Linux Basics

1 User and Group

- Users
  - root: super user (uid = 0)
  - daemon: handle networks.
  - nobody: owns no files, used as a default user for unprivileged operations.
    - Web browser can run with this mode.
  - User needs to log in with a password. The encrypted password is stored in `/etc/shadow`.
  - User information is stored in `/etc/passwd`, the place that was used to store passwords (not anymore). The following is an example of an entry in this file.

    ```
    john:x:30000:40000:John Doe:/home/john:/usr/local/bin/tcsh
    ```

- Groups
  - Sometimes, it is more convenient if we can assign permissions to a group of users, i.e. we would like to assign permission based on groups.
  - A user has a primary group (listed in `/etc/passwd`), and this is the one associated to the files the user created.
  - Any user can be a member of multiple groups.
  - Group member information is stored in `/etc/group`

    ```
    % groups uid   (display the groups that uid belongs to)
    ```

  - For systems that use NIS (Network Information Service), originally called Yellow Page (YP), we can get the group information using the command `ypcat`.

    ```
    % ypcat group  (can display all the groups and their members)
    ```

2 File Permissions

- File Permissions
  - The meaning of the permission bits in Unix.
    - Owner (u), Group (g), and Others (o).
    - Readable (r), Writable (w), and Executable (x).
    - Example: -rwxrwxrwx (777)
- Permissions on Directories:
  - r: the directory can be listed.
  - w: can create/delete a file or a directory within the directory.

  - x: the directory can be entered.

- Change permission: `chmod`
- Full Access Control List: using `getfacl` and `setfacl`.
- Default File Permission
  - What is the default file permission assigned to the newly created files?
  - This default permission is stored in the `umask` environment variable.
  - `umask`: permissions you do not want
  - Default value in some systems: 022
    - This set the permission of new files (non-executable) to rw-r–r–.
  - Safest value: 077
    - This sets the permission of new files (non-executable) to rw——-.
  - Check your own setting by executing the following

    ```
    % umask
    ```

  - Change the umask value. You can execute the following command or put it in your .profile file.

    ```
    % umask 077
    ```

3 Security-Related Commands

- Switch user
  - Change your user ID to xyz, su means “substitute user”)

    ```
    % /bin/su xyz
    ```

  - Change to root. This is a common way to invoke superuser access). Once you are in the superuser account, the prompt becomes the pound sign (#).

    ```
    % /bin/su -
    ```

  - Running a command using superuser privilege. Sometimes, we just want to run a command using the superuser privilege. Instead of `su` to root, and run the command, we can use the `sudo` command.

    ```
    (view the shadow file as a superuser)
    % sudo more /etc/shadow
    ```

    To be able to use `sudo` to run a command as the superuser, permissions must be granted (by the root) to the user. This is done through the `/etc/sudoers` file.

- Change the owner of files

- The `chown` command.

```
% chown wedu file
```

- Q: Can we allow a user to change the owner of files to another user?
  - No. Actually, only root can use `chown`. Why?
  - We will understand why after we have learned `Set-UID`

- Change the group of files
  - The `chgrp` command.

```
% chgrp seed  /home/seed/785
```

  - Q: Can we allow a user to change the group of files to another group?
    - Yes/No. If you want to change to group XYZ, you must be a member of XYZ
    - The reason is similar to the `chown` command (Set-GID).

- Miscellaneous

```
% whoami (to print out your current user name)
% /usr/bin/id    (display both uid and gid)
% man chmod  (find the manual for the chmod command)
```