System Administration HW1

B04705003 資工三 林子雋

1 Reference

1.1 Short Answer Questions

1.1.1 More Permissions!

(1) Execute vs Read bit. How do directory permissions in Linux work?

(2) (a) How do file permissions apply to symlinks?

(b) Why chmod cannot change symbolic link permission?

(3) Special File Permissions (setuid, setgid and Sticky Bit)

1.1.2 Deeper, deeper

(1) (a) Why is #!/usr/bin/env bash superior to #!/bin/bash?

1.1.3 Copy Monster

1. https://stackoverflow.com/questions/6339287/copy-or-rsync-command
2. https://everythinglinux.org/rsync/
3. What is the difference between ‘dd‘, ‘cp‘ and ‘rsync‘?

2 Short Answer Questions

2.1 More Permissions!

(1) Conclusion: ”Delete, rename and create files” requires ”wx”; ”List dir” requires ”r”; ”Read file content” requires ”x”; ”Write file content” requires ”x”; ”cd dir” requires ”x”; ”cd subdir” requires ”x”; ”List subdir” requires ”x”; ”Access subdir files” requires ”x”.

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

(2) In Linux, symbolic link will have same permission as the file it points to, which means symbolic link's permission itself is never used.

(3) (a) setuid: If a executable file's setuid bit is set, when a user runs it, his process will be granted the file owner's permission. (The effective gid of his process will be changed to file owner's real uid.)

(b) setgid: When a user runs a executable file whose setgid bit is on, his process will be granted the file owner's group permission. (The effective gid of his process will be changed to file owner's real gid.)

(c) sticky bit: If a file has the sticky bit on, that file can only be deleted by its owner or root.

2.2 Deeper, deeper

(1) It will search "PATH" for bash(because "bash" may not always in "/usr/bin/", particular on non-Linux system).

(2) 1. Because "bash" may not in "/usr/bin/" folder. 2. From a portability perspective, I think "#!/usr/bin/env bash" is better than "#!/usr/bin/bash", because "#!/usr/bin/env bash" can be compatible in different operating system.

2.3 Copy Monster

For the comparison between "cp" and "rsync", please refer to Table 1.

When should we use "rsync" or "cp"?: I think if there are many news file you want to copy from, you may choose "cp" because "cp" requires less checking algorithm. However, if you want to synchronize two folders with just several files, you will better choose "rsync", because it will only update parts of the updated files.

Which is better?: I think it depends on what scenario you encounter. If you want to copy many files which are all new, you should choose ”cp”. But if you want to synchronize folders, you will better choose ”rsync”.

Table 1: Comparison between cp and rsync

	cp	rsync
Support remote copy	No	Yes
Support ssh	No	Yes
Support compression	No	Yes
Rsync Algorithm	No	Yes