COMP 225: Network and System Administration
Notes #5: Users and Groups

K. L. Eddie Law, PhD
Associate Professor
Computing, MPI

Academic Year 2nd Semester, 2019-2020

Topics

- Tab completion
- Users and groups
- Access and file permissions

Tab Completion

- "bash" has a shortcut that completes filenames for you
  - Start typing a path
  - Hit tab once…
- it searches and finds a file that matches
- If there are more than one match it will do nothing, if hit it a second time, then it will show all matches

Filename Globing

***** – 0 or more characters

? – a ny 1 character

[] – matches characters in brackets
Example: **[abc]**

Example

- Consider the following directory listing

  file1.txt file2.txt file.txt coolgame coolpictures
  vacation.txt poolpictures oolgame

- What files will the following commands match?

ls *.txt	**ls file*.***
ls file?.txt	**ls ?ool***
ls [c]ool*	**ls ?????.txt**

User Creation and Management

On Users and Groups

- Different commands for adding/deleting users/groups
- Linux binary
  - `useradd`
  - `userdel`
  - `groupadd`
  - `groupdel`
- Perl scripts (more user friendly)
  - `adduser`
  - `deluser`
  - `addgroup`
  - `delgroup`

useradd

- Linux is a multi-user system
  - A special user called root has unlimited rights
  - Normal users are “un-privileged” and their rights are limited on the system
- System administrator need to be very comfortable with creating and managing users and groups
- For simplicity, as root, run for user creation

```
$ sudo adduser newUserName
```

- Traditionally, run

```
$ sudo useradd newUserName
$ sudo password newUserName
```

useradd (cont'd)

useradd [**-c** *name_field]* [**–d** *home_dir*] [**–e** *expire_date*] [**–g** *group_id*] [**–s** *shell*] [-p password_hash] username

passwd *username*

usermod

- change user information once the account has been created

usermod [**–c** *name_field*] [**–d** *home_dir*] [**–g** *group_id*] [**–l** *username*] [**–s** *shell*] [**–L**] [**-U**] *username*

- For example, if there is a group called "students", run the following to add user "frank" to the "students" group

```
$ sudo usermod -aG students frank
```


/etc/passwd

- The local users are defined in the file
- For the format of the file, fields separated by ":"

  `username:password:uid:gid:name:homedir:shell`

  where
  - username – username
  - password – x in most instances
  - uid – user id, a unique user identifier number
  - gid – group id, defines the primary group
  - homedir – personal space for users account
  - shell – the users shell

/etc/shadow

- A very important file on Unix that stores users password information
- Similar to the /etc/passwd file, each line is for one user
  - Username
  - Password hash
  - Last password change
  - Days until password can be changed again
  - Days before password expires (must be changed)
  - Days warning before password expires
  - Days after password expires that account is disabled
  - Date when account expires
  - Reserved

chfn & chsh

- File /etc/passwd contains the user configuration information
- No permissions for normal users to edit this file
- There are special programs on the system that lets a user change their shell and their name entries
  - Change name and other info

    `$ chfn`
  - Change the shell

    `$ chsh`

Programs to alter password settings

$ passwd *username* – change a users password

$ chage [options] *username* – change password policies

-l or **--list**

-E or **--expiredate** *YYYY-MM-DD*

-m or **--mindays** *number_of_days*

-M or **--maxdays** *number_of_days*

whoami, logname, id, groups

$ whoami

- displays you who you currently are

$ logname

- displays who you logged in as

$ id

- displays information about your user

$ groups

- Displays information about your group memberships

Groups

- Create groups and assign users to groups
- Access files and resources can be shared among multiple people working on the same project.
- Group information is in /etc/group, with format

  `group_name:password:group_id:[username[,]…]`

Managing Groups

 # groupadd [options] *group_name*

-g or --gid *group_id*

 # groupdel *group_name*

 # groups *username*

 # usermod --append --groups *group1[,group2…]* **username**

Note: always use the --append option; if not, the system will reset the user to ONLY be in the groups typed in the command; therefore, could accidentally remove a user from old groups!

userdel

- To delete a user in the system

userdel [**-r**] *username*

-r deletes data in users home directory and mailspool

su

- A command to switch between users
- Good security practice
- Switch from one "normal user" to another (password required)
  - (for Red Hat, Fedora) `$ su -l username`
- Can become root from a "normal user", user password needed
  - (for Red Hat, Fedora) `$ su -`
  - (for Ubuntu –root login disabled) `$ sudo su -`
- If you are already root you can become any other user without a password

Secure Shell (SSH)

- Permits us to log in a remote computer
- Apart from using "su", we can use "ssh" to log in local computer too
- ssh is a secure replacement for the legacy "telnet" program

```
$ ssh computerName -l username
```

- ssh requires that an ssh daemon (sshd) be running on the remote host, also need the password of the user for logging in

.ssh Directory

- The `.ssh` directory holds important `ssh` files, e.g.,
  - `id_rsa` - users rsa private key
  - `id_rsa.pub` - user rsa public key
  - `id_dsa` - users dsa private key
  - `id_dsa.pub` - user dsa public key
  - `authorized_keys` - users allowed to login with using digital signatures
  - `known_hosts` - known hosts and keys

A Bit More on File Permissions

Default Permissions

- The `umask` command allows a user to change the default permissions for new any file/directory
- `umask`
  - The actual permissions are "default" permissions
  - REMOVES the specified bits from the system's default creation permissions
- In general (for ubuntu: 0002, for fedora: 0022)
  - System default for files = rw-rw-r--
  - System default for directories = rwxrwxr-x
- To check the current umask, `$ umask`
- To change the umask, `$ umask new_removal_mask`

File Attributes

- Linux also has file attributes, they are not permissions
- Rarely used
- Lists the file attributes with
  `$ lsattr`
  - Usually shows "-e" the regular extent file system

  `$ chattr +i filename`
  - Add attribute to a file, make it static, cannot be removed

  `$ chattr -i filename`
  - Remove the "+i" attribute from a file

Remarks

- On users and groups
- Using ssh for remote login
- A bit more on file permissions