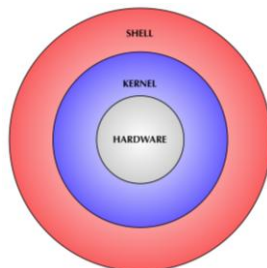


「作業系統」作業三 (Due 2025/4/2(三)13:00)

請於截止時間前把完成的程式碼或把整個專案(但**移除所有執行檔**)壓縮成單一檔案，主檔名設為 OS3 再透過 <https://reurl.cc/M37Mzp> 上傳繳交，上傳頁面的「學號欄」務必填寫以利識別繳交資料的同學，謝謝。

Command interpreter 是使用者呼叫作業系統服務或其他程式的常用重要介面。UNIX 或 Linux 之 command interpreter 稱為 *shell* (外殼)，在 Windows 則稱為「命令提示字元」。使用者可透過 shell 下達文字命令，shell 把收到的命令轉換為計算機可以執行的指令或呼叫某些程式進行後續處理，再

將結果回傳給使用者。Shell 為使用者提供與作業系統互動的介面，它像是系統和使用者之間的橋樑，包裹在作業系統之外，猶如甲殼，因此得名。

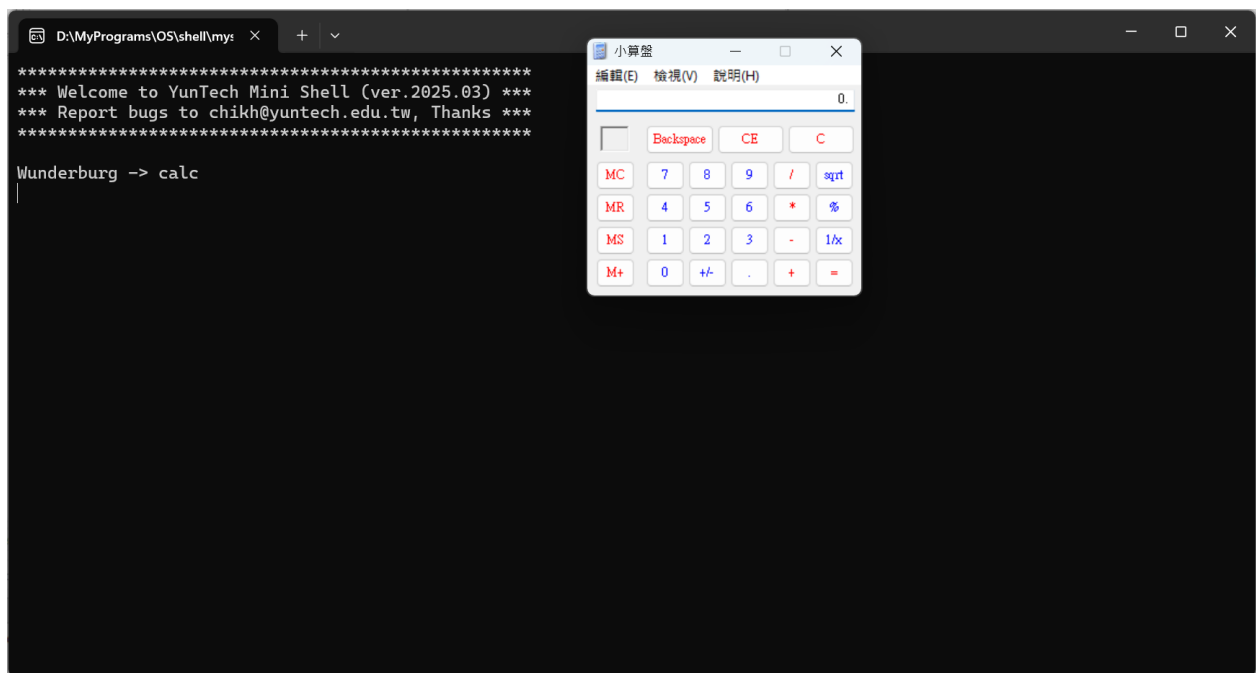


請在 Windows 環境中以 C 撰寫一個 shell 程式，該程式可以執行 **calc**、**notepad**、**hostname**、**help**、**time**、**product**、**bye** 等指令。其中，**calc** 為「小算盤」程式，**notepad** 為「記事本」，**hostname** 列出使用者目前所用的機器名稱，**help** 列出命令列指令的使用說明，**time** 顯示目前機器時間，**product** 計算給定若干組數字之乘積，**bye** 則結束 shell 程式。shell 預設以前景(background)模式執行程式，但若在命令列多鍵入「&」符號則為背景(background)執行模式。前景模式係指：命令列啟動的新程序將主導接下來的流程，新啟動的程序運作期間將阻塞 shell 的動作，直到該程序結束，shell 才恢復運行。例如：在命令列鍵入 **calc** <Enter>，shell 將啟動「小算盤」程式，從此 shell 如同凍結而 **calc** 開始執行；迨「小算盤」關閉，shell 才恢復響應。若在命令列鍵入 **calc** & <Enter>，則 shell 以背景模式啟動「小算盤」程式，使得 shell 與 **calc** 同時獨立運作，shell 不會因叫出 **calc** 而被阻塞。

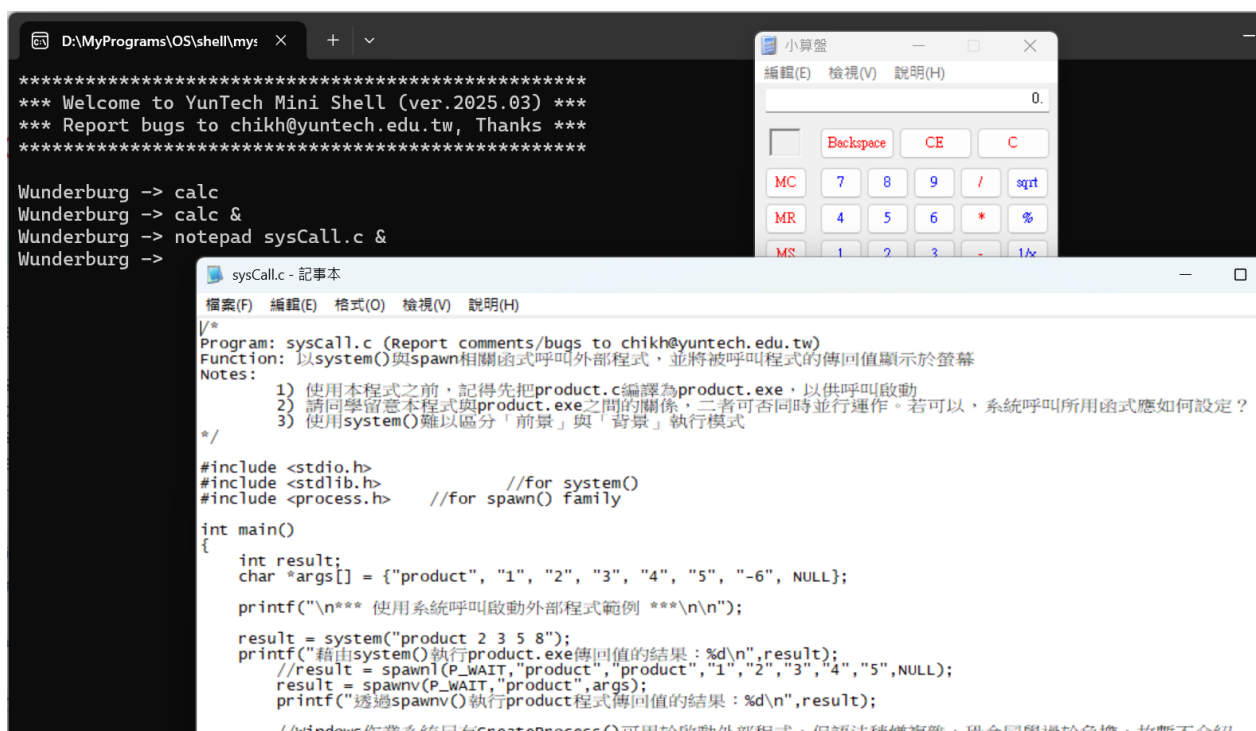
這裡所用的 **calc.exe**、**notepad.exe**、**hostname.exe**、**help.exe** 係 Windows 作業系統一併安裝的工具程式，位於 c:\windows\system32\目錄夾，為便利同學觀察所開發的 shell 與這些程式互動模式(支援背景執行)之差異，此次作業所用的這幾份程式以舊版為主，可[按此](#)下載取得。下載之壓縮檔內另有 **product.c** 與 **sysCall.c** 二份程式，由紀老師設計給同學練習所用，**sysCall.c** 內有線索可幫助同學完成此次作業。

本次作業尚用不到「多執行緒」程式技巧，背後原理我們將於課程第三單元討論，屆時同學可更充分理解作業系統內部運作的部分機轉。若能另在 Linux 平台完成實作，有額外加分，前述之 **calc**、**notepad** 等程式則可以 Linux 環境中的其他既有的程式替代。

底下為完成的 shell 執行畫面供同學參考。剛載入 shell 之初螢幕顯示簡單的宣告訊息，隨後顯示“Wunderburg ->”之提示詞(prompt)並等待使用者輸入命令。例中的 Wunderburg 係 shell 程式經由 **hostname.exe** 自動取得而設入。倘提示詞後鍵入 **calc** <Enter> 命令，則以前景模式啟動「小算盤」程式，畫面如下：



若鍵入 `calc` & <Enter>, 接著鍵入 `notepad sysCall.c` & <Enter> 命令(以「記事本」編輯 `sysCall.c` 檔案), 程式畫面將如下所示。注意 shell 此時仍可正常接受使用者鍵入的新命令, 不被阻塞, 被叫出的「小算盤」與「記事本」均與 shell 可同時獨立運作。



鍵入 `product 5 2 10 -3 4` <Enter> 隨後在命令列鍵入 `time` <Enter> 所得畫面如下：

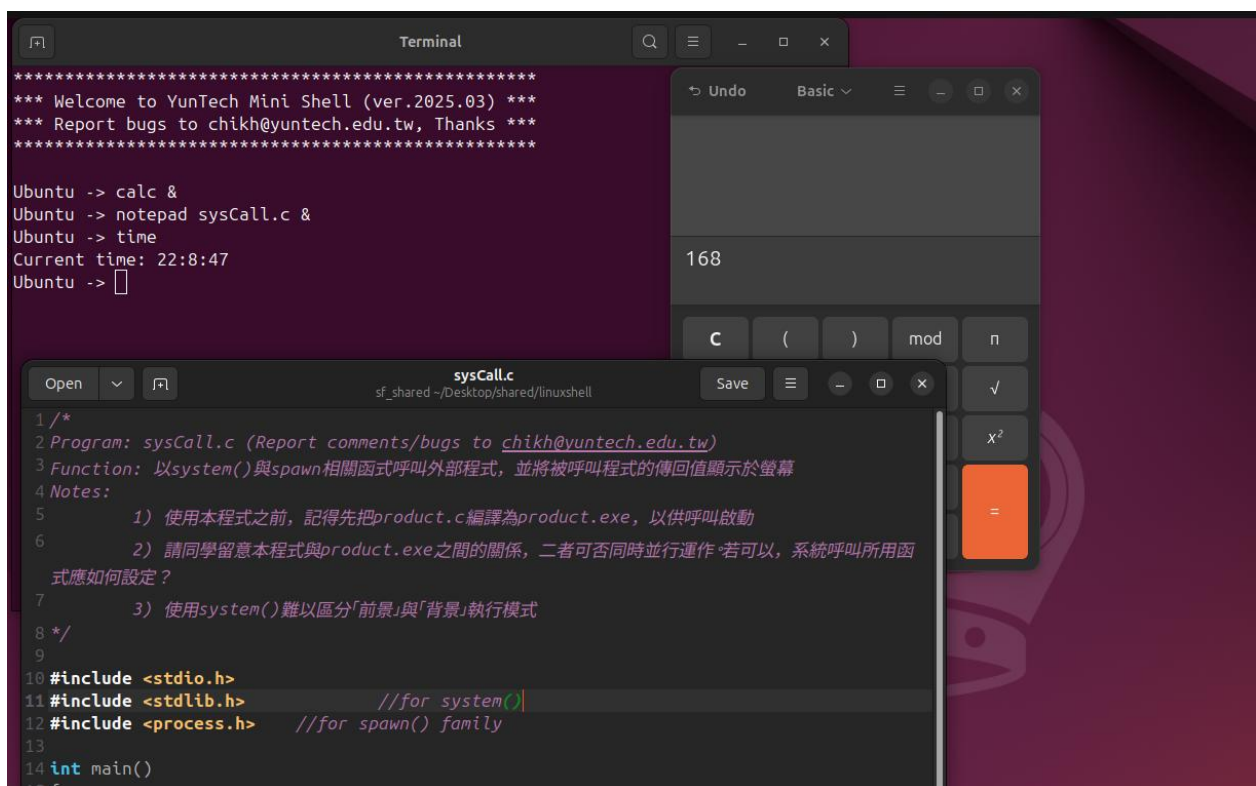
```
D:\MyPrograms\OS\shell\mys  X + v
*****
*** Welcome to YunTech Mini Shell (ver.2025.03) ***
*** Report bugs to chikh@yuntech.edu.tw, Thanks ***
*****

Wunderburg -> calc
Wunderburg -> calc &
Wunderburg -> notepad sysCall.c &
Wunderburg -> product 5 2 10 -3 4
product.exe讀入5個數，乘積=-1200
Wunderburg -> time

目前時刻：21:17:58

Wunderburg -> |
```

以下是 Linux Ubuntu 環境完成的 shell 執行畫面供同學參考，主機名稱改設為 Ubuntu 為例：



The screenshot displays a Linux Ubuntu desktop environment. In the background, a terminal window titled "Terminal" shows the execution of the YunTech Mini Shell. The shell's welcome message and the user's commands are visible. In the foreground, a calculator application is open, showing the number 168. Below the calculator, a code editor window titled "sysCall.c" is open, displaying the source code for the program. The code includes comments in Chinese and C code for system calls.

```
Terminal
*****
*** Welcome to YunTech Mini Shell (ver.2025.03) ***
*** Report bugs to chikh@yuntech.edu.tw, Thanks ***
*****

Ubuntu -> calc &
Ubuntu -> notepad sysCall.c &
Ubuntu -> time
Current time: 22:8:47
Ubuntu -> |

sysCall.c
sf_shared ~/Desktop/shared/linuxshell

1 /*
2 Program: sysCall.c (Report comments/bugs to chikh@yuntech.edu.tw)
3 Function: 以system()與spawn相關函式呼叫外部程式，並將被呼叫程式的傳回值顯示於螢幕
4 Notes:
5     1) 使用本程式之前，記得先把product.c編譯為product.exe，以供呼叫啟動
6     2) 請同學留意本程式與product.exe之間的關係，二者可否同時並行運作 若可以，系統呼叫所用函
   式應如何設定？
7     3) 使用system()難以區分「前景」與「背景」執行模式
8 */
9
10 #include <stdio.h>
11 #include <stdlib.h>           //for system()
12 #include <process.h>         //for spawn() family
13
14 int main()
15 {
```

以上為程式要求達成的功能，其餘未規範的項目可自行設計。