

Operation system

Homework1-4

1063513廖婉貽

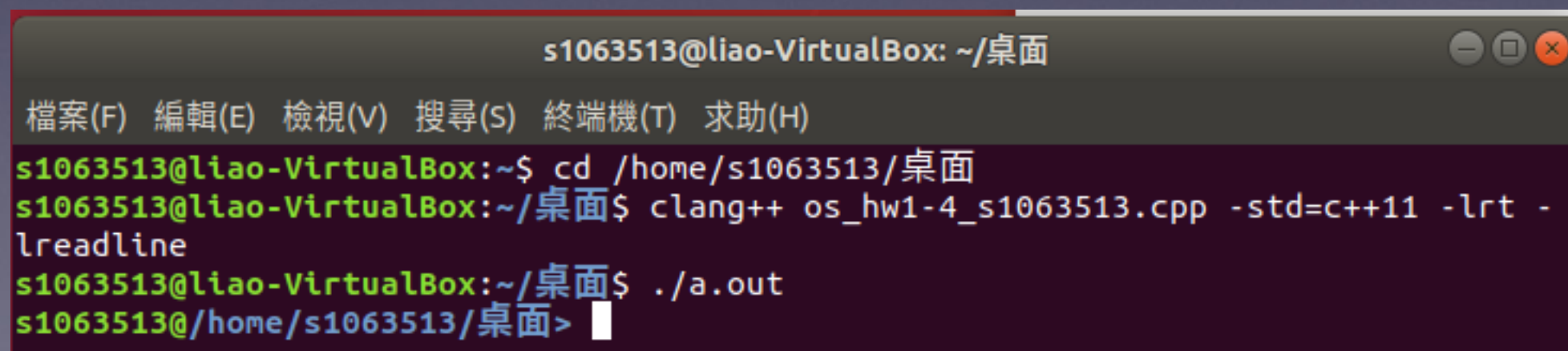
Introduction

- 套件

- ▶ clang
- ▶ libreadline6-dev

- 執行方式

- ▶ clang++ OS_hw1-3_s1063513.cpp -std=c++11 -lrt -lreadline
- ▶ ./a.out



```
s1063513@liao-VirtualBox: ~/桌面
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)
s1063513@liao-VirtualBox:~$ cd /home/s1063513/桌面
s1063513@liao-VirtualBox:~/桌面$ clang++ os_hw1-4_s1063513.cpp -std=c++11 -lrt -lreadline
s1063513@liao-VirtualBox:~/桌面$ ./a.out
s1063513@/home/s1063513/桌面> 
```

Step1&2

可以接收並執行任意不帶參數的指令
可以接收並執行任何帶有參數的指令

```
else
{
    vector<string> store; //store用來儲存指令與參數
    pid_t pid;
    pid=fork();           //fork出子程序
    int counter = 0;
    store.push_back(cut); //儲存指令與參數
    while(getline(ss,cut,' ')) //分割字串，遇到space就分割一次，並且counter++
    {
        store.push_back(cut); //儲存指令與參數
        counter++;
    }
    if(pid == 0) //pid == 0時代表是子程序,才可用execlp()執行指令
    {
        if(counter == 0) //counter == 0,代表只有指令
            execlp(store[0].c_str(),store[0].c_str(),NULL);
        else //counter != 0,代表指令帶有參數
            execlp(store[0].c_str(),store[0].c_str(),store[1].c_str(),NULL);
    }
    else //父程序等待子程序執行
        wait(NULL);
}
```

Step1 & 2

可以接收並執行任意不帶參數的指令
可以接收並執行任何帶有參數的指令

- Output

```
s1063513@liao-VirtualBox: ~/桌面
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)
s1063513@liao-VirtualBox:~$ cd /home/s1063513/桌面
s1063513@liao-VirtualBox:~/桌面$ clang++ os_hw1-4_s1063513.cpp -std=c++11 -lrt -lreadline
s1063513@liao-VirtualBox:~/桌面$ ./a.out
s1063513@/home/s1063513/桌面> ls
aaa.cpp  abcde  op.cpp          os_hw1-4_s1063513.cpp
abb      a.out  OS_hw1-3_s1063513.cpp
s1063513@/home/s1063513/桌面> ls -al
總計 68
drwxr-xr-x  4 s1063513 s1063513  4096  6月  1 14:24 .
drwxr-xr-x 15 s1063513 s1063513  4096  6月  1 14:25 ..
-rw-rw-r--  1 s1063513 s1063513  5163  6月  1 13:25 aaa.cpp
drwxrwxr-x  2 s1063513 s1063513  4096  5月 31 16:29 abb
drwxrwxr-x  2 s1063513 s1063513  4096  5月 31 16:33 abcde
-rwxr-xr-x  1 s1063513 s1063513 30864  6月  1 14:24 a.out
-rw-rw-rw-  1 s1063513 s1063513  1659  5月  7  2017 op.cpp
-rw-rw-r--  1 s1063513 s1063513  2533  5月 31 11:49 OS_hw1-3_s1063513.cpp
-rw-rw-r--  1 s1063513 s1063513  2490  6月  1 04:21 os_hw1-4_s1063513.cpp
s1063513@/home/s1063513/桌面> █
```


Bonus_A

提供使用者可以用方向按鍵(上下)
來瀏覽之前輸入過的指令

```
while(1)
{
    string cmd;
    char cwd[100];
    getcwd(cwd, sizeof(cwd)); //利用getcwd()來獲得所在目錄並存於cwd
    string output;
    output += GREEN;
    output += getenv("USER"); //利用getenv("USER")來獲得user id
    output.append("@");
    output += BLUE;
    output += cwd;
    output.append("> ");
    output += WHITE;
    cmd = readline(output.c_str()); //利用readline()來獲得user輸入的指令
    add_history(&cmd[0]); //讀取命令後用add_history()將命令存入歷史列表中
    stringstream ss(cmd);
    string cut;
    getline(ss, cut, ' ');
```

Bonus_A

提供使用者可以用方向按鍵(上下)
來瀏覽之前輸入過的指令

- Output

```
s1063513@liao-VirtualBox: ~/桌面
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)
s1063513@liao-VirtualBox:~$ cd /home/s1063513/桌面
s1063513@liao-VirtualBox:~/桌面$ clang++ os_hw1-4_s1063513.cpp -std=c++11 -lrt -lreadline
s1063513@liao-VirtualBox:~/桌面$ ./a.out
s1063513@/home/s1063513/桌面> ls
aaa.cpp  abcde  op.cpp                os_hw1-4_s1063513.cpp
abb      a.out  OS_hw1-3_s1063513.cpp
s1063513@/home/s1063513/桌面> ls -al
總計 68
drwxr-xr-x  4 s1063513 s1063513  4096  6月  1 14:24 .
drwxr-xr-x 15 s1063513 s1063513  4096  6月  1 14:25 ..
-rw-rw-r--  1 s1063513 s1063513  5163  6月  1 13:25 aaa.cpp
drwxrwxr-x  2 s1063513 s1063513  4096  5月 31 16:29 abb
drwxrwxr-x  2 s1063513 s1063513  4096  5月 31 16:33 abcde
-rwxr-xr-x  1 s1063513 s1063513 30864  6月  1 14:24 a.out
-rw-rw-rw-  1 s1063513 s1063513  1659  5月  7  2017 op.cpp
-rw-rw-r--  1 s1063513 s1063513  2533  5月 31 11:49 OS_hw1-3_s1063513.cpp
-rw-rw-r--  1 s1063513 s1063513  2490  6月  1 04:21 os_hw1-4_s1063513.cpp
s1063513@/home/s1063513/桌面> █
```



```
s1063513@/home/s1063513/桌面> ls -al█
```

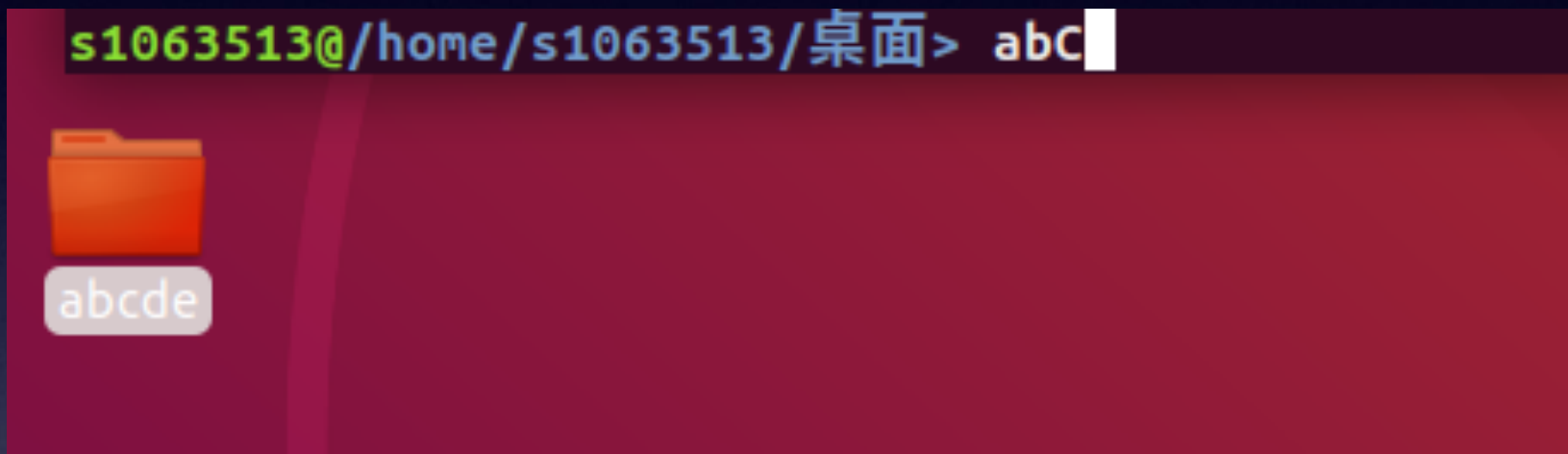


```
s1063513@/home/s1063513/桌面> ls█
```

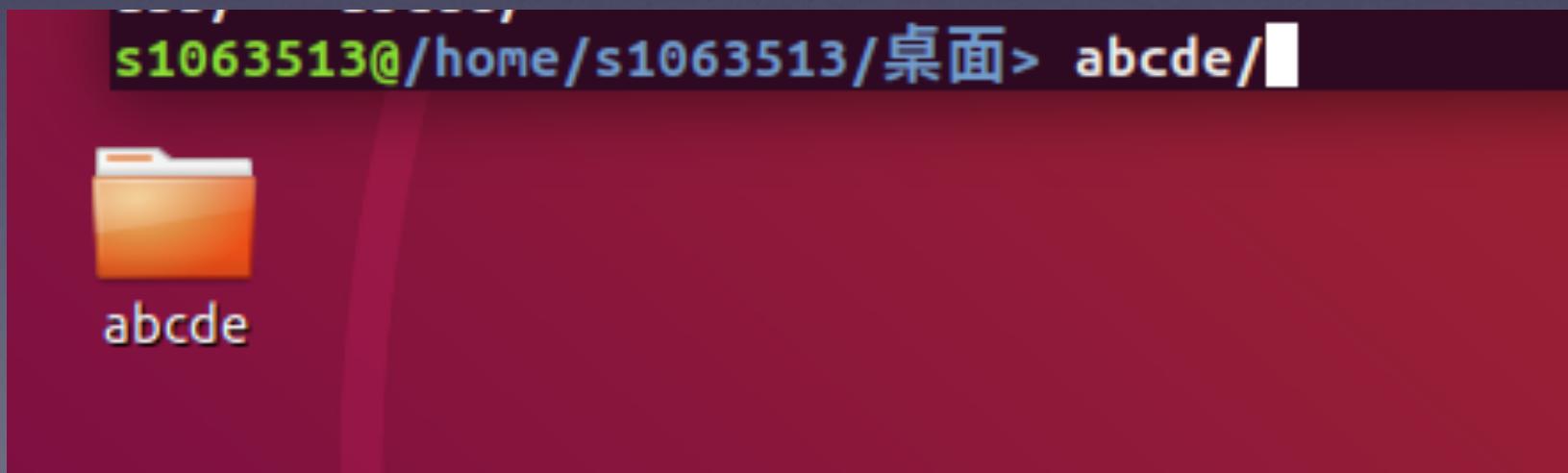
Bonus_B

使用 Tab(或是雙擊 Tab)來進行檔名自動完成

- Output



雙擊 Tab後



Conclusion

在做**Bonus_A**時一開始我輸出的**user**會因為加上新功能而被吃掉，和同學一起研究了許久才發現如果將輸出輸入皆用**readline()**就不會產生這個問題了。