Halit Özsoy
2016400141
05.12.2017
CMPE 230 – HOMEWORK 2

REPORT / DOCUMENTATION FOR HOMEWORK 1

Problem:

Develop a python program that given a directory, it traverses the subdirectories and create a makefile according to the .c and .h files found.
"./generatemakefile.py [root_c_project_path]" should create a makefile.

My Solutions & Challenges I Faced

Firstly, I read the problem and focused on solving how to give warnings & errors.

I made a program that traverses subdirectories of a folder and notes down .h and .c files found.
Then, checks the .c files for "#include" statements by regex, notes down all .h files.
Then, creates two set, one for every .h file found by traversing and one for all .h files found in include statements.
Compares the two sets, and gives error / warning if they are not equal.

Then, I saw the testcases, and thought I understood the problem very wrong, and it was a lot harder than I thought.

After a week, I started programming again from scratch. This time, I wrote similar things to my previous program in a better formatted way, and added another function. This time, the program checks every .h files for function definitions and .c files for declarations / calls. Additionally, the program also checks for #define statements in .h and uses of those in .c files. (but does not check for the cases where the define statement has parameters, checks only cases like #define PI 3.14)

That way, I created a program that mostly achieves what I've been asked, other than the parametered define case, it would successfully give warnings / errors. (I thought there wouldn't be a case with parametered defines.)

Anyways, next day I found out that what I understood initially was right, and the test cases were wrong. So, I changed a few lines and reimplemented what I was asked. Also, the instructor told us to check for the case when there are more than one file with the same name, and give errors in such case, and I added that function. That was the end of me implementing warnings / errors.

Then, the next day I started implementing the generatemakefile function part. (Today - 5th Dec)

That part was relatively easy, since I have already implemented all other functions I needed.
First, I created an INCDIR variable in makefiles consisting of all directories I have traversed, so when I used it in gcc, any include file would be found without problem.
(For example INCDIR = -I"." -I"./src" -I"./headers" -I"./headers/headers_2")
(Then, gcc -c $(INCDIR) ./main.c)

Then, I created the generator of the rest of the makefile. (easier parts)

Note, I did not delete what I've added for the wrong test cases, but added Deprecated before the warning/errors. So; a Deprecated Warning will be printed if a file is included but not used, and a Deprecated Error will be printed by my program if a file is used but not included.

Halit Özsoy
2016400141

05.12.2017
CMPE 230 – HOMEWORK 2

What I Learned

I have gained some practice on python, and learned how to create a makefile in more detail. Read and learned some parts of the GNU make manual. Realized that there are a lot more than what's commonly used is possible in makefiles. I enjoyed trying to solve the wrong test case's warning/error challenge. That felt like I was implementing a tiny portion of a compiler.

Documentation

```
#!/usr/bin/python

import os
import sys
import re

## DEBUG - TEST HELPER STARTS
exBase = 'cmpe230fall2017hw2_Testcases/tc2/'
#prints the func name, and result with the given args
def dumper(func, *args):
## DEBUG - TEST HELPER ENDS

baseDir = './'
#baseDir = sys.argv[1]
progName = 'prog.exe'
#progName = sys.argv[2]

sourceFiles = []
#each file is in format (abc.c) => {'fileName':'abc', 'filePath':'./', 'includes': ['dom.h','pat.h']}
headerFiles = []
#each file is in format (abc.h) => {'fileName':'abc', 'filePath':'./', 'methods': ['solve','getName']}
directories = []
#all subdirectories and '.'

#returns all method names declared in a header file
def getMethods(filePath):
#dumper(getMethods, exBase + 'addTwoInt.h')

#returns the fileName from a filePath
def reducePath(filePath):
#dumper(reducePath, 'abc/def/inc.h')

#returns all file includes (except system includes) in a source file
def getIncludes(filePath):
#dumper(getIncludes, exBase + 'main2.c')

#returns all file defines in a header file
def getDefines(filePath):
#dumper(getDefines, exBase + '../tmp.h')

#splits a fileName to [fileBase, fileExt], and if there are no extension/'.' in the name, returns [fileName, 'err']
def splitExtension(fileName):
#dumper(splitExtension, 'abc.def.ghi')

#traverses all files in baseDir, identifies header files, source files and directories
def traverseFiles():
        # sets directories, headerFiles, sourceFiles lists

#returns true if a method is either called or defined in the source file
def isHeaderUsed(sourcePath, methods, defines):
#dumper(isHeaderUsed, exBase + 'main2.c',  ['addTwoInt'])
#dumper(isHeaderUsed, exBase + 'addTwoInt.c',  ['addTwoInt'])

#checks errors and warnings, returns false on error and true on success/warning
def checkErrorsAndWarnings():

#generates the make file
def generateMakeFile():

#checks system arguments, calls traverseFiles, and checkErrorsAndWarnings()
#calls generateMakeFile after validating that checkErrorsAndWarnings() resulted in no error
def main():
        traverseFiles();
        if checkErrorsAndWarnings():
                generateMakeFile()
# call main - start the program
main()
```

Halit Özsoy
2016400141

TESTS

A. Tests on New Test Cases

```
corupta@compta:~/Desktop/CMPE 230/HW/HW2$ for i in {1..5}; do echo "Tescase $i"; ./my_solution/generatemakefile.py cm
pe230fall2017hw2_Testcases_up/tc$i/ tc$i.exe; cd cmpe230fall2017hw2_Testcases_up/tc$i; make; cd ../..; echo ''; done
Tescase 1
Makefile is generated successfully!
gcc -c -I"." -I"./headers1" -I"./src1" ./main1.c
gcc -c -I"." -I"./headers1" -I"./src1" ./src1/multTwoInt.c
gcc -c -I"." -I"./headers1" -I"./src1" ./src1/addTwoInt.c
gcc main1.o multTwoInt.o addTwoInt.o -o tc1.exe
Makefile finished successfully. Created program: tc1.exe

Tescase 2
Makefile is generated successfully!
gcc -c -I"." -I"./pat pat" -I"./dom" -I"./bom" ./main2.c
gcc -c -I"." -I"./pat pat" -I"./dom" -I"./bom" ./addTwoInt.c
gcc -c -I"." -I"./pat pat" -I"./dom" -I"./bom" ./dom/multTwoInt.c
gcc main2.o addTwoInt.o multTwoInt.o -o tc2.exe
Makefile finished successfully. Created program: tc2.exe

Tescase 3
Error! Below files are included in main3.c but was not found:
['multTwoInt.h']
make: *** No targets specified and no makefile found.  Stop.

Tescase 4
Warning! Below header files are found but not included in any source file.
['unusedFunc.h']
Makefile is generated successfully!
gcc -c -I"." -I"./headers4" -I"./src4" ./main4.c
gcc -c -I"." -I"./headers4" -I"./src4" ./src4/multTwoInt.c
gcc -c -I"." -I"./headers4" -I"./src4" ./src4/addTwoInt.c
gcc main4.o multTwoInt.o addTwoInt.o -o tc4.exe
Makefile finished successfully. Created program: tc4.exe

Tescase 5
Makefile is generated successfully!
gcc -c -I"." -I"./headers5" -I"./headers5/headers5_2" -I"./src5" ./main5.c
gcc -c -I"." -I"./headers5" -I"./headers5/headers5_2" -I"./src5" ./src5/multTwoInt.c
gcc -c -I"." -I"./headers5" -I"./headers5/headers5_2" -I"./src5" ./src5/addTwoInt.c
gcc main5.o multTwoInt.o addTwoInt.o -o tc5.exe
Makefile finished successfully. Created program: tc5.exe
```

B. Tests on Old Test Cases

```
corupta@compta:~/Desktop/CMPE 230/HW/HW2$ for i in {1..5}; do echo "Tescase $i"; ./my_solution/generatemakefile.py cm
pe230fall2017hw2_Testcases/tc$i/ tc$i.exe; cd cmpe230fall2017hw2_Testcases/tc$i; make; cd ../..; echo ''; done
Tescase 1
Makefile is generated successfully!
gcc -c -I"." -I"./headers1" -I"./src1" ./main1.c
gcc -c -I"." -I"./headers1" -I"./src1" ./src1/multTwoInt.c
gcc -c -I"." -I"./headers1" -I"./src1" ./src1/addTwoInt.c
gcc main1.o multTwoInt.o addTwoInt.o -o tc1.exe
Makefile finished successfully. Created program: tc1.exe

Tescase 2
Makefile is generated successfully!
gcc -c -I"." ./main2.c
gcc -c -I"." ./multTwoInt.c
gcc -c -I"." ./addTwoInt.c
gcc main2.o multTwoInt.o addTwoInt.o -o tc2.exe
Makefile finished successfully. Created program: tc2.exe

Tescase 3
Deprecated Error!  The header file addTwoInt.h was used, but not included in the file, main3.c
Makefile is generated successfully!
gcc -c -I"." -I"./headers3" -I"./src3" ./main3.c
./main3.c: In function ‘main’:
./main3.c:7:17: warning: implicit declaration of function ‘addTwoInt’ [-Wimplicit-function-declaration]
  printf("%d\n", addTwoInt(5, 8));
                 ^
gcc -c -I"." -I"./headers3" -I"./src3" ./src3/multTwoInt.c
gcc -c -I"." -I"./headers3" -I"./src3" ./src3/addTwoInt.c
gcc main3.o multTwoInt.o addTwoInt.o -o tc3.exe
Makefile finished successfully. Created program: tc3.exe

Tescase 4
Deprecated Warning! The header file unusedFunc.h was included, but not used in the file, main4.c
Makefile is generated successfully!
gcc -c -I"." -I"./headers4" -I"./src4" ./main4.c
gcc -c -I"." -I"./headers4" -I"./src4" ./src4/unusedFunc.c
gcc -c -I"." -I"./headers4" -I"./src4" ./src4/multTwoInt.c
gcc -c -I"." -I"./headers4" -I"./src4" ./src4/addTwoInt.c
gcc main4.o unusedFunc.o multTwoInt.o addTwoInt.o -o tc4.exe
Makefile finished successfully. Created program: tc4.exe

Tescase 5
Makefile is generated successfully!
gcc -c -I"." -I"./headers5" -I"./headers5/headers5_2" -I"./src5" ./main5.c
gcc -c -I"." -I"./headers5" -I"./headers5/headers5_2" -I"./src5" ./src5/multTwoInt.c
gcc -c -I"." -I"./headers5" -I"./headers5/headers5_2" -I"./src5" ./src5/addTwoInt.c
gcc main5.o multTwoInt.o addTwoInt.o -o tc5.exe
Makefile finished successfully. Created program: tc5.exe
```