

呼叫外部程式與副程式

常用的呼叫外部程式有下列幾種方式

1. call 外部程式

從批次檔中呼叫外部程式並可傳遞參數，在同一個 shell 環境下，可共同存取相同的環境變數。

2. rem test.cmd

3. @echo off

4. call test1.cmd hello

5. pause

6. rem test1.cmd

7. @echo off

8. echo %1

9. timeout 6

10. cmd /c 外部指令

呼叫一個新的 shell 程式(cmd)並於指令執行完成後結束這個 shell，返回原來的 shell 環境，外部程式執行在新的 shell 之中，因此存取的环境變數與原程式不同，有區域變數的遮蔽效果

11.@echo off

12.cmd /c notepad.exe

13.exit

留意上述範例執行情形，在 notepad.exe 關閉以前，不會執行 exit 指令，因為指令尚未結束，還留在新的 shell 當中。

14. 開始一個新視窗執行程式 => start [program] [parameters]

因為 cmd /c 具有同步特性(會等外部程式執行完畢)，因此不太適合呼叫需與使用者互動的視窗程式，因為命令提示視窗會因等待而保持開啟，所以適合改用 start 來呼叫執行。

15.@echo off

16.start notepad.exe

17.exit

使用 /wait 也可以改讓 start 具有同步特性，詳情參考 start /?

副程式呼叫

批次檔的副程式呼叫也是利用 `call` 指令，不同的是 `call` 的對象不是外部程式，而是相同檔案中的標籤，也可以傳遞參數。

因為批次檔的副程式僅利用標籤代表區塊的開始，因此副程式都放在程式的尾部，之後就不要寫任何命令敘述了，代表區塊的結束。

因為批次檔循序讀取的特性，就算副程式沒有被呼叫，也會被當作標籤一般順序執行下來，因此在副程式之前通常要加上 `goto :EOF` 強制結束批次檔。

- `call :標籤 參數 1 參數 2...`

```
@echo off
rem for 迴圈使用方式請參考本手冊相關章節
for %%i in (*.dll *.exe) DO CALL :SubRoutin "%%i"
pause
goto :EOF

:SubRoutin
echo %1, %~n1, %~x1
```