EASTERN WASHINGTON UNIVERSITY
start something big

find command and Quotes

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

Regular Expression

- **Basic Regular Expression MetaCharacters**

  . any one character

  [...] any character listed in a character class

  [^...] any character NOT listed in the class

  ^ beginning of line anchor

  $ end of line anchor

  \< start of word anchor

  \> end of word anchor

  | or bar ("or" logic separating expressions)

  () parentheses (limits scope of | "or bar")

  \ escape (used before a metacharacter to match a literal)

Regular Expression

- **Extended Regular Expression MetaCharacters**

  ? one optional match on preceding, no match required.

  * unlimited optional matches on preceding, no match required.

  + one match on preceding required, unlimited allowed.

Regular Expression

- In basic regular expressions the metacharacters ?, +, {, |, (, and) lose their special meaning;
- instead use the backslashed versions \?, \+, \{, \|, \(, and \).
- By Default We are using **Basic Regular Expression.**

Summary

- grep –r ‘pattern’ directory_name
- other options -n, -v, -x,
- basic regular expression
  - . * [ ] ^ $ \
  - grep ‘[^a-zA-Z]$’ myfile
  - grep –rn dbConnect --include=*.c ./

– Be default, we are using basic regular expression.

Outline for Today

- find command
- Quotes ‘ ’ and “ ” and backquotes `

find command

- **find directoryName –name targetfile -print**
  - find will look for a file called targetfile in any part of the directory tree rooted at **directoryName**.
  - By default, it searches recursively.

find command

- **find directoryName –name targetfile –print**
  - targetfile can include wildcard characters
    - find /home -name "*.txt" -print 2>/dev/null
    - '2>/dev/null' redirects error messages to this special file /**dev/null**, which behaves like a 'black hole'. It consumes whatever we send to it as trash.
    - search all directories under /home for any file whose **name** ending with ".txt" and output any matching files.
    - show full absolute path.
    - single quotes or no quote is also fine for '.txt'.

find command

- It can find files by type
  - -type f for files
  - -type d for directories
- It can find files by permission
  - -perm o=r for all files and directories that can be read by others.
- find . -type f -name file22
  - Looking for a regular file named ***file22*** in the directory tree rooted at the current directory.

find command

- It can also find files by size

  -size

  – find . -size 0c
    - all files of 0 bytes length
  – find / -size +100k
    - all files on whole system larger than 100k
  – find / -size -100k
    - all files on whole system smaller than 100k

find command

- You can also execute commands on the files you find.
- **find . -name "*.txt" -exec wc -l {} ‘;’**
  - Counts the number of lines in every text file inside and the current directory and its subfolders.
  - The {} will be replaced by the name of each file found.
  - the ‘;' ends the -exec clause. ‘’ here is required.
- Using **man find** to get more information

find Vs. grep

- **find differs from grep command**
  - find command to search files in file system by name, size, permission etc.
  - grep to search specific content in files.
  - Different command layout
    - grep –r 'pattern' **directory_name**
    - find **directory_name** –name targetfile –print
  - Find by default traverses folders and all subfolders.

Escape Character \ with echo

- \
  - removes whatever special meaning the next character would have had to the shell.

Quotes with echo command

- Single Quotes
  - Variables names not expand
  - Everything inside is treated literally.
    - NAME=John #here NAME is a shell variable
    - echo '$NAME' → $NAME
    - echo '\\' → \\

Quotes with echo command

- Double Quotes
  - Variable names are expanded.
  - Special meaning of metacharacters are preserved.
    - NAME=John
    - echo “$NAME” → John
    - echo “\\” → \

Quotes with echo command

- Double quotes will remove the special meaning of all characters except the following,
  - $ Parameter Substitution.
  - ` Backquotes
  - \$ Literal Dollar Sign.
  - \` Literal Backquote.
  - \” Embedded Doublequote.
  - \\ Embedded Backslashes.

Command Substitution

- ``
  - Back quotes– called accent characters.
  - the output generated by the command replace the command itself.
    - echo `pwd`
    - Display your current working directory,
      - E.g. /home/ytian/cscd240
    - The output of command pwd replaces the command name itself, which is enclosed in back quotes.

Command Substitution

- ``
  - echo “The current time is `date`.”
    - Output example: “The current time is Wed Dec 25 20:32:55 PST 2013.”
    - Double quotes preserve the special meaning of backquotes.
    - Output of **date** command replaces the corresponding spot in the echo command.
    - compare with **echo ‘The current time is `date`’**

Other Useful Commands

- ln
  - make links between files
  - ln –s ~/cscd240/mylab1.txt mylab_link
    - -s means symbolic links or soft links
    - Create a symbolic link mylab_link, which links to file ~/cscd240/mylab1.txt.
  - if you modify mylab_link, you actual modify ~/cscd240/mylab1.txt.
  - The symbolic link can exist on a different disk partition than the target.
  - directory also can be linked with **symbolic** link.

Other Useful Commands

- uname -a
  - show system and kernel
- mount
    - show mounted filesystems
- whoami
  - show your username.
- export NAME=value
  - set variable Name to value

Other Useful Commands

- **tar –cvf myArch.tar file1 file2 folder1**
  - Create an archive file myArch.tar, which contains two regular files **file1**, **file2** and all contents in **folder1**.

  -c means creating new archive file

  -v means verbose output and listing all files in the archive file.

  -f means writing to the provided archive file name.
- **tar –xvf myArch.tar** to extract contents in archive myArch.tar.

Other Useful Commands

- **zip -r hw1.zip folder1/**
  - Create a package and compress it into zip file
  - -r means traveling directories recursively
    - Everything in **folder1** including files and subdirectories are added to hw1.zip
- **unzip hw1.zip**
  - To decompress and extract all files and folders in hw1.zip.

Summary

- **find differs from grep command**
  - find command to search files in file system by name, size, permission etc.
  - grep to search specific content in files.
  - Different command layout
    - grep –r 'pattern' directory_name
    - find directory_name –name targetfile –print
  - Find by default traverses folders and all subfolders.

Summary

- Quotes with echo command
  - everything inside **single** quote has literal meaning.
  - special metacharacters $ and ` in **double** quotes are preserved.
  - backquotes
    - echo “The current time is `date`.”
- **find directoryName –name targetfile –print**
- ln –s ~/cscd240/mylab1.txt mylab_link