Software Construction Laboratory

Week 3

Lab 3

Mushi Zhou

Winter 2017

Prof. Paul Eggert

UCLA

Some Help For Assignment 2

- Assignment 2 is due Tuesday this week.
- Assignment 3 is still due this Saturday Midnight.

Step Approach to Lab2

- Get the English dictionary “words”
- Spell check the assignment page
- Build a script “buildwords” that executes the rules mentioned in the lab
- Run script against “English to Hawaiian” page to form Hawaiian dictionary “hwords” after sorting the output
- Verify Hawaiian spell checker by running against itself
- Spell check the assignment page with “hwords” after ensuring all lower cases
- Log your findings

Some help for HW2

- Find duplicate files in a given directory
- Sort them and only keep one of each duplicates (Prefer “.”)
- Use hard link to replace others (Read what’s hard link)
- Only immediate files
- Only regular files
- Special character file names (space, *, -)
- Report Errors

Step Approaches

1. find all regular files and list the names of them
2. **run through the list and generate a list of group of duplicates**
3. Find files start with "." in each group
4. Sort "." files (or all files if there are not any ".") in each duplicate groups
5. Replace duplicates with hard links to the first in each group

Useful Tips

- Find –type
- Sed
- Find –maxdepth
- Diff/cmp
- Ln {s} {l}
- Sort is case sensitive by default
- Need to use regex (wildcards) to match names starting with “.”
- Pay attention to names with special characters (need to escape them)
- Nested loop

Week 3 Outline

- Very Basic introduction to Python
- Very Basic introduction to Java
- Make (Part 2)
- Automake and Autoconf and Cmake (Part 2)

Python Introduction

- High-Level
- General-purpose
- Interpreted*   .py (script)
- Dynamic
- Developed in late 1980s

Motivations

- Supports multiple programming paradigms: Object-oriented, imperative, functional, or procedural
- High readability
- Less code
- Can handle both small and large scaled programs
- Dynamic types
- Automatic memory management
- Large comprehensive libraries
- Free & open source licensed with Python Software foundation

Motivation

- Used to show graphical visualizations/animations
- Used to build small projects/production/gaming software
- Used as scripts to process data in Medical Science, Biology, Accountings

Example of Python in Script

```
#!/usr/bin/python
import getopt, random, sys

class randline(file):
   def __init__(self, filename):
      f = file (filename, 'r')
      self.lines = f.readlines ()
      f.close ()

   def chooseline(self):
      choice = random.randrange (len (self.lines))
      return self.lines[choice]

   def usage (e):
      sys.stderr.write ('randline.py: %s\n' % e)
      sys.stderr.write (\
         '''
         Usage: randline.py [OPTION]... FILE
         Output a line selected randomly from FILE.
         Options: -n LINES Output LINES lines (default 1).
         '''\
         )
      sys.exit (1)
   …………
```

Tells the shell to use python interpreter

Import statements, similar to include statements in C/C++

The beginning of the class statement: randline
The constructor
    Open the file
    Reads the file into array of strings called lines
    Close the file

Self-defined function
Randomly select a number between 0 and the size of lines
Returns the line corresponding the randomly selected number

The beginning of a function: usage
    Write the error
    Write instructions on how to use randline.py.

It follows similar formatting syntax as printf in C
Note the use of the “triple quote”
Note the use of the “line continuation character ” \
You don’t normally need \ when you’re using triple quotes
Exit with error

Indentation matters!

Python 2 vs Python 3

- Python 2.7x -> Legacy
- Phthon3 -> new and future, released in 2008
- Backward incompatible
- Many new futures on Python3
- Many 3-rd party libraries on Python2
- Link for a Python tutorial: https://www.tutorialspoint.com/python/

Introduction to Java

- General-purpose
- Concurrent
- Class-based
- Object-oriented
- Static Typing
- Complied language but with some compromise
- “write once, run anywhere”

Motivations for Java

- Most popular language in use (commercially)
- Particularly for client-server web applications
- Released in 1995
- Much fewer low-level facilities than C/C++
- Free and support majorly by Oracle
- Open source under GNU General Public License but not re-distributable
- Current Version 8
- Require the use of JDK (Java Development Kit)

Comparison Between Python and Java

- Dynamic vs Static Typing
- Braces vs Indentation
- Speed vs Portability
- Python is considered easier for new programmer to start with
- Java very popular and considered essential for app developmens for android

Something to Think About

- Very large programs -> Java could be faster as it is compiled (So everything is relative)
- Small Simple Programs -> python could be faster
- Java program take more time to develop as you unusually need to write multiple files for one task
- Java & Python has different emphases in terms of purpose & development