*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*\* WEB322 – Assignment 1*

*\* I declare that this assignment is my own work in accordance with Seneca Academic Policy.*

*\* No part of this assignment has been copied manually or electronically from any other source*

*\* (including web sites) or distributed to other students.*

*\**

*\* Name: Kojo Anyane Obese Student ID: 137653226 Date: May 15, 2024*

*\**

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

Brief explanation of the code

This script allows the user to process either a file or a directory. Based on user input, it performs different actions.

* File Processing: Counts characters, words, finds the longest word, and identifies the most frequently repeated word.
* Directory processing: Lists all files in reverse alphabetical order and shows their sizes.

Code Breakdown

1. Import node modules:

*// Import the 'fs' module for file system operations*

*const* fs = require("fs");

*// Import the 'readline' module to create an interface for reading input from the command line*

*const* readline = require("readline");

* ‘fs’: used for file systems operations
* ‘readline’: used for reading input from the command line

1. Create interface for User input:

*const* rl = readline.createInterface(process.stdin, process.stdout);

1. Function definitions:

* Find longest word:

function findLongestWord(fileDataArray) {

*return* fileDataArray.reduce((longest, current) => {

*// Compare the length of the current word with the longest word found so far*

*return* current.length > longest.length ? current : longest;

});

}

* Find most frequent word:

function findMostFrequentWord(fileDataArray) {

*// Return null if the array is empty*

if (!fileDataArray.length) *return* null;

*// Object to store the frequency of each word*

*var* words = {};

*// Initialize the most repeated word and the maximum repetitions*

*var* mostRepeatedWord = fileDataArray[0], maxRepetitions = 1;

*// Iterate over each word in the array*

fileDataArray.forEach(word => {

*// Increment the word count in the 'words' object*

if (!words[word]) words[word] = 1;

else words[word]++;

*// Update the most repeated word and the max repetitions if current word has more repetitions*

if (words[word] >= maxRepetitions && word != mostRepeatedWord) {

mostRepeatedWord = word;

maxRepetitions = words[word];

}

});

*// Return the most repeated word and its count*

*return* [mostRepeatedWord, maxRepetitions];

}

* User interaction:

rl.question(

"Do you wish to process a File (f) or directory (d): ",

function (answer) {

*// If the user chooses to process a file*

if (answer === "f") {

*// Ask for the file name*

rl.question("File: ", function (filename) {

*// Read the file content*

fs.readFile(filename, function (err, file) {

if (err) {

*// Print an error message if there is an error reading the file*

console.log(err.message);

} else {

*// Convert the file content to a string and replace multiple spaces with a single space*

file = file.toString().replace(/\s+/g, " ");

*// Print the number of characters (including spaces) in the file*

console.log(

"Number of Characters (including spaces): " + file.length

);

*// Remove non-word characters and split the file content into words*

file = file.replace(/[^\w\s\']/g, "").split(" ");

*// Print the number of words in the file*

console.log("Number of Words: " + file.length);

*// Find and print the longest word in the file*

console.log("Longest Word: " + findLongestWord(file));

*// Find and print the most repeated word in the file and its count*

console.log("Most repeated Word: " + findMostFrequentWord(file).join(" - ") + " times.");

}

});

*// Close the readline interface*

rl.close();

});

*// If the user chooses to process a directory*

} else if (answer === "d") {

*// Ask for the directory name*

rl.question("Directory: ", function (directory) {

*// Read the content of the directory*

fs.readdir(directory, function (err, dirContent){

if(err){

*// Print an error message if there is an error reading the directory*

console.log(err.message);

} else {

*// Sort the directory content in reverse alphabetical order*

dirContent.sort().reverse();

*// Print the sorted directory content*

console.log("Files (reverse alphabetical order): " + dirContent.join(","));

*// Print the size of each file in the directory*

dirContent.forEach(file => {

console.log(file + ": " + fs.statSync(directory+"/"+file).size + " bytes");

})

}

})

*// Close the readline interface*

rl.close();

});

*// If the user enters an invalid selection*

} else {

console.log("Invalid selection");

rl.close();

}

}

);

Explanation of Outputs

1. File processing

Do you wish to process a File (f) or directory (d): f

File: post.txt

Number of Characters (including spaces): 2974

Number of Words: 437

Longest Word: Pellentesque

Most repeated Word: amet - 11 times.

1. Directory processing:

Do you wish to process a File (f) or directory (d): d

Directory: files

Files (reverse alphabetical order): zucchini.txt,potatoes.txt,oranges.txt,apples.txt

zucchini.txt: 1360 bytes

potatoes.txt: 1830 bytes

oranges.txt: 1305 bytes

apples.txt: 1512 bytes

1. Error handling:

Do you wish to process a File (f) or directory (d): a

Invalid selection