CS 35L

Week 3

TA: Tomer Weiss
Jan-21-2016

goo.gl/0hhFJC

Slides

Modifying and rewriting software

Lab

web.cs.ucla.edu/classes/winter16/cs35L/assign/assign3.html

Announcements

- 2 day extension to assignment 3 deadline
- Submit by Sunday 2016-01-22
- Follow requirements on: piazza.com/class/ij094jwyhvp56n

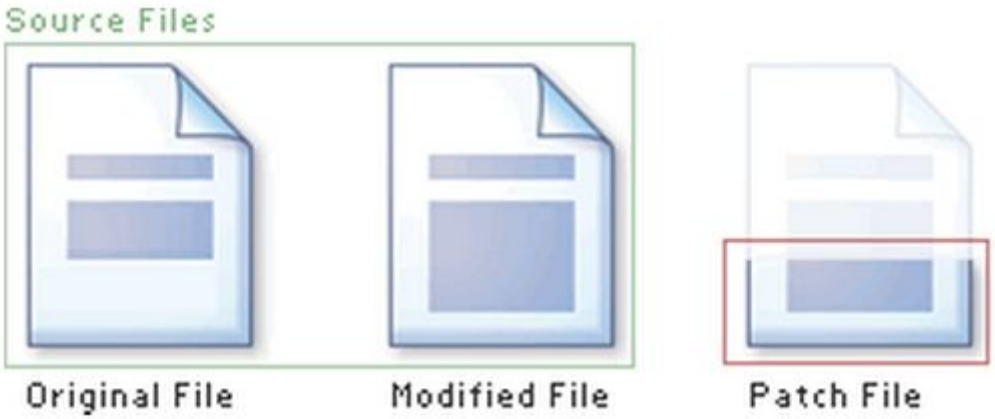
Applying the Patch

- Read the patch bug report
  - lists.gnu.org/archive/html/bug-coreutils/2009-09/msg00410.html
- Understand what part of the code is being fixed

Patching

- A patch is a piece of software designed to fix problems with or update a computer program.
- It's a `diff` file that includes the changes made to a file
- A person who has the original (buggy) file can use the `patch` command with the diff file to add the changes to their original file

Applying a Patch



Applying the Patch

```
diff --git a/src/ls.c b/src/ls.c
    index 1bb6873..4531b94 100644
    --- a/src/ls.c
    +++ b/src/ls.c
    @@ -2014,7 +2014,6 @@ decode_switches (int argc, char **argv)
                  break;

                case long_iso_time_style:
    -           case_long_iso_time_style:
                  long_time_format[0] = long_time_format[1] = "%Y-%m-%d %H:%M";
                  break;

    @@ -2030,13 +2029,8 @@ decode_switches (int argc, char **argv)
                       formats.  If not, fall back on long-iso format.  */
                    int i;
                    for (i = 0; i < 2; i++)
    -                 {
    -                   char const *locale_format =
    -                     dcgettext (NULL, long_time_format[i], LC_TIME);
    -                   if (locale_format == long_time_format[i])
    -                     goto case_long_iso_time_style;
    -                   long_time_format[i] = locale_format;
    -                 }
    +                 long_time_format[i] =
    +                   dcgettext (NULL, long_time_format[i], LC_TIME);
                  }
                }
              /* Note we leave %5b etc. alone so user widths/flags are honored.  */
```

```
--- /path/to/original ''timestamp''
+++ /path/to/new      ''timestamp''
@@ -1,3 +1,9 @@
+This is an important
+notice! It should
+therefore be located at
+the beginning of this
+document!
+
 This part of the
 document has stayed the
 same from version to
@@ -5,16 +11,10 @@
 be shown if it doesn't
 change.  Otherwise, that
 would not be helping to
-compress the size of the
-changes.
-
-This paragraph contains
-text that is outdated.
-It will be deleted in the
-near future.
+compress anything.

 It is important to spell
-check this dokument. On
+check this document. On
 the other hand, a
 misspelled word isn't
 the end of the world.
```

diff Unified Format

- --- path/to/original_file
- +++ path/to/modified_file

- @@ -l,s +l,s @@
  - @@: beginning and end of a hunk
  - l: beginning line number
  - s: number of lines the change hunk applies to for each file
  - A line with a:
    - - sign was deleted from the original
    - + sign was added in the new file
    - ‘ ‘ stayed the same

Patching

- `cd` into coreutils-7.6
- vim or emacs patch_file: copy and paste the patch content
  - `patch [options] [originalfile [patchfile]]`
  - `patch -p`**`num`** `<patch_file`
  - *`man patch`* to find out what p**num** does and how to use it
- `cd` into the coreutils-7.6 directory and type `make` to rebuild patched `ls.c`
- More patch command examples - link

Homework

What is Python?

- Not just a scripting language
- Object-Oriented language
  - Classes
  - Member functions
- Compiled and interpreted
  - Python code is compiled to bytecode
  - Bytecode interpreted by Python interpreter
- Not as fast as C but easy to learn, read and use

Optparse Library

- Powerful library for parsing command-line options
  - **Argument:**
    - String entered on the command line and passed in to the script
    - Elements of sys.argv[1:] (sys.argv[0] is the name of the program being executed)
  - **Option:**
    - An argument that supplies extra information to customize the execution of a program
  - **Option Argument:**
    - An argument that follows an option and is closely associated with it. It is consumed from the argument list when the option is

Python List

- Common data structure in Python
- A python list is like a C array but much more:
  - **Dynamic**: expands as new items are added
  - **Heterogeneous:** can hold objects of different types
- How to access elements?
  - List_name[index]

Example

- >>> t = [123, 3.0, ‘hello!’]
- >>> print t[0]
  - 123
- >>> print t[1]
  - 3.0
- >>> print t[2]
  - hello!

List Operations

- >>> list1 = [1, 2, 3, 4]
- >>> list2 = [5, 6, 7, 8]
- Adding an item to a list:
  - list1.append(5)
  - **Output: [1, 2, 3, 4, 5]**
- Merging lists:
- >>> merged_list = list1 + list2
- >>> print merged_list
  - **Output: [1, 2, 3, 4, 5, 5, 6, 7, 8]**

for loops

```
list = [‘Mary’, ‘had’, ‘a’, ‘little’, ‘lamb’]
```

```
for item in list:
    print item
```

Result:

Mary
had
a
little
lamb

```
for i in range(len(list)):
    print i
```

Result:

0
1
2
3
4

Indentation

- Python has **no braces** or keywords for code blocks
  - C delimiter: {}
  - bash delimiter:
    - then…else…fi (if statements)
    - do…done (while, for loops)
- **Indentation** makes all the difference
  - Tabs change code's meaning!!

Running Python scripts

- Download *randline.py* from assignment website
- Make sure it has executable permission:

  chmod +x randline.py
- Run it, for example

  ./randline.py –n 4 filename

  n: is an option indicating the number of lines to write

  4: is an argument to n (you can use any integer number)

  Filename: is a program argument

Example run

- I downloaded text version of ‘Alice in Wonderland’ for testing the script.
  - Available free from:
    - www.gutenberg.org/ebooks/19033
  - Ran the script on file

```
> ./randline.py  -n 4 alice_in_wonderland.txt
Gutenberg-tm electronic work under this agreement, disclaim all
[Illustration]
with the permission of the copyright holder, your use and distribution
that she had put on one of the Rabbit's little white kid-gloves while
```

What does the script do?

Homework 3 - Overview

- randline.py script
  - Input: a file and a number *n*
  - Output: *n* random lines from *file*
  - Get familiar with language + understand what code does
  - Answer some questions about script
- Implement the comm command in python

Python Walk-Through

```
#!/usr/bin/python

import random, sys
from optparse import OptionParser

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

     def chooseline(self):
          return random.choice(self.
lines)


def main():
     version_msg = "%prog 2.0"
     usage_msg = """%prog [OPTION]...
FILE Output randomly selected lines
from FILE."""
```

Tells the shell which interpreter to use

Import statements, similar to include statements
Import OptionParser class from optparse module

The beginning of the class statement: randline
The constructor
Creates a file handle
Reads the file into a list of strings called lines
Close the file

The beginning of a function belonging to randline
Randomly select a number between 0 and the size of lines minus 1 and returns the line corresponding to the randomly selected number
The beginning of main function
version message
usage message

Python Walk-Through

```
parser = OptionParser(version=version_msg,
      usage=usage_msg) parser.add_option("-n", "--
numlines",            action="store", dest="
numlines",       default=1, help="output NUMLINES
      lines (default 1)")

options, args = parser.parse_args(sys.argv[1:])

try:
    numlines = int(options.numlines)
except:
    parser.error("invalid NUMLINES: {0}".
      format(options.numlines))
if numlines < 0:
    parser.error("negative count: {0}".
format(numlines))
if len(args) != 1:
    parser.error("wrong number of operands")
input_file = args[0]
try:
    generator = randline(input_file)
    for index in range(numlines):
        sys.stdout.write(generator.chooseline())
except IOError as (errno, strerror):
    parser.error("I/O error({0}): {1}". format
(errno, strerror))

if __name__ == "__main__":
    main()
```

Creates OptionParser instance

Start defining options, action “store” tells optparse to take next argument and store to the right destination which is “numlines”.
Set the default value of “numlines” to 1 and help message.
options: an object containing all option args
args: list of positional args leftover after parsing options
Try block
  get numline from options and convert to integer
Exception handling
  error message if numlines is not integer type, replace {0 } w/ input
If numlines is negative
  error message
If length of args is not 1 (no file name or more than one file name)
  error message
Assign the first and only argument to variable input_file
Try block
  instantiate randline object with parameter input_file
  for loop, iterate from 0 to numlines – 1
    print the randomly chosen line
Exception handling
  error message in the format of “I/O error (errno):strerror

In order to make the Python file a standalone program

Comm.py

- Support all options for comm
  - -1, -2, -3 and combinations
  - Extra option –u for comparing unsorted files
- Support all type of arguments
  - File names and – for stdin
- Assume C locale for sorting purposes
- Change usage message to describe script behavior
- Port comm.py to Python 3
- `man comm` or link for more details

Homework 3 Hints

- Sample skeleton python script available here: ccle.ucla.edu/mod/forum/discuss.php?d=299519

- Read first 9 chapters here: docs.python.org/3.5/tutorial/

- The comm options -123 are Boolean
  - Which action should you use?
- Q4: Python 3 vs. Python 2
  - Look up “automatic tuple unpacking”
- Use `python` in shell for Python 2
- Use `python3` in shell for Python 3

```
> which python3
/usr/local/cs/bin/python3
```

Assignment 10

- 5 minute presentation with slides

- Brief Research report
- Please sign-up here

- Link to assignment 10

- Signups will be **first come, first serve** basis. For reference on presentation, grading, please refer to this rubric.

Lab

web.cs.ucla.edu/classes/winter16/cs35L/assign/assign3.html