B03102082
陳孝思

SA HW1

1 Short Answer Question

1.1 More Permissions!

Dir Permission	Delete Rename Create files	List dir (ls)	Read file content	Write file content	cd dir	cd subdir	List subdir (ls)	Access subdir files
--- (0)	F	F	F	F	F	F	F	F
-w- (2)	F	F	F	F	F	F	F	F
r-- (4)	F	T	F	F	F	F	F	F
rw- (6)	F	T	F	F	F	F	F	F
--x (1)	F	F	T	T	T	T	T	T
-wx (3)	T	F	T	T	T	T	T	T
r-x (5)	F	T	T	T	T	T	T	T
rwx (7)	T	T	T	T	T	T	T	T

(1) Brief conclusion:

r 決定資料夾下的檔案名稱與子資料夾名稱是否能被讀取

w 決定資料夾裡的檔案與子資料夾能否被刪除或改名，也決定能否在資料夾下建立新檔案與新資料夾

x 決定能否進入資料夾，且決定其他程序能否經過此資料夾，去存取此資料夾下的檔案或子資料夾；而在沒有設 x 的情況下，即使有 r，也只能讀到檔名，讀不到檔案資訊，即使有 w，也不能建立或刪除檔案與改檔名

ref: https://unix.stackexchange.com/questions/21251/execute-vs-read-bit-how-do-directory-permissions-in-linux-work

http://linux.vbird.org/linux_basic/0210filepermission.php - filepermission_dir

(2) POSIX.1-2008 規定 symbolic link 本身必須為可讀，但是沒有規定一個 symbolic link 的 stat structure 裡的 st_mode 必須長怎樣。在 linux 上的 permission 為 lrwxrwxrwx，且不能更改；但在 mac 和 FreeBSD 上，可以更改 symbolic link 本身的 permission

ref: https://superuser.com/questions/303040/how-do-file-permissions-apply-to-symlinks

(3)

(a). setuid：一個檔案如果有 setuid，則執行該檔案後，那個 process 的 effective UID 會變成該檔案的擁有者，用於短暫提升 process 的權限，讓 other 群組的使用者在執行該檔案時，得到檔案擁有者的權限

(b). setuid: 類似於 setuid，只是該 process 的 effective groud ID 變成檔案擁有者的 group id

(c). sticky bit: 如果資料夾有 sticky bit，表示該目錄下，對於任一檔案與子資料夾來說，只有檔案的擁有者或 superuser 可以刪除或更改檔案的名稱，用途是讓共享資料夾下的檔案不會被非檔案擁有者亂刪亂改名

如果非資料夾的檔案擁有 sticky bit，在 linux 上會被忽略

Ref: https://systemprogrammingatntu.github.io/quiz3

http://linux.vbird.org/linux_basic/0220filemanager/0220filemanager.php - suid_sgid_sticky

https://ceiba.ntu.edu.tw/course/17b448/content/W5 Chap 4.pdf

https://www.youtube.com/watch?v=Soo-Yqp2oXY&t=0s&index=1&list=PLtstG443LLj8ZoicrlQ24XlK7vqnl6pwW

https://en.wikipedia.org/wiki/Sticky_bit

1.2 Deeper, deeper

(1) 從 env 裡找到 bash 的路徑來用 bash 執行以下 script

(2) 一個是絕對路徑，另一個是去 env 裡面找 bash 的路徑，各有各的優缺點，用絕對路徑的話，可以加上執行的參數，但如果不確定 bash 的路徑，可能會無法執行，等於說換一台電腦可能無法找到 bash 來執行；去 env 裡找 bash 的話代表不會因為換了電腦，相同的 script 會找不到 bash，但缺點是無法吃參數

Ref:

https://www.zhihu.com/question/68052314

https://en.wikipedia.org/wiki/Shebang_(Unix)

https://unix.stackexchange.com/questions/29608/why-is-it-better-to-use-usr-bin-env-name-instead-of-path-to-name-as-my/29620 - 29620

1.3 Copy Monster

各有各的優點，rsync 會將要拷貝的檔案拆成好幾個大小一樣的片段，並且做雜湊值運算，只傳輸雜湊不一樣的片段，可以做到一半暫停，之後繼續剩下的部分，傳輸大檔案時，發生問題中斷沒關係，可以從被中斷前的進度開始，完成剩下的傳輸。cp 就比較簡單，沒有這些功能，被中段就得從頭開始，也不能暫停之後，從已完成的進度開始，也不會將檔案拆開來，一一作雜湊運算

rsync 適合傳輸大檔案和要同步兩端資料的時候使用，cp 適合單純複製小檔案

Ref: http://mpov.timmorgan.org/use-rsync-instead-of-cp/

https://serverfault.com/questions/43014/copying-a-large-directory-tree-locally-cp-or-rsync

https://www.google.com.tw/url?sa=t&rct=j&q=&esrc=s&source=web&cd=8&cad=rja&uact=8&ved=0ahUKEwiLqMXowc_ZAhUJOJQKHSh8ABUQFghrMAc&url=https%3A%2F%2Fwww.quora.com%2FWhat-is-the-difference-between-cp-and-rsync&usg=AOvVaw1TdMclRNlyPSxsAKIA3vNb