Author: Zhixin Liu
UO ID: 951452405
Date: 6/3/2016

# Project_Extra Report

## *fileCrawler.c*

### - *State:*

- successfully compiles
- compiles with no warnings
- no memory leak happens
- work correctly in multiple tests

### - *Notice:*

- those files tsiterator.h, tsiterator.c, tslinkedlist.h, tslinkedlist.c, tstreeset.h, tstreeset.c are based on the thread-safe stack part in the course slices of Lecture06
- fileCrawler.c revises the function processDirectory() and applyRe() which are provided in fileCrawlerSingle.c, for the revised version, processDirectory() is called in applyRe()
- I tested several times by typing “for i in {1..1000};do ./fileCrawler '*.[ch]' files | diff - P3.out; done”, etc, no output printout
- For my test on time consuming, the multi-threading version fileCrawler.c improve about 0.03s when testing “./fileCrawler '*' /usr” comparing to “./fileCrawlerSingle '*' /usr”.