徑，並串到我要用的字串上面。



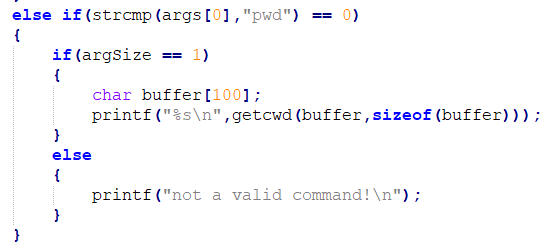
SplitInput會把command依據空格切成數個arguments，最後會取得總共有幾個arguments。buffer[]會儲存目前建構的argument，當遇到空白或是字串結尾時，就會加進args[][]並開始重新建構一個新的argument，bufferLen用來記錄目前argument的長度，加進args[][]之前會在最後加上一個結束字元\0。



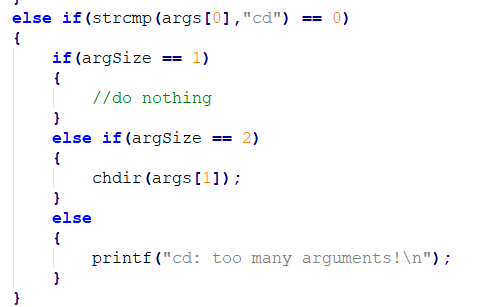
最後是commandHandler，一開始會先去接收zombie process，避免之後會產生問題，之後會根據第一個argument去處理指令。



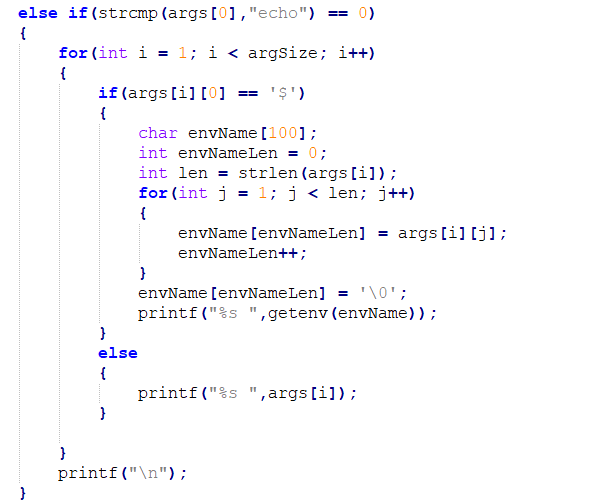
如果第一個argument是export，一開始會先檢查arguments count是否為2，是的話才能執行，之後會根據’=’來取得variable name跟value，如果沒有’=’或是’=’左邊沒有字串則不是正常的指令，如果variable name跟value都存在則透過serenv來設定環境變數。



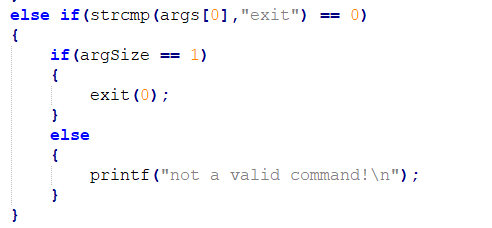
如果第一個argument是pwd，且後面沒有其他東西，則透過getcwd()取得現在的位置。



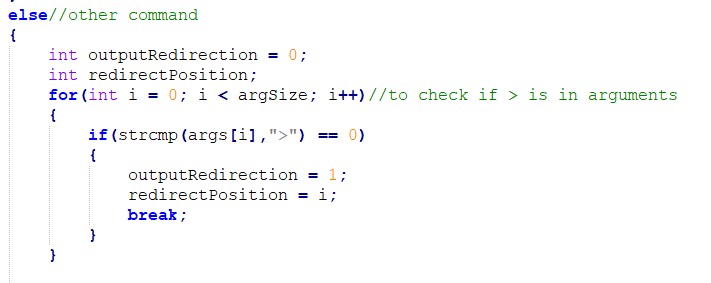
如果第一個argument是cd，若後面沒有其他東西，則什麼都不做。若剛好有2個arguments，則執行chdir()到指定的路徑，若超過2個arguments則輸出錯誤訊息。



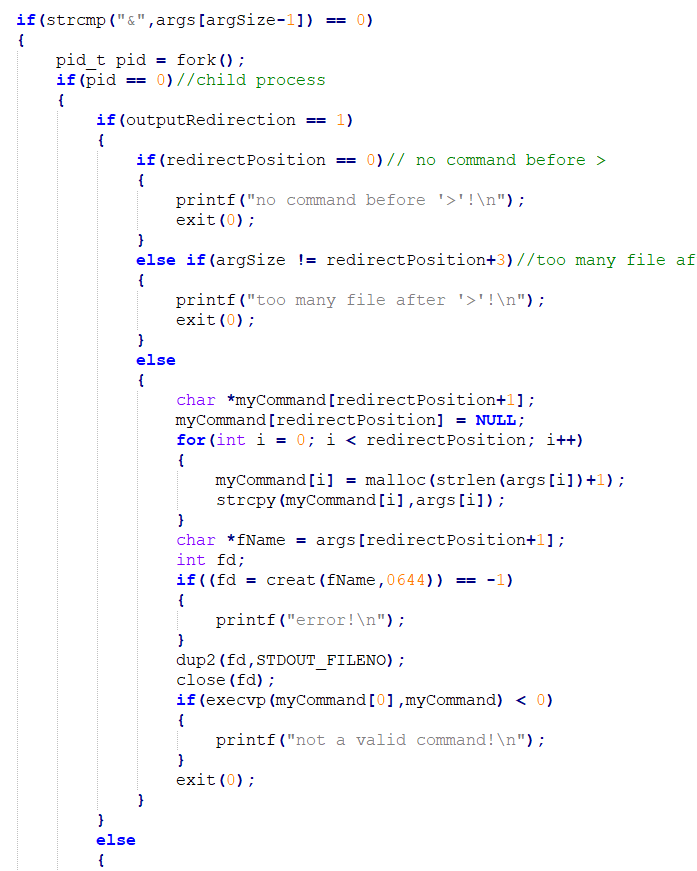
如果第一個argument是echo，基本上會直接print後面的arguments，但如果argument的開頭是$，則要透過getenv()來取得環境變數的值。

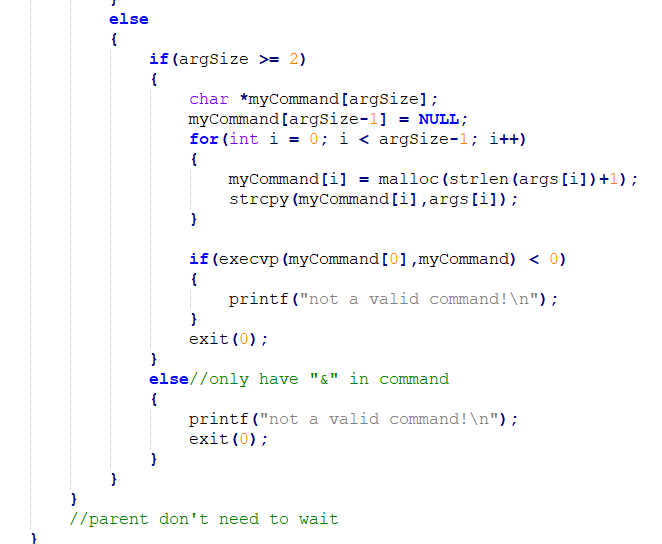


如果第一個argument是exit，且只有一個argument，則結束程式。

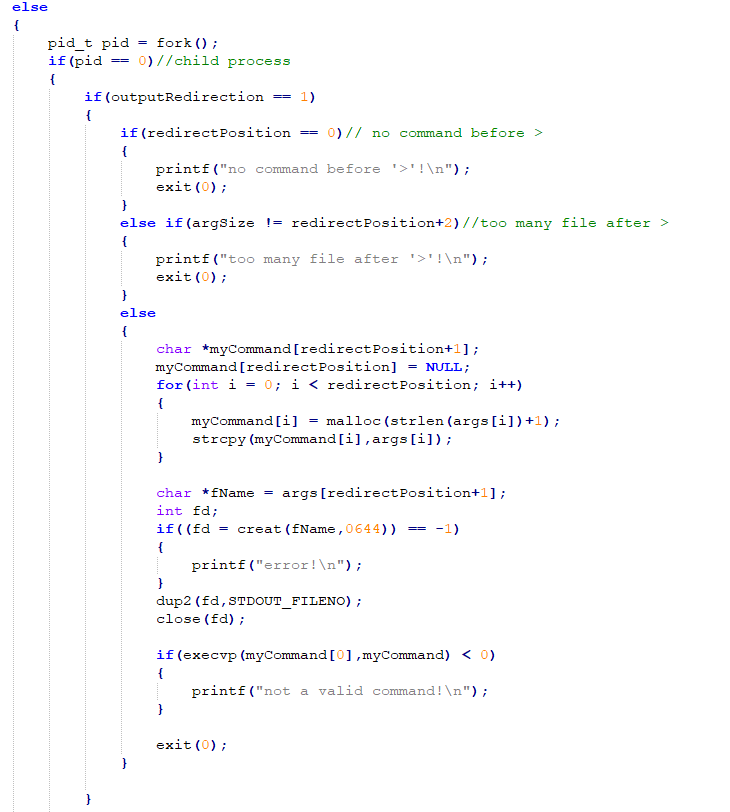


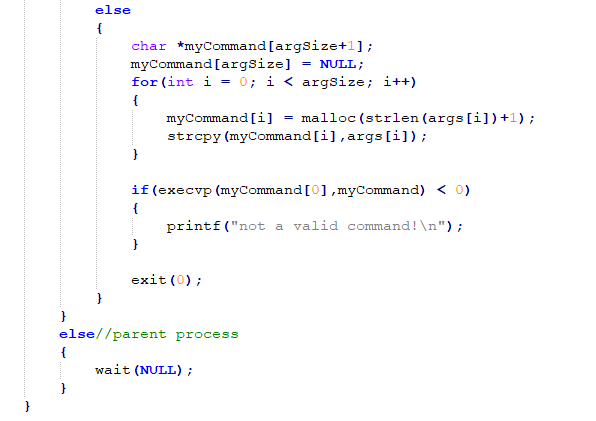
若是其他的command，則當作外部指令來執行，一開始會先檢查arguments中有沒有’>’。





如果最後一個argument是’&’，代表是background execution，之後parent不會等待這個process，一開始會先檢查是否為output redirection，是的話會檢查>的左右兩邊是不是正常的格式，之後會設定輸出的file，如果不是的話則跳過上述的步驟，之後會進行execvp()來執行外部指令。





else則是非background execution的部分，最後parent process會等待child process執行完，一樣一開始會先檢查是否為output redirection，是的話會檢查>的左右兩邊是不是正常的格式，之後會設定輸出的file，如果不是的話則跳過上述的步驟，之後會進行execvp()來執行外部指令，上述的程式由child process來執行，parent process則要wait() child process執行完畢。