grep command

Computer Science Department

Eastern Washington University

Yun Tian (Tony) Ph.D.

Recall Last Class

- Shell sequence, when input a command, how the shell search many places.
- I/O redirection, >, <, >> operators
- **set -o noclobber** to prevent accidental overwirte.
- **Piping**
  - **ls –l | wc –l**
- **whereis** and **locate** to find a program

Recall Last Class

- **which <command name>**
  - **Search in your PATH**
- **whereis <command name>**
- **locate <pattern to search>**
  - You use locate or whereis to find a command such as you know it is there somewhere but not in your path.

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

Outline for Today

- grep command
  - origin of name
    - Regular Expressions -- Matching Text Patterns
    - Global Regular Expression Print

grep command

- **grep [options] PATTERN [FILE]**
  - locates particular content within files.
  - Searches for PATTERN in all files specified by FILE
    - if no file specified, input from stdin
- E.g.
  - grep CD catalog
    - output lines in the file catalog
  - grep ‘Compact Disc’ catalog

grep command

- grep only matches patterns that appear on a single line.
  - grep -i 'compact disc' catalog
  - if one line in `catalog' ends with the word `compact' and the next begins with `disc', grep will not match either line.

grep command

- Options
  - n
    - prints lines found within filenames and line numbers.
  - i
    - regardless of case of its letters.
    - grep -i 'compact disc' catalog
    - This command outputs lines in the file `catalog' containing any variation of the pattern `compact disc', including `Compact Disc', `COMPACT DISC', and `comPact dIsC'.

grep command

- Options
  - r
    - Search a given directory recursively, searching all subdirectories it contains.
    - *grep -r CD ~/doc*
      - Matches lines containing the word `CD' in all of files in the `~/doc' directory and in all of its subdirectories
    - Very useful when programming large project
      - grep –rn dbConnect --include=*.c   ./

grep command

- Options
  - r
    - Very useful when programming large project
      - grep –rn dbConnect --include=*.c ./
    - Searches for all *.c files containing dbConnect recursively starting in the current directory and all subdirectories.
    - --include allows file name wildcards since FILE is used to specify the current directory.

grep command

- Options
  - v
    - invert the sense of matching, to select non-matching lines.
  - c
    - Suppress the printing of matching lines, and only display the total number of lines that match the query.
  - x
    - The provided pattern has to match a line exactly.

EASTERN WASHINGTON UNIVERSITY
start something **big**

grep command

- Powerful when using with piping
- ls –LR | grep errfile
  - ls –LR will output all files and directories in your current directory and all its subdirectories.
  - Then we look for a file whose name contains ‘errfile’ in the output above.
- who | grep tony
  - see if tony is logged on.
- tail -n8 a_file | grep "boo“

Regular Expression

- A regular expression or "regexp“
  - a text string of special characters that specifies a set of patterns to match.
- Most characters represent themselves.
  - For example, the regexp pattern *1* matches the string `1', and the pattern *bee* matches the string `bee'.

Regular Expression

- Metacharacters
  - Metacharacters that don't represent themselves in a regular expression,
  - but they have a special meaning that is used to build complex patterns. These metacharacters are as follows,

    .   *   [ ] ^ $ \

Regular Expression

- **. (the dot)**
  - Matches any one character, with the exception of the newline character.
  - ‘.wn’ matches wn preceded by a character.
- *** (the asterisk)**
  - Matches the preceding character zero or more times. For example, -* matches `-', `--', `---', `-------'

Regular Expression

- **$**
  - Matches the end of the line. So ‘a$’ matches ‘a' only when it is the last character on a line.
  - what if ‘$a’?
- ^
  - Matches the beginning of the line. So ‘^a’ matches `a' only when it is the first character on a line.
  - what about ‘^$’, ‘^.’, ‘^.*$’ ?

Regular Expression

- [ ]
  - Encloses a *character set*, and matches any member of the set.
    - For example, ***[abc]*** matches either `a', `b', or `c'.
  - The hyphen specifies a range of characters, ordered according to their ASCII value.
    - [0-9] is same as [0123456789];
    - [A-Za-z] matches one uppercase or lowercase letter.

Regular Expression

- [….]
  - grep '[1-5].[aeiou]' myfile
  - match '46a' '1Qu' '2Pe'
- [^ ….]
  - Any character NOT listed in the class
  - grep '[^a-zA-Z]$' myfile
    - match lines that do NOT end with alphabetical letters.

Regular Expression

- ‘\’
  - \ before a metacharacter when you want to specify that literal character.
  - grep ‘\.$’ myfile
    - Search lines that ends with a period.
    - The dot in the pattern is treated literally.
    - grep ‘\$1\.99’ myfile

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

- tail and head
- grep –r ‘pattern’ directory_name
- basic regular expression
  - .   *   [ ] ^ $ \
  - grep ‘[^a-zA-Z]$’ myfile
- Be default, we are using basic regular expression.

Next Class

- Quotes in Shell
- find command