CECS 342 Assignment 3 - Functional Languages with Erlang

August 23, 2021

Assignment Description. The purpose of this assignment is to familiarize you with the features and strengths of functional programming languages and also demonstrate the power that these languages can offer for tasks in their areas of specialties.

Part One - Students and Candy. Three students sit in a circle while their teacher gives them candy. Each student initially has an even number of pieces of candy. When the teacher blows a whistle, each student simultaneously gives half of his or her own candy to the neighbor on the right. Any student who ends up with an odd number of pieces of candy gets one more piece from the teacher.

Write an Erlang program that includes a recursive function with four parameters, the amounts of candy for the three students and the number of turns. Each time the function is called it adjusts the number of candy of each student according to the above rules. A helper function should be defined to make this adjustment. At each whistle blow output the amount of candy of each student and the turn number. Stop when all students have the same amount of candy.

The `div` and `rem` operators produce integer values. Remember that the output value using `io:format` must be enclosed in list brackets, `[]`. Test your program using different input including some larger numbers.

Part Two - Frequency Count. Write an Erlang program that counts the word frequencies in the file *assign3-part2.txt*. Erlang uses a list of tuples as a hash table. Write four Erlang functions.

1. This function has a string file name parameter and returns a list of words in the file. Open the file with file:open. The file *assign3-part2.txt* was written in a text editor as one line, so `io:get_line` will read the whole file. `string:tokens` will separate it into words. Its second argument specifies all the delimiters.

   ie: `file:open(‘‘assign3-part2.txt’’,read).`

   ie: `L1= io:get_line(S, ‘’).`

2. This function has two parameters, a string word and a list of tuples, and returns a list of tuples with the word added appropriately. Each tuple is a word key and a frequency value. The `lists:keyfind` method will find a tuple if it exists and return false if it does not. If the word is not found use `lists:append` to add a tuple with the word as key and 1 as the frequency. If the word is found use `lists:replace` to replace the tuple with a new tuple with frequency increased by one.

3. This function has a list of words as parameter and returns a hash table of tuples of words and their frequencies. The `string:to_lower` method will make a word lower case to provided the desired case insensitivity. Use the `lists:foldl` method to build the answer using the function 2.

4. This function outputs the final hash table sorted by frequency from high to low using the `lists:sort` function. Its one parameter is the file name. It uses functions 3 and 1.

Deliverables. Submit your source code files through Beachboard Dropbox. Make sure that you have adequately commented your original source code, else I will not grade it and you will receive zero points for the assignment!