Shell Scripting 1

Week 1 - Thursday

Introduction to Shell

- Shell
  - Software that provides a user interface to access the service of the OS kernel
  - What's the kernel?
- Various shell implementations on Unix-like systems
  - bash, zsh, csh
  - Check your default shell with "echo $SHELL"
  - Change default shell with "chsh"
- Shell script
  - Including
    - Commands
    - Variables, control-flow constructs, comments, etc.
  - Like a C program?
  - Shell interprets & executes the script

From Command to Script

- Basic string matching -> Advanced string matching: regular expression
- Basic commands -> Advanced commands
- How to combine different commands: pipelines & redirection
- Programming with commands: shell scripting

Focus: Deal with text

Matching a String with Wildcards

Pattern	Match	Not match
a*	ab, abc, abcde	bc
a?	ab, ac, ad	abc
a[bc]	ab, ac	ad
a[b-d]	ab, ac, ad	ae

- Match one/more arbitrary character: *, ?
- Match specified characters/ranges: []
- Are these tools powerful enough?

Example: Checking C++ Variable Names

- A C++ variable name
  - Can consist of
    - Letters
    - Digits
    - Underscores
  - Should not start with an digit
- How to check whether a given string is a legit C++ variable name?

Regular Expression: A Powerful Tool to Deal with Strings

- RE: A sequence of characters that forms a search pattern, mainly used for string matching
- Metacharacters to improve the flexibility of patterns
- Backreferences to extract substring
- Widely used by different languages
- NOTE: not all the commands support regular expression (hint: check the manual)

RE: Metacharacters

Character	Meaning
^	Beginning of a line/string
$	End of a line/string
.	Any single character
[]	A range of character
\	Turn on/off the special meaning of a character

RE: Pre-defined Patterns

Class	Meaning	Class	Meaning
[:alnum:]	Alphanumeric characters	[:lower:]	Lowercase characters
[:alpha:]	Alphabetic characters	[:print:]	Printable characters
[:blank:]	Space and tab characters	[:punct:]	Punctuation characters
[:cntrl:]	Control characters	[:space:]	Whitespace characters
[:digit:]	Numeric characters	[:upper:]	Uppercase characters
[:graph:]	Nonspace characters	[:xdigit:]	Hexadecimal digits

NOTE: use [[:alpha:]]

RE: Quatifier

Character	Meaning
?	There is 0 or 1 of the preceding element
*	There is 0 or more of the preceding element
+	There is 1 or more of the preceding element
\{n\}	There is n of the preceding element
\{n,m\}	There is x of the preceding element, where n<=x<=m

Examples

- What do the following patterns stand for?
  - [a-zA-Z].*
  - [abc]\{10\}
  - [[:alpha:]]
  - ABC, ^ABC, ^ABC$
- How to check a legit C++ variable name?

Backreferences

- Match whatever an earlier part of the regular expression matched
  - Enclose a subexpression with \(and \).
  - There may be up to 9 enclosed subexpressions and may be nested
  - Use \digit, where digit is a number between 1 and 9, in a later part of the same pattern.

Pattern	**Matches**
\(ab\)\(cd\)[def]*\2\1	abcdcdab, abcdeeecdab, abcdddeeffcdab, ...
\(why\).*\1	A line with two occurrences of why

Example: Merge & Sort Data

- We have two files “grade1.txt” & “grade2.txt”
  - Each file have one or more lines
  - Each line only has an integer
- We want to merge the records in the two files and sort the grades, then write the sorted results into “sorted_grade.txt”
- How to solve these problems with C/C++?
- How long it will take to program?

Advanced Commands

Focus: Deal with text

- Print a string: echo
- Concatenate and print files: cat
- Extract top/bottom of files: head, tail
- Word count: wc
- Sort lines of text files: sort
- File pattern searcher: grep
- Stream editor: sed
- Pattern-directed scanning and processing language: awk

grep

- Search input files and select lines that match one or more pattern
  - How to express a pattern? RE
  - Patterns may consist of one or more lines
- Usage: “grep pattern input file“
- Try it out
  - Compose a text file, each line containing one string
  - Use grep and RE to extract the ligit C++ variable names in the file

sed

- With grep and RE, you can extract (read) the required text
- But how to edit the extracted text?
- Stream editor: sed
  - Read – Match – Write
  - E.g., replacing text
    - sed ‘s/matchPattern/replaceText/’
    - sed 's/:.*//' /etc/passwd -> remove everything after colon
- More options available, check the manual!

awk

- Scanning: scan the input file and find the lines meeting the given requirements
- Processing: you can write c-like code to process the data
- Super powerful tool for data processing
  - E.g., sum the data that meets given requirements

Redirection & Pipelines: Prerequisite

- Standard stream
  - Pre-connected input/output channels between a program and its environment
  - Three basic streams: input, output, error
    - C: stdin, stdout, stderr
    - C++: cin, cout, cerr
  - What happened when running “ls” in a terminal?

Redirection & Pipelines: The Motivations

- How can I write the content printed on the screen into a file (for my submission)?
- How can I take the output of one program as the input of another program?

Redirection & Pipelines

- IO redirection
  - Redirect the standard stream of a program to a file
    - Write the output to a file: >
      - ls > list.txt
    - Read input from a file: <
- Pipelines
  - Take the output of a program as the input of another program
    - ls | less
- Can you solve the merge/sort problem with shell commands? How long it takes?