Assignment #2
Bash, Makefiles, and Utilizing the HPCC
60 Points

Problem

Develop an HPCC submission script written in *BASH* that can perform all the actions described in the Script Requirements (below). Your script is required to work in the HPCC using the standard bash scripting environment and should execute when run using the following command: `sbatch assignment_2.sh`

Prerequisites

Before writing your bash script, you need to perform the following steps:

1) Create a new directory to work in and change into that directory.
2) Make a recursive copy of the required assignment #2 code directory located at "`/lustre/work/errees/courses/cs4352/assignment2/code`"
    a. Contains 2 directories, *decrypt* and *transform*, each containing source code that requires compilation.
3) Change into the *decrypt* directory and compile (preferably using a makefile) the C++ code.
    a. Compile the application on quanah.hpcc.ttu.edu using the g++ compiler which is part of the GNU 5.4.0 module on HPCC.
    b. You must name the compiled executable as `decrypt`. Then copy the compiled executable into the same directory you will write and run your bash script in.
4) Change into the *transform* directory and compile (preferably using a makefile) the C code.
    a. Compile the application on quanah.hpcc.ttu.edu using the gcc compiler which is part of the GNU 5.4.0 module on HPCC.
    b. You must name the compiled executable as `transform`. Then copy the compiled executable into the same directory you will write and run your bash script in.
5) Your assignment #2 bash script will need to reference these applications using relative, not absolute, paths. Ensure your script is written with the assumption these two applications will be in the same directory that your script is executed from.

Script Requirements

You must produce a bash script written to work with the HPCC's SLURM scheduler. Your submission script should use the default HPCC values for all SBATCH options, with the following required exceptions:

Job Name	`cs4352_a2`
Partition Name	`quanah`
Number of Nodes	`1`
Tasks per Node	`1`
RAM per CPU	`5370MB`
Run Time	`00:05:00`

Your BASH script should then perform the following actions:

1) Load the required GNU 5.4.0 module using the module load command.
2) Print your Full Name to a file called "*a2_results.txt*" located in the current working directory.
3) Append your R# to the 2nd line of a file call "*a2_results.txt*" located in the current working directory.
    a. Example: R123456
4) Execute your compiled `decrypt` application by passing along the absolute path to the encrypted file located at:
   `/lustre/work/errees/courses/cs4352/assignment2/input/encrypted.txt`
    a. Keep in mind, unlike other references in your script, this file is required to be referenced using an absolute path.
    b. The `decrypt` application requires this file path be passed to it as a required command line argument.
    c. The output of this application will be the absolute path to a file located elsewhere in the HPCC storage system.
    d. The file path returned by `decrypt` may be different during grading than the one given by the application during testing.
5) Execute your compiled `transform` application by passing along your R# as a command line argument.
    a. Keep in mind, this application requires you pass along a positive integer value, so leave off the 'R' when passing your R# to the application.
    b. The `transform` application requires this integer be passed to it as a required command line argument.
    c. The output of this application will be a single-digit or multi-digit positive integer value.
6) Using the command(s) of your choice and the output from the `decrypt` and `transform` applications, your script must then do the following:
    a. Open the file located at the path given by the output of the `decrypt` application.
    b. Append the *N*th line of this file to your "*a2_results.txt*".
        i. Where *N* is the integer value returned by the `transform` application.

Additional Information and Hints

1) Start by running the commands one at a time without a script. Then add the commands that work to your script. Don't forget to test run the script to ensure everything is working as it should be.
2) Once your script is completed, work on making it run as an HPCC submission script.
3) Your program may not make use of absolute paths except to access the *encrypted.txt* file. All other paths must be done using **relative** paths.
4) Keep in mind that the *encrypted.txt* file used by you for testing may be different than the one present when your code is graded.
5) Helpful Links:
    a. Makefile Tutorial
    b. HPCC User Guides
        i. Pay particular attention to the "Job Submission Guide" for Slurm.

What to turn in to BlackBoard

A zip archive (.zip) named <FirstName>_<LastName>_R<#>_Assignment2.zip that contains the following files:

- assignment_2.sh
    - This name is required and is case sensitive.
- The binary executable produced when you compiled the `decrypt` application.
- The binary executable produced when you compiled the `transform` application.

Your files will be unzipped and then executed using only the command "`sbatch assignment_2.sh`". It is your responsibility to ensure that you have all of the necessary files and executables added to the zip archive and in a directory tree structure that will work from wherever your code is unzipped in the HPCC. This means your script file should know the relative path to all of your binary executables.

Your output will be graded as entirely incorrect if it fails to correctly execute using the sbatch command listed above. This includes for reasons such as naming files incorrectly, forgetting to add in required binary files, or improperly utilizing absolute paths that will not work when tested by another HPCC user.

Once you have zipped up your assignment to be turned in, I would strongly advise you to do the following:

1) Rename your pervious working directory to ensure all absolute paths into it would now break.
    a. For example, if you had everything in /home/errees/assignment_2, you rename it to something like /home/errees/assignment_2_old.
2) Create a completely new directory in the HPCC that has a completely different name than you used for testing before, for example /home/errees/abc123_test
3) Unzip your zip archive into this new directory.
4) Change into the new directory and run the command: `sbatch assignment_2.sh`
5) Once the run has completed in the HPCC, check to see if your output file is correct.