Modifying Programs

Week 3

Laboratory

Getting Set up

- Download coreutils-7.6 to your home directory
- Download the patch to your home directory
- Untar and Unzip it
  - gunzip coreutils-7.6.tar.gz
  - tar –xf coreutils-7.6.tar
- You now have the files you need in a usable form

Compiling

- Go into coreutils-7.6 directory. This is what you just unzipped.
- Run the configure script so that when everything is done you'll have a makefile for coreutils
- Compile it: make
- **Do not install it!**

Applying the Patch

•Download 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.  */
--
```

Apply the Patch

- Just use an editor to do it.
- Recompile it: make
- Try it out! (use `find` to locate the ls executable)

Homework

Running Python scripts

- Make sure it has executable permission: chmod +x randline.py
- Run it
  - ./randline.py –n 2 filename
  - n: is an option indicating the number of lines to write
  - 2: is an argument to n (you can use any number)
  - Filename: is a program argument

Python Walk-Through

```
#!/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 which interpreter to use

Import statements, similar to include statements

The beginning of the class statement: randline
- The constructor
  - Creates a file handle
  - Reads the file into array of strings called lines
  - Close the file
- The beginning of a function belonging to randline: chooseline
  - 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, but we threw it in there for kicks***

Exit with error

Note the use of indentation for encapsulation.

Python Walk-Through

```
if __name__ == '__main__':
    opt = {}
    opt['n'] = 1

    try:
        opts, args = getopt.getopt (sys.argv[1:], 'n:')
    for option, value in opts:
            v = int (value)
             if v < 0:
                raise ValueError, 'negative count: %d' % v
             opt[option[1]] = v
    except (getopt.error, ValueError), e:
        usage (e)

    if len (args) != 1:
        usage ('wrong number of operands')
    input_file = args[0]
    line_count = opt['n']

    try:
        generator = randline (input_file)
        for i in range (line_count):
            sys.stdout.write (generator.chooseline ())
    except IOError, e:
        sys.stderr.write ('randline.py: %s\n' % e)
        sys.exit (1)
```

This tests whether or not this script is being imported or run. When its name is "__main__" its being run.

Initialize an array

Start an exception catching block
- Get option and program args. If an option requires an arg, include ":"
- For each option-value pair
  - Get the value
  - If the value is less than 0
    - Throw/Raise an exception
  - Store the value in opt array at option[1]

Handle the exception ValueError
- Print the usage info

If the number of program arguments is not 1
- Print the usage info

Store the name of the input_file

Store the number of lines from the opt array

Start an exception catch block
- Create a randline object
- Range returns an array [0,1,…,line_count].
  - Write out the line

Handle the exception IOError
- Write error
- Exit with an error