Seneca | SCHOOL OF INFORMATION AND COMMUNICATIONS TECHNOLOGY

Unix Scripting

Lecturer: Shahdad Shariatmadari

June 2020

What we have learned…

- Introduction to Shell Scripting
  - Categories of variables
  - Conditional Statements
  - Loops
- stdin, stdout, stderr Redirection and piping
- File descriptor
- Filtering
  - Simple filter commands: head, tail, cut, sort, wc
  - grep utility
- Multiple commands : () vs {}
- Sed, Awk, Grep

Agenda

- Globing shell options
- Extended Globing
- Named Character Classes

Globing shell options

- Using wild characters like : * and ?
  - ls t?.*
- Using []
  - ls make.[1-3]
  - ls [^abc]
- Using {}
  - touch myfile{1..10}
  - echo {1..10}
  - echo {1..10..2}

Globing shell options : Activity

- Create a script, myscript.sh, add the followings:
  - echo $10
  - echo ${10}
- Run “ myscript.sh p1 p2 p3 p4 p5 p6 p7 p8 p9 p10
- What happen?
- Run myscript.sh pABC p2 p3 p4 p5 p6 p7 p8 p9 p10
- What happen?

Question:

- What does the following command do?
- echo *

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) :**a pattern-list is a list of items separated by a vertical bar
  - ***(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)

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

Example

- tr "[:lower:]" "[:upper:]" < cars

Activity

- Explain how the following works
- ```
  echo $1 | grep "^[[:digit:]]*$"
  >/dev/null || { echo "First
  argument must be numeric" >&2;
  exit 2; }
  ```
- ```
  ls pic[[:digit:]].*
  [[ $1 = *[^[:digit:]]* ]] && {
  echo "First argument must be
  numeric" >&2; exit 2; } || exit 4
  ```