

編譯指令:clang OShw14\_s1061443.c -o a.out

執行指令:./a.out

## 1. ls, ls -al

```
peter@peter-VirtualBox:~/OS$ clang OShw14_s1061443.c -o a.out
peter@peter-VirtualBox:~/OS$ ./a.out
peter@home/peter/OS$ ls
a.out      ls.c      OShw13_s1061443.c  shell.out
hw1-3.out  myoutput  OShw14_s1061443.c
peter@home/peter/OS$ ls -al
總計 88
drwxr-xr-x  2 peter peter  4096  5月 30 00:21 .
drwxr-xr-x 16 peter peter  4096  5月 30 00:04 ..
-rwxr-xr-x  1 peter peter 14592  5月 30 00:21 a.out
-rwxr-xr-x  1 peter peter 13488  5月 29 23:24 hw1-3.out
-rw-r--r--  1 peter peter 10375  5月 27 23:13 ls.c
-rw-r--r--  1 peter peter   76  5月 30 00:20 myoutput
-rw-r--r--  1 peter peter  4624  5月 29 23:32 OShw13_s1061443.c
-rw-rw-rw-  1 peter peter  6770  5月 30 00:15 OShw14_s1061443.c
-rwxr-xr-x  1 peter peter 14584  5月 30 00:15 shell.out
```

## 2. pipe, redirect

```
peter@home/peter/OS$ ls > output
peter@home/peter/OS$ cat output
a.out
hw1-3.out
ls.c
myoutput
OShw13_s1061443.c
OShw14_s1061443.c
output
shell.out
peter@home/peter/OS$ ls | head -3
a.out
hw1-3.out
ls.c
peter@home/peter/OS$ ls | head -3 | tail -1
ls.c
peter@home/peter/OS$ exit
ByeBye!
peter@peter-VirtualBox:~/OS$
```

ls:

call 內建指令完成。

pipe, redirect:

從 command()開始，先 pipe()與 fork()，尋找指令中的<與>來判斷是做

redirect\_input 或 redirect\_output。

`void with_pipe_execute()`先把有|的指令切割把沒有|的指令分開

`static int split(char *cmd_exec, int input, int first, int last)`這個時候如果指令沒有|，就會在這裡執行，ex: cd, exit。

`static int command(int input, int first, int last, char *cmd_exec)`這裡是 pipe 與 fork 產生的地方，redirect 也是在這裡執行。

`void tokenise_redirect_input_output(char *cmd_exec)`

`void tokenise_redirect_input(char *cmd_exec)`

`void tokenise_redirect_output(char *cmd_exec)`

不同 redirect 的方式 call 不同的 function()。