UNIX SHELL

在這個專案中，你將建立一個簡單的 Unix shell。Shell 是命令列介面的核心，也是 Unix/C 程式設計環境的中心。這個作業有三個具體目標：

1.進一步熟悉 Linux 程式設計環境。

2.學習如何建立、銷毀和管理程序。

3.體驗 shell 所需的基本功能。

概述

你將實作一個命令列直譯器（CLI），也就是所謂的 shell。Shell 的基本運作方式如下：當你在提示符下輸入指令時，shell 會建立一個子程序來執行你輸入的指令，然後在執行完畢後再次提示你輸入。你要實作的 shell 會類似於你每天在 Unix 上使用的 shell，但會更簡單。

你的基本 shell，稱為 myshell，本質上是一個互動式迴圈：它會重複印出提示 myshell>（注意大於號後有一個空格），解析輸入，執行該行指定的指令，並等待指令執行完畢。這個迴圈會持續，直到使用者輸入 exit。你可以在每個指令後印出一個空行以便於閱讀。例如：./myshell

myshell> ./mycat in.txt

bird chirps

bird chirps

dog barks

dog barks

myshell> ./myhead in.txt

bird chirps

bird chirps

dog barks

myshell> ./myuniq in.txt

bird chirps

dog barks

myshell> exit

這種模式稱為互動模式，允許使用者直接輸入指令。你應該設計 shell，讓它為每個新指令建立一個程序。**你不能用 exec 執行原生 Unix 指令**，而是要將每個指令對應到一個獨立的 C 檔案。例如，當使用者輸入 cat filename 時，你的 shell 應該執行目前目錄下的 cat.c 程式。

Shell 會在 while 迴圈中運作，不斷詢問要執行什麼指令，然後執行該指令。這個迴圈會一直持續，直到使用者輸入內建指令 exit，這時 shell 結束。執行指令時，請查閱 fork()、execvp()、wait()、waitpid() 的 man page，並閱讀相關書籍章節。請記住，如果 exec 成功，將不會返回；如果返回，表示發生錯誤（例如指令不存在）。

要實作的指令

你需要實作以下指令，這些指令類似於原生 UNIX 指令，但功能有限：

1. **cat**：
   - **說明**：顯示檔案內容或合併檔案並顯示。為簡化，最多只允許兩個檔案。
   - **用法**：./mycat filename （單一檔案）或 ./mycat file1 file2 （合併檔案）。
2. **head**：

- **說明**：顯示檔案的前 3 行。如果檔案少於 3 行，則顯示全部。
- **用法**：./myhead filename

3. **uniq**：
   - **說明**：移除檔案中**相鄰**的重複行並顯示。加上 -c 選項時，顯示每行出現次數。
   - **用法**：./myuniq filename （顯示唯一行）或 ./myuniq -c filename （顯示唯一行及次數）。

假設檔案中每一行最多 1024 個字元。

輸出重定向

通常 shell 使用者會希望將程式輸出導向檔案而不是螢幕。一般 shell 用 > 字元來實現這個功能。你要讓你的 shell 也有這個功能，但用 “-o” 來取代 >。例如：

```
./mycat filename -o out.txt
```

這時不應該在螢幕上印出任何東西，而是將內容寫入 out.txt。-o 必須出現在其他參數之後、輸出檔案名稱之前。

錯誤處理

防禦性程式設計在作業系統中很重要：OS 遇到錯誤時不能直接失敗，而是要檢查所有參數。你的程式不應該 core dump、無限等待或異常終止。像原生終端機一樣，遇到錯誤（如 cat non_existent_file ），應該印出錯誤訊息（如 “No such file or directory”），然後繼續提示輸入。你必須合理處理所有輸入。以下情況都算錯誤：

- shell 程式的命令列參數數量不正確

- 錯誤的旗標（如 -t 而不是 -o，或缺少 -）
- 超過一個檔名或檔名無效

myshell 程式應該印出錯誤訊息（到 stderr），然後繼續提示新指令。

提交內容

你要將每個指令實作為獨立程式，需提交以下檔案：

1.myshell.c
2.mycat.c
3.myhead.c
4.myuniq.c
5.提供的 makefile
6.提供的測試用文字檔

每個檔案都是獨立的程式，有自己的 main 函式。myshell.c 負責建立程序來執行這些程式。這些程式不會連結在一起，而是各自編譯成可執行檔。你可以用提供的 makefile 和測試檔案來測試。

測試

請徹底測試你的程式。多嘗試各種輸入，確保 shell 表現正常。可以在另一個終端機執行原生 Unix 指令或查閱 man page 來了解每個指令。良好的程式來自於充分測試。