Hacettepe University

Computer Science and Engineering Department

Name and Surname:	Ozan Özenoğlu
Identity Number:	20724945
Course:	Bil-235
Experiment :	Experiment 5
Subject :	Bash Programming
Data Due:	28.12.2010
Advisors:	Dr. Ahmet Burak CAN , R.A Ersin ER
e-mail:	b20724945@cs.hacettepe.edu.tr

Software Using Documentation:

First you need to make all script files executable before execute them. “chmod 700 *.sh” command will give permissions that you need to execute the scripts. You can run a script with this command “./” . After this operation firstly you need to run exp_init.sh with a parameter.. In input section the paramater is described more clearly, it is enough for now if you know this parameter is local base directory name. Next you will need copy orginal submission into orginal_submission directory. Also you must to copy referance input and output. There is one more thing you need to do you must write a exp.conf file. After preparing you should run exp_build.sh script. After building , you should run exp_execute.sh . And next we need calculate points that stundets gain. So you need to run exp_eval_execs.sh , exp_eval_report.sh and finally exp_merge_results.sh . İf you want to clean files that generated by scripts you can run exp_clean.sh

Important note: You must give base directory name as a parameter to all scripts file before run.

You can find final result in basedirectory/evaluation_results directory.

Provided Possibilities: There is no extra utilities except main goal of this program.

Error Messages: There is no error message that can given by program so we assume user will enter in perf

correct parameter in commond line. More about parameters info please read below (Section System Chart)

Software Design Notes: This scope is for software developers who interest the developing bash script in unix operating systems.

Description of the problem: The problem is that we job which need to get handled by automaticly because of count of jobs. There are many submissions file which are zip . These submissions file include a source code . So we need to build source code which described how to do in exp.conf file. There are referance input file and referance output file so we can need to use this referances as parameter to program or look for difference between execution results and reference outputs while calculating score. After all we need to clean unnecessary file from base directory which given start .

Description of the Solution: First ı create a bash script file that prepare folder that we'll need after. This script is very simple so ı don't focus on this. After user copy necessary file , we need to unzip file. First ı enter orginal submissions directory with cd command then with for and by using ls command I'm making a loop on all submission files. By using unzip command ı extract all file into extracted_submissions directory with –d ../extracted_submissions paramater . After extracting operation ı need to build source code but we don't know what is the platform and how we can build the code so we user must define build command in exp.conf. by using cat command ı read exp.conf and give result to grep function. I search the command_line by using grep and give output(command_line) to awk . In awk ı parse the command according to "=" and saved second part of line. So we have a build command now. Next ı loop on all extracted submissions and enter submissions directory. In the directory ı run build command by using eval command. After build operation we need to execute binary file. But we don't know how we do this. So user must define it in exp.conf file. In same method while getting build command ı get execute command from exp.conf file.Before execution operation ı saved reference input names into a array by using ls commad.After that ı loop on all submission directory . I enter submissions directory while loop and make a new loop in reference input array then by using eval command ı execute the execute command with ../..reference_input_data/$reference_name and ../..execution_output_data/$reference_name after this operation we have execution output that generated by student program. Now we need to calculate score. To calculate score we need to find difference between execution output and reference output. If is there any difference we'll give zero point for this test. Else we'll give one point for this test and write execution_result file. To do this operation first we need get reference output data names . I get the names into a array with list=$(ls) command . Then ı enter execution output data and ı made a loop on all submisson directory. While loop ı enter all directory by one by and ı look for if is there any difference between files. While ı looking any difference first ı controll the file is exists or not. İf there is no file ı give zero point. İf everything is okay point one is gone to student for this test. But all this look and difference operation ı need to prepare result file. So in start I enter into eveluation_result directory and make a file with name execution_result with touch command. And while loop on submission directories ı write submission directory name first into this result file then ı start to calculate and write result. So result file like look great and understandable. After this operation ı need to calculate score which belong report files. To this operation everything is same with execution output score calculating but there is one difference which is the way calculate score. İf the report file size greater than zero we give one point else zero point. So ı look up size of report file with du command and write results into /evaluation_result/report file . Finally we can merge two results of evaluation . To do this firstly ı open report file with cat command then ı redirection output cat command with pipe operator "|" to awk then awk split the string according to ":" then write it as parse1 parse2 into final_results file. Then by awk ı get stundet number and give output to grep to find execution points in execution_result file that belong student. Then ı write points into final_result file. Everything is complete now. İf user want to delete unnecessary file from base directory we have to remove files from build_logs , eveluation_results , execution_output_data and if user want to delete a specifics file or files . To know if there is additional remove request from user we need to get it from exp.conf file. By using grep and awk command we get this command from exp.conf file and we execute this command while loop on directories which is in extracted_submissions directory. I remove files with rm –rf destination/*. So ı can remove all files that belong destination folder. Finally we done all job.

System Chart

INPUT

input device: keyboard

input file : reference input,output files and exp.conf file

other inputs : arguments

arguments formation : string

an example argument : bil137-3

argument is the base directory which all folders will be make in it.

output device: monitor

output file: execution , reports , final_results,*.log

other outputs: There is no other outputs.

output formation for *. Log files : if is there any error or warning while build operation we store the stderror from compiler in this files.

Output formation for execution:

Student Id: point point point ...

Output formation for reports:

Student Id:point point point ...

Output formation for final

Student-Id point point point

an example output for final: 21233322 1 1 0 1

an example output for report: 212323221:1

an example output for executio: 232322212: 1 1 1 0

an example output for *.log files: source/main.c: In function ‘main’:

source/main.c:10: warning: format ‘%d’ expects type ‘int *’, but argument 3 has

Main Data Structures: There is no data structure

Algorithm:

Create folder into base directory

Extract submission file into extracted submissions folder

Build all submissions

Execute all submissions

Evaluate all submissions execution

Evaluate all submissions report

Merge execution and report file

Clean unnecessary file

Done!

Special Design Properties : There is no extra design properties . Only ı try to make things in a single line as possible as.

Bugs and Software Reliability:

There is no bugs that ı can find.

Software Extendibility and Upgradability:

In future additional features may be added easily.

Performance Considerations

Program works well

Comments

This experiment is a window that opening linux experience. I love bash scripting and also linux o.s. and Everything is installed in linux. Linux is very safe and fast operating system. And also it's completely free.

Referance:

http://tille.garrels.be/training/bash/