Lec-04-2
Advanced Permissions

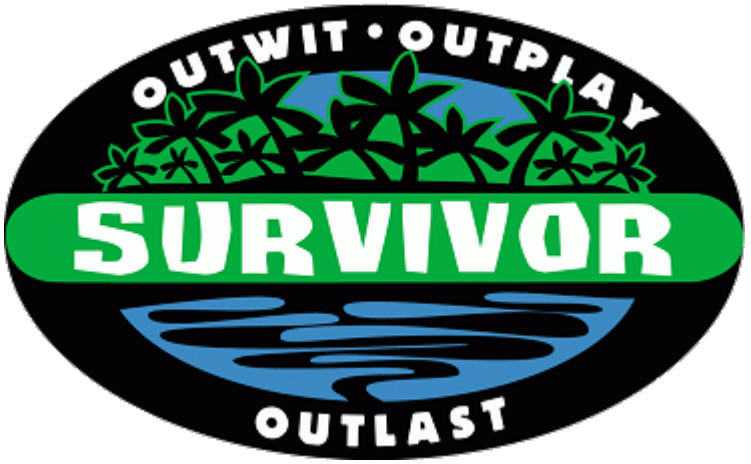
Dr Syed Faisal Hasan and Dr. Hymie Latif
Computing and Information Technology
College of Enterprise and Development
Otago Polytechnic
Dunedin, New Zealand

Bachelor of Information Technology
IN616 – Operating Systems Concepts
Semester 1, 2020

Schedule

- Recap
- Advanced permissions
  - Default permissions and **`umask`**
- Linux File System Principles

Are you going to ***survive*** Linux?



- **Are you up to date?**
  - Finished all labs
  - Finished extra readings
  - Understand the primary concepts we have covered?
  - Been practicing at home? ← This is key

- I would not recommend doing a late dash in this paper (the late assessments are the hardest)

TOPIC:

File System Permissions: Default Permissions

Default Permissions

- We know how permissions work:

  `user, group, other (`**`ugo`**`)`

  `read, write, execute (`**`rwx`**`)`

  `-rwxrwxrwx == 777`

- We know how to set owners, set permissions:

  `chown`

  `chmod`

- **But what about default permissions?**
  - When we create a file
  - When we create a directory

Default Permissions and **umask**

- Directory and file creation have **base permissions**
  - Directories → 777
  - Files → 666
- Setting **umask** can modify default permissions
- ***base permission – umask = effective permission***

- **Effective permission:**
  - The permissions used when creating files/directories
- So what is **umask**?
  - A command/method to set default permissions

Default Permissions: **Files**

- **`umask 002`**

user value is	0
group value is	0
other value is	2

`base permissions (666)`

user value is	6
group value is	6
other value is	6

- File permissions:
  - `user: 6 - 0 = 6`
  - `group: 6 - 0 = 6`
  - `other: 6 - 2 = 4`
- ***So what are the resultant permissions?***

`664` **OR** **`rw-rw-r--`**

Default Permissions: **Directories**

- **`umask 002`**

		`base permissions (777)`	
user value is	0	user value is	7
group value is	0	group value is	7
other value is	2	other value is	7

- Directory permissions:
  - `user: 7 - 0 = 7`
  - `group: 7 - 0 = 7`
  - `other: 7 - 2 = 5`
- ***So what are the resultant permissions?***

`775` **OR** **`rwxrwxr-x`**

`umask` Syntax

- **`umask <permission-specification>`**

- Method 1: octal values
  - **`umask 002`**
- Method 2: symbolic representation
  - Remember we can use: **`+`** to add, **`-`** to remove, **`=`** to exact
  - **`umask -S u+w`** → adds a specific permission
  - **`umask -S u-w`** → removes a specific permission
  - **`umask -S u=w`** → removes all other but specified permission
- To make umask permanent, add entry to: **`~/.bashrc`**

`umask` Symbolic Examples

- **`umask -S g+w`**

  Groups have write permissions

- **`umask -S u+r,g=w,o-x`**

  Users have read, group write, and others execute permissions

- **`umask -S u+r,g-w,o=rwx`**

  Users have read, group NOT write, and others can do anything

- NOTE: You need to know octal and symbollic!

umask Resources

- **umask** Quiz
  - http://www.webune.com/forums/umask-calculator.html

- All **umask** modes
  - https://www.linuxtrainingacademy.com/all-umasks/

TOPIC:

File System Permissions: Sticky Bit

Default Permissions and **umask**

- When you run **umask**, it returns 4 digits!
  - So far, we only looked at the last 3! (The permission octals)
- The first digit manages special modes

  4 → executables run with user permissions

  2 → executables run with group permissions

  1 → sets the ***sticky bit*** during file creation

  files deleted based on user ownership (not directory)

- Potentially a powerful feature
- Not used much in everyday operation
- Use **umask** with 3 digits, or 4 digits with leading 0

https://linuxconfig.org/how-to-use-special-permissions-the-setuid-setgid-and-sticky-bits

Permissions: **Sticky Bit**

- Sticky bit is a **safety net** for file management
  - If used, we will see a "**t**" in permissions
- Files/directories marked as sticky:
  - ***Can only be deleted by owner and root***

    `rw-rw-rwt` (execute permission allowed)

    `OR rw-rw-rwT` (execute permission not allowed)
- Setting sticky bit:
  - **`chmod +t filename`**
  - **`chmod 1755 filename`**
- Removing sticky bit:
  - **`chmod -t filename`**
  - **`chmod 0755 filename`**

Lab-04-2 – Start

- **TOPICS:**
- **Setting default permissions/effective permissions**
- **Understanding umask**