
Hw01-3 : Shell it! (level one)

A homework of making your own shell

Due date: 5/26(SUN.) 11:59 noon (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，我們將要求你實做一個程式來模仿 UNIX-like 系統中非常常見的一隻程式 shell，只是功能上會大幅縮減以符合作業預期的難度。在這個作業中我們假設你已經對 C/C++ 的程式設計有相當的瞭解，因此，你將於作業中專注於使用 fork / exec 等相關家族指令以及各種字串/輸出入處理來完成工作。

***此作業僅可使用 C/C++、執行環境為 ubuntu 18.04 LTS / mac OS X 10.14 (無保證)**

在這個作業中，你必須從零開始打造一個 shell（共分兩個關卡）這意味著你在 Level1 必須自行處理使用者輸入、切割輸入成許多單獨的指令，並且依據需要把他們進行輸入與輸出的串接。我們預期你未來（Level 2）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：<https://manpages.ubuntu.com/>

基本上我們也允許你使用任意的 open source 函式庫(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 Level1 中，本 Shell 應至少包含以下幾個基本功能：**

1. 應該要有命令提示字元，並且在提示字元的前面加上 **使用者帳號名稱**及**所在目錄的位置**。舉例：“**ichenyeh@/home/OS/>**” 表示使用者名稱為 **ichenyeh**，所在目錄為**路徑是/home/OS 的資料夾內**，注意，使用者名稱跟路徑都需要根據執行環境更新。

2. 可以接收並回應任意不帶參數的指令。舉例：

ichenyeh@/home/OS/> ls << **使用者輸入 ls** 並按下 Enter
[CMD] ls << **你回應[CMD] + 使用者輸入的指令**

3. 可以接收並任何帶有參數的指令。舉例：

ichenyeh@/home/OS/> ls -al << **使用者輸入 ls** 並加上附加參數 **"-al"** 最後按下 Enter
[CMD] ls [ARGV] -al << **你回應[CMD] + 使用者輸入的指令 以及 [ARGV] + 參數列表**

***多個參數間用,來間隔，舉例：**

ichenyeh@/home/OS/>ls -al -ap
[CMD] ls [ARGV] -al, -ap

-
4. 可以透過特殊指令 `exit` 來離開這個 shell 並且印出離開訊息。舉例：

```
ichenyeh@/home/OS/> exit
```

ByeBye!

> (這邊回到使用者本來的 shell)

5. 提供使用者切換目錄的功能。舉例：`cd..`, `cd [folder]`，以下為連續操作的舉例：

```
ichenyeh@/home/OS/> cd ..
```

```
ichenyeh@/home/> cd ABC
```

```
ichenyeh@/home/ABC/> cd /etc
```

```
ichenyeh@/etc/> exit
```

ByeBye!

(其中紅字為使用者輸入)

* 注意，在操作過程中的當前路徑變化，不需要影響外層的 shell

*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：OShw13_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：OShw13_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

下頁還有補充說明～

***附註：**本次作業測試環境使用 Ubuntu 18.4LTS，可額外搭配以下套件
libreadline-dev, libreadline6, libtinfo-dev, readline-doc (使用 GNU readline 需要),
clang (編譯用，必備)
可使用 apt-get 來安裝這些套件 教學請[按此](#)
****你不一定要用 readline 來完成，如果只要達成基本要求可以手刻即可**

助教編譯程式的預設指令：

clang++ XXX.cpp -lreadline -o XXX.out

clang++ XXX.cpp -o XXX.out

(請務必在你的報告中說明清楚你的程式如何編譯)

指令含意：運用 clang 來進行 XXX.cpp 程式碼的編譯（同時連結名為 readline 的函示庫），並將編譯好的程式輸出成 XXX.out（副檔名非必要）

-lreadline 為可選用選項

執行程式的指令: ./XXX.out

認識編譯：<https://www.slideshare.net/jserv/how-a-compiler-works-gnu-toolchain>

開始使用 clang：http://clang.llvm.org/get_started.html

Linux 教學參考：

1. 官方常用指令集：[這裡](#)
2. 鳥哥的基礎文件，推薦先快速看過前七課：[這裡](#)