CSCI251 Assignment 1 – README

Full name: Quy Binh Nguyen

Student ID: 7613623

---

1) Describe the role and content of each source-code file.

(a) driver.cpp

- Role:

This file contains main() function, which is used to execute the whole program.

- Content:

On the top of the file, I #include “header.h”. “header.h” is a header file, which includes the declaration of variables and functions of the program.

Within the main() function, first of all, I initialized the variables (the maximum number of workers and tasks is 20), and also the default values of totalTasks, currentTask, totalWorkers, currentWorker to 0, so that I can execute all of them from the first task and first worker of each task.

After, I wrote 4 functions in sequence for program execution. These functions are declared in “header.h” file, and are defined in “header.cpp” file.

(b) header.h

- Role:

This file contains the prototypes for the functions that I write in “header.cpp” file.

- Content:

On the top of the file, I #include <string>. This library is used to work with the strings that I declared in the strucks.

Next, I wrote “using namespace std;” in order to avoid the need to write “std::” before each of the library components when I use them in the whole file.

Next, I declared “Tasks” and “Workers” structs to group related data members of each struck together.

Finally, I declared 4 functions for program execution. Their definition is written in “header.cpp” file.

(c) header.cpp

- Role:

This file contains the implementations of my functions declared in “header.h”

- Content:

On the top of the file, I #include all libraries that I need to use in the program:

+ “header.h”: to include the declaration of variables and functions of the program

+ <iostream>: to perform input and output

+ <fstream>: to perform file input and output

+ <sstream>: to provide a stream that can be used to read/write strings

+ <string>: to work with the strings

+ <random>: to perform randomness

+ <ctime>: to work with time and date

Next, I wrote “using namespace std;” in order to avoid the need to write “std::” before each of the library components when I use them in the whole file.

First function - tasksInData(): This function opens the input file “Tasks.txt” and then read the file, using getline() and istringstream. Brief explanation is in the comment of the file. However, one more explanation in this function is that, after the taskId, description, uncertainty, difficulty, and the ‘workers’ string (which is variable literalStr in my program), it is the array of the workersIdList which differs from each task. Therefore, I used nested loop to read the workersIdList of each task.

Second function – workersInData(): This second function basically is the same with the first function (as it has the same logic with the first function, therefore I didn’t comment in this function). It opens the input file “Workers.txt” and then read the file. This function is used to retrieve the value of each worker’s ability and worker’s variability, which are used to determine the mean and standard deviation in order to perform worker performance and his/her average performance.

Third function – workerPerformance(): This function is used to generate random numbers to calculate the average performance of a worker. The returned value is used to display the average performance of a worker in writeData() function.

Fourth function – writeData(): This function is used to write the data after reading the inputs. This is just an easy function to output everything as the same with the example output, therefore all of the explanations are demonstrated in the comment of the file.

2) How to run my code.

Step 1: Open and log in into capa.its.uow.edu.au (I use Bitvise SSH Client as instructed by the lab demonstrator)

Step 2: Click ‘New SFTP window’

Step 3: Find the folder that I hold all of the files needed to run the program.

Step 4: Drag and drop all of the files from ‘Local files’ to ‘Remote files’

Step 5: Click ‘New Terminal Console’

Step 6: Write the command

g++ driver.cpp header.h header.cpp -std=c++17 -o run

./run Tasks.txt Workers.txt Output.txt