Assignment 3: File Manipulation via Unsynchronized PERL Processes
Assigned: Thursday, January 30 2020
Due: Thursday, February 13, 2020

Complete the following activities.

1. Study the programs `fork_child.pl` and `fork_parent_Nchild.pl`; both are shown in the lecture handout.

2. Write two PERL programs: `YOURLASTNAME_YOURFIRSTNAME_parent.pl` and `YOURLASTNAME_YOURFIRSTNAME_child.pl`, but substitute your last name for YOURLASTNAME and your first name for YOURFIRSTNAME.

3. The program `YOURLASTNAME_YOURFIRSTNAME_parent.pl` opens (no command line arguments) and reads from the file `in.txt`. The file `in.txt` will contain only two lines. The first line begins with a character 't' or 'T' followed by a positive integer number that is greater than or equal to 1. This number represents the number of children processes the program `YOURLASTNAME_YOURFIRSTNAME_parent.pl` will create. The second line begins with a character 'f' or 'F' followed by the names of one or more file names. The number of file names listed will be equivalent to the number of children processes `YOURLASTNAME_YOURFIRSTNAME_parent.pl` will create. The program `YOURLASTNAME_YOURFIRSTNAME_parent.pl` must handle the following scenarios: (1) the file `in.txt` does not exist and (2) the file `in.txt` cannot be opened. Study the example file `in.txt` provided.

4. Once the parent program reads/processes the data from file `in.txt`, it then creates several children processes.

5. Each child process that executes `YOURLASTNAME_YOURFIRSTNAME_child.pl` opens one of the files specified on the second line of the file `in.txt`. The file opened by a child process is based on its order of creation; the first (id = 0) child process opens the first file, the second (id = 1) child process opens the second file, and so on. Each child process then reads a line from its file and writes the line read to an output file named out#.txt, where # represents the child process's numeric identifier. The program `YOURLASTNAME_YOURFIRSTNAME_child.pl` must handle the following scenarios: (1) the file `in#.txt` does not exist, (2) the file `in#.txt` cannot be opened, or (3) the file `out#.txt` cannot be opened.

6. I strongly recommend that you consider using the PERL `GetOpt` module to process command line argument that you might think about passing to the program `YOURLASTNAME_YOURFIRSTNAME_child.pl`.

7. After all children processes finish execution, the program `YOURLASTNAME_YOURFIRSTNAME_parent.pl` will open all output files `out#.txt`, reads a line from each file, and writes the line to an output file `out.txt`. To determine whether any of the children processes are alive, I recommend reading about the `wait()` PERL system call. Study the example files `out0.txt` and `out.txt` provided. The program `YOURLASTNAME_YOURFIRSTNAME_parent.pl` must handle the following scenarios: (1) `out#.txt` does not exist, (2) `out#.txt` cannot be opened, or (3) `out.txt` cannot be opened.

8. Study the diagram in the file `CSCI4011-63100-201_A03_PERL_Unsync_Proc_Poems.pdf` to understand better the unsynchronized ordering of events.

9. Submit the PERL files `YOURLASTNAME_YOURFIRSTNAME_parent.pl` and `YOURLASTNAME_YOURFIRSTNAME_child.pl` using Moodle before the specified due date and time.

You may speak with your classmates about the assignment, but the work you turn in for a grade must be your own. All instances of academic dishonesty will be addressed per the course syllabus.