Seneca | SCHOOL OF INFORMATION AND COMMUNICATIONS TECHNOLOGY

Unix Scripting

Week6

Agenda

- Globing shell options
- Extended Globing
- Named Character Classes

Globing shell options

- Pathname Expansion also called globbing, used to find filenames that match a pattern, Using wild characters like : * and ?
  - ls t?.*
- globbing is performed by the shell, not by commands, so globbing may be used with any command
- Using []
  - ls make.[1-3]
  - ls [^abc]
- Using {}
  - touch myfile{1..10}
  - echo {1..10}
  - echo {1..10..2}

Question

- What is the difference of the followings:
  - `ls -l [Ss]*`
  - `ls -l ' [Ss]*'`

- What does the following command do?
  - `grep '[Ff]irst' *.txt`

Globing examples

- `ls tests/?at.js`
  - This will match files such as tests/cat.js, test/Cat.js, test/bat.js etc.
- `ls tests/feature[1-9]/HelloWorld.js`
  - This glob will match files like tests/feature1/HelloWorld.js, test/feature2/HelloWorld.js and so on... upto 9.

Double Asterisk (**)

- Double Asterisk (**) matches zero or more characters across multiple segments. It is used for *globbing files* that are in **nested directories**.

- Example: `ls Tests/**/*.js`
  - Here, the file selecting will be restricted to the Tests directory. The glob will match the files such as Tests/HelloWorld.js, Tests/UI/HelloWorld.js, Tests/UI/Feature1/HelloWorld.js.

null command [colon]

- The ":" command is itself a Bash builtin, and its exit status is true (0).
- This is the shell equivalent of a "NOP" (no op, a do-nothing operation)
  - may be considered a synonym for the shell builtin true
- This command also useful for assigning default value to variables.

null command [colon]

- while :
- do
  - operation-1
  - operation-2
  - …
  - operation-n
- done

- while true
- do
  - operation-1
  - operation-2
  - …
  - operation-n
- done

: colon command for bash

- This command also useful for assigning default value to variables.
  - `RSRC=$1`
  - `LOCAL=$2`
  - `: ${RSRC:="/var/www"}`
  - `: ${LOCAL:="/disk2/backup/remote/hot"}`

shopt

- **shopt** is a builtin command of the Bash shell which can enable or disable options for the current shell session.
- **shopt** [**-o**] [**-p**] [**-q**] [**-s**] [**-u**] [*optname*...]
- https://www.computerhope.com/unix/bash/shopt.htm

Try the following examples:

- nullglob - non-matching globs are removed, instead of preserved
  ```
  echo [0-9]
  shopt -s nullglob
  echo [0-9]
  ```
- failglob - non-matching globs cause an error, command is not executed
  ```
  echo [0-9]
  shopt -s failglob
  echo [0-9]
  ```
- nocaseglob - matches are done ignoring case
  ```
  echo file*5
  shopt -s nocaseglob
  echo file*5
  ```

Extended Globing

- **extended globbing** may be enabled via a shell option: `shopt -s extglob`, but is on by default
- It allow us to add
  - **?(pattern-list) :**Matches zero or one occurrence of the given patterns
  - ***(pattern-list) :** matches zero or more occurrences of the given patterns
  - **+(pattern-list) :** matches one or more occurrences of the given patterns
  - **@(pattern-list) :** matches one of the given patterns
  - **!(pattern-list) :** matches anything except one of the given patterns

Example

- ls pic*.jp?(e)g
- ls pic*(3).*
- ls pic+(3).*
- ls pic*@(jpg|gif)
- ls pic!(*jpg|*gif)
- More example in Bash Extended Globing: https://www.linuxjournal.com/content/bash-extended-globbing

Named Character Classes

- Named character classes are useful, ensuring that collating sequences are correct regardless of the locale
- [:alnum:] - alphanumeric - same as [:alpha:] and [:digit:]
- Can be used with TR
- can be used within regular expressions, including within the "[[...]]" structure (must be enclosed within a second set of square brackets)

tr command in Linux

- **tr** is used to translate characters to different characters
- **tr a A < filename**
  - translate all characters "a" to "A“
- **tr ' ' '\n' < filename**
  - translate all spaces to newline characters
- **tr -d '\n' < filename**
  - delete all newline characters
- tr "[:lower:]" "[:upper:]" < cars
  - What does this do?

Observation: What does this do?

- echo */x*

- ls -l {m*,*est*}

** globing operator

- `**` globing operator matches filenames and directories recursively.
- The *globstar* shell option needs to be set:
  - `shopt -s globstar`
  - This is new shell option in version 4 of Bash.
- Example
  - `for filename in **`
  - `do`
    - `echo "$filename"`
  - `done`

Examples of extended globing

- What extended globbing? https://learnbyexample.github.io/tips/cli-tip-19/
- Formulate a command to search those filenames which are starting with character ‘a’ and has the extension ‘bash’ or ‘sh’
  - ls a*+(.bash|.sh)
- Formulate command to search those files whose names are 5 characters long and the extension is **‘sh’** or the last two characters of the files are **‘st’** and the extension is **‘txt’**.
  - ls -l {?????.sh,*st.txt}

Good to read

- https://tldp.org/LDP/abs/html/globbingref.html
- https://teaching.idallen.com/cst8207/15w/notes/190_glob_patterns.html