EASTERN WASHINGTON UNIVERSITY
start something big

Unix File System Command 4

Computer Science Department
Eastern Washington University
Yun Tian (Tony) Ph.D.

Recall Last Class

- mv : move file/directory to another place or rename a file/directory.
- rm deletes a file.
- alias creates a short cut for a command.
- clear the terminal screen.
- touch creates a empty text file.
- User and groups

Recall Permissions

- We can use **ls -l** to tell about ownership and permissions of file.
- **ls -l**

  -rw-r--r-- 1 ytian ytian 4096 Sep 26 20:52 name.txt
- -rwxr-xr-x
  - User's Permissions
  - Group's Permissions
  - Other's permissions

Today

- Change Permission
- Change group
- Deal with plain files
- Print to the terminal

chmod Symbolic

- **chmod <mode> <file>**
- Change file/directory permissions based on what is specified in the <mode> argument.
- The format of <mode> is a combination of 3 fields:
  - Who is affected : a combination of ***u***, ***g***, ***o***, or ***a*** (all)
  - Whether adding or removing permissions: **+** or **-**
  - Which permissions are being added/removed -any combination of **r**, **w**, **x**

chmod Symbolic

- For Examples
  - chmod ug+rx myfile
    - adds read and execute permissions for user and group.
  - chmod a-r myfile
    - remove read access for everyone
  - chmod ugo-rwx myfile
    - remove all permissions from myfile

chmod Numeric

- chmod with Numeric Mode
  - chmod numeric values are often given as three digits of Octal numbers, e.g. 777, 755.
    - chmod 755 my_file.cs
  - First digit defines permissions for file owner (owner permissions)
  - Second digit defines permissions for group users who belong to file's group.
  - Third digit gives permissions for all other users.

chmod Numeric

- We have three types of permissions, i.e. permission of **read**, **write** and **execute**.
- When using chmod, these three digits can be calculated digit by digit as a sum of permission values.
- Permission values are listed below.
  - +4 : Read Permission weighs 4 points.
  - +2 : Write Permission weighs 2 points.
  - +1 : Execute Permission weighs 1 points.

chmod Numeric

- We see permissions in this fashion
  - rwxr-xr-x
  - rwx is the user permission. r-x is group permission.
  - change rwx to a octal value = 7
  - change r-x to a octal value = 5

place2	place1	place0
r	w	x
$2^2 = 4$	$2^1 = 2$	$2^0 = 1$

$2^2 + 2^1 + 2^0 = 7$

place2	place1	place0
r	-	x
$2^2 = 4$	no	$2^0 = 1$

$2^2 + 0 + 2^0 = 5$

chmod Numeric

- For example if you want to give the owner write and execute permissions(-wx), to the group execute permissions(--x) and to others no permissions(---) you would calculate the correct chmod value like this:
  - First Digit: 0 + 2 + 1 = 3
  - Second Digit: 0+ 0 + 1 = 1
  - Third Digit: 0
- So you would get 310,
  - chmod 310 myfile.c

Command chgrp

- **chgrp group <target>**
  - Change the group ownership of file <target>
- If you have root access and you want to change who owns a file,
- **chown user:group <target>**
  - changes ownership of file <target>.
  - group is optional.
  - use the option "-R" to do a recursive change to a directory and the files within it.

Type of Files

- There are two main types of files.
- The first is plain text files.
  - Like something you would create in notepad.
  - Editable using many existing editors.

Type of Files

- **Binary files** are written in machine code.
  - Not human readable (at least without using hex editors)
  - Commonly used for executable, libraries, media files, zips, pdfs, etc.
  - To create need some sort of binary-outputting program.

Dealing with Plain Text

- The shell is designed to allow the user to interact in powerful ways with plain text files.
- Before we can get to the fun stuff let's cover the basics.

Dealing with Plain Text

- **nano filename**
  - Opens filename for editing
  - In terminal editor
  - Since you (most likely) will be sshing into UNIX machines, this editor will do fine for everything we do in this course.
  - Shortcuts for saving, exiting all begin with CTRL.

Reading Files

- Often we only want to see what is in a file without opening it for editing.
- **cat <filename>**
  - Prints the contents of the file to the terminal window.
- **cat <filename1> <filename2>**
  - Prints the first file then the second in the terminal.
  - Looks like content of filename2 is appended to filename1.

Reading Files

- **more <filename>**
  - allows you to scroll through the file 1 page at a time.
- **less <filename>**
  - Lets you scroll up and down by pages or lines.
  - Using **j** to scroll down, **k** to scroll up, **q** to quit.
  - Using **g** to move to the beginning of the file.
  - Using **G** to move to the end.
  - Using **h** to show help information.
  - Using **/pattern** to search the pattern in the file.

Reading Beginning and End

- Sometimes you only want to see the beginning of a file (maybe read a header) or the end of a file (see the last few lines of a log).
- **head -[numlines] <filename>**
- **tail -[numlines] <filename>**
  - Prints the first/last numlines of the file
  - Default is 10 lines

Reading Beginning and End

- tail /var/log/Xorg.0.log
  - Prints the last ten lines of the log file.

Printing to the Terminal

- We have already seen a variety of ways to print text to the screen. If we just want to print a certain string, we use,
- **echo <text_string>**
  - Prints the input string to the standard output (the terminal)
  - echo This is a string
  - echo 'This is a string'
  - echo "This is a string"
- all print the same thing
- We will see why we talk about these three cases later.

Summary

- Now you know how to change permission for a file.
  - chmod 755 myfile1
  - chmod ug+rw myfile2
- We learned nano used to edit a file
- less and more
- head and tail
- echo and cat

Next Class

- Shell shortcut keys
- History
- Special characters or metacharactgers