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:	11/12/2001
Deadline:	02/01/2002
Programming Language:	Bourne Again Shell (`bash`), ANSI C

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 multitasking 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 loops and conditions. It 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.

BIL235 Experiment V

OBJECT

In this experiment you are asked to write a shell script executing 3 different tasks as a whole every night with the help of `crontab(1,5)`[1] . You are also asked to develop a small C program in order to use within your script for output creation purposes.

Task A: Create an information file for each user defined on the system including the following fields:

1. Date (Day on which the script runs)
2. User real name, User ID,
3. Group name and total number of group members,
4. Total login time in minutes for the day,
5. Number of e-mails remaining in user's inbox,
6. Total disk space used by user's home directory.

This file should be created following a text template including the above fields. The text part of the template should be made changeable.

Task B: Compress using `tar(1)` files in defined directories within the script and Back-up them under a given destination directory also defined in the script.

Task C: For each user Find files with a given extention under the users' home directory and Send the output list as an e-mail to the admin's e-mail address.

C Program: You have to write a simple program in order to use it in your script to create output. This program will append a source file (e.g. an information file for a specific user) to the end of a destination file (e.g. the total information for all the users). The name of the destination file will be given to the script as an argument. The usage of this program should be thougth with UNIX input/output redirection.

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 the files following the "<" sign.) 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.

REFERENCES

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)`, `mail(1)`, `bash(1)`, `for(1)`, `if(1)`, `diff(1)`, `netstat(8)`, `md5sum(1)`, `let(1)`, `expr(1)`, `last(1)`

NOTES

1. You are asked to give your `Makefile` with your program.

[1] Numbers in paranthesis indicates that the manual pages section.

2. Backup files must include their creation date.
3. In **Job C** more than one file extention should be possible.
4. You are asked to follow announcements made to "bil235 discussion list". If you are not subscribed yet, please subscribe to it by sending an e-mail to:

   `majordomo@cs.hacettepe.edu.tr`

   with a message body of

   "`subscribe bil235`".
5. Your report and program must be submitted at the same time.
6. Your report must include your source codes and should be based on a special template given at:

   `http://www.penguen.net/turkce/belgeler/rapor.pdf` .
7. Office hours will be held on Tuesday's and Friday's. You can also send e-mails to `kerem@linux.org.tr` for your additional questions

Good Luck