Hacettepe University
Department of Computer Science and Engineering
BIL235 Programming Laboratory
Experiment 5

Subject:	Writing UNIX Shell Scripts
Advisors:	Prof. Dr. Ali SAATÇİ
	R.A. Kerem ERZURUMLU
Submission Date:	26/11/2008
Deadline:	19/12/2008 – 23:59
Programming Language:	Bourne Again Shell (`bash`)

AIM

The aim of this experiment is to get students familiar with both the session and the programming environment of the Unix (Linux) Operating System.

Unix and its variants have become the de-facto (i.e., not in writing but by common consent) industrial standard for server systems. In recent times, flavors of Unix (Linux) have been ported to machines with limited hardware, such as PC's (Personal Computers). A working knowledge of Unix will probably be the best advantage of a computer engineer in the future.

BACKGROUND

With Unix, users can access the system via a terminal. After the autentication phase (login) a special program called "shell" runs for each user outputting a "prompt" (normally a $ sign) on the screen. A shell can be described as a layer between the user and Unix itself. It is a command line interpreter through which a user can execute ready made programs called shell commands such as `ls, ps, cp, mv, cat` etc. A shell has also programming capabilities like loops and conditional executions. It then gives users the possibility to write "scripts" to define a specific job as a whole. A script is a collection of commands (or executable programs) along with programming language structures such as loops and conditions. A script can be considered as a program where the statements (instructions) are the commands. A shell script is stored in a file with a given name. It is executed every time this file name is stated.

The most common shells for Unix are `sh (Bourne Shell)` and `bash (Bourne Again Shell)`. In this experiment, you have to use `bash`. For its usage, please refere to the documents defined in the References.

Shell redirection is a base for UNIX systems. Unless clearly specified, shell commands takes their input from the keyboard and directs their output to the screen. The ">" operator specifies to the shell that the output of a command is to be redirected to a file (following the ">" sign). Similarly, the "<" operator tells the shell that the input is to be taken from a file. Additionally, there is another feature of UNIX that allows the user to specify successive file redirections. Called *pipe*, this feature allows the redirection of the output of a program to the input of another one.

OBJECTIVE

In this experiment you are asked to write a simple "table manipulator script" (a shell program). A table consists of a defined number of columns and rows. The script will be used to handle tables (to create or delete tables) and records (insert or delete data rows within the tables) related to a given database.

Your script should have a command line in order for the users to type commands such as: create table, delete table, insert or delete row to a given table, quit the program, help etc.

Table Operations:

“`create_table`”: Before creating the table your program should ask the user: the table name, the number and the names of the columns.

“`delete_table`”: Your program should ask the user the table name and confirmation before deleting.

Record Operations:

“`insert_record`”: This command will append a row at the end of a table. Your program should ask the user: the table name, fields’ names and contents.

“`delete_record`”: This command will delete row(s) according to a given criteria. For instance, delete records where a given column includes a given string; delete record #3;#7; delete record #3-#7; etc. Your program should ask confirmation before deleting.

“`sort table`”: This command will put rows in alphabethical/numerical order according to a given column. Your program should ask the user: the table and column names as well as the sorting method (alphabethical/numerical; increasing/decreasing order).

Further Reading

1. Unix’s on-line manual system which is accessed by command `man(1)`,
2. Using UNIX (CS Library No=122),
3. Advanced Unix Programming (CS Library No=139,151),
4. http://www.linux.org.tr
5. http://www.google.com
6. http://www.penguen.net

Please read also the following manuals: `cut(1)`, `sed(1)`, `awk(1)`, `grep(1)`, `sort(1)`, `bash(1)`, `for(1)`, `while(1)`, `if(1)`, `diff(1)`, `let(1)`, `expr(1), dialog(1)`

NOTES

1. You are asked to follow announcements made to Courses.Bil235 newsgroup which is located at nntp://news.cs.hacettepe.edu.tr .
2. Your report and program must be submitted at the same time. If an experiment does not have report, it will not be graded.
3. Your programs will be checked with an automated Plagiarism Detector.
4. Valid Soft-Copy formats for your report are HTML and PDF.
5. e-mail submissions are not accepted.
6. Late submissions will not be accepted.
7. At the submission your experiments will be checked for common errors. Works with this kind of errors may be refused for grading. Do not leave everything to the last minute.
8. The submit system's properly working is assured until 17:00. The brake down of the submit system after 17:00 is your own risk.
9. Please send any additional questions to Courses.Bil235 newsgroup.

Good Luck