ISE 302 – Operating Systems
2014 – 2015, Homework 1
10.10.2014

Introduction

You will code a multi-processing and multi-threading program that calculates course grade averages. You need to create 3 processes for the project, midterm and final grades. Then, these processes should create 1 thread for each course and print corresponding average.

Program Input

Program input is a folder that contains text documents where each document represents a course. Each line in a document is the grades of a student. First grade is for the project, next one is for the midterm exam and the last one is for the final exam. These grades are separated by space characters.

A sample folder is provided. Please note that, there may be any number of course documents in the folder and each course document may have a different number of lines (students). However, number of grades is fixed (3). To evaluate your programs, another test folder will be used.

Processes and Threads

Your program should have the following functionality for the processes and threads.

- **Master Process:** Master process should receive the input folder path as the argument. Then, it should create 3 slave processes for the 3 types of grades.
- **Slave Processes:** Each slave process should read the folder contents and create n threads for the n files in the folder.
- **Worker Threads:** Each worker thread is responsible for reading the contents of 1 file and printing the average of 1 type of grade. For instance, if there are 5 courses, slave process for the midterm exams should create 5 threads and each thread should print the midterm average for only 1 course.

Program Output

A sample program output is given below. Please note that, the order of lines may be different at each run.

```
Master: Start
Slave 1: Project averages
Thread 1.1: Project average for ISE302.txt: 50
Thread 1.2: Project average for ISE101.txt: 60
Slave 1: Done
Slave 2: Midterm averages
Thread 2.1: Midterm average for ISE302.txt: 40
Thread 2.2: Midterm average for ISE101.txt: 50
Slave 2: Done
Slave 3: Final averages
Thread 3.1: Final average for ISE302.txt: 30
Thread 3.2: Final average for ISE101.txt: 40
Slave 3: Done
Master: Finish
```

Appendix

You may make use of the following code examples for File I/O operations in C and C++. First one reads a text file line by line and prints each line. Second one reads the files names in a directory and prints them if they match a certain pattern (regular expression). You need the list of all files, so do not implement pattern matching part for this project.

- http://rosettacode.org/wiki/Read_a_file_line_by_line
- http://rosettacode.org/wiki/Walk_a_directory/Non-recursively