CSc352 (Summer 03) Homework 7

Start: 08/01/03
Due: 08/08/03 (9am)
Turnin ID: cs352_assg7

Turnin file list: backup, regtest and factors

0. Background Readings

- "date" command man page
- "diff" command man page

1. Backup Utility

You are to write a C Shell script named **backup**. It helps you to backup files or directories. The command line syntax of the utility is like this:

backup pathname

"pathname" is the name of the file or directory you want to backup.
It will tar and gzip the file or directory (specified by **pathname**) into a single file named **pathname-mmddyy-hhnn.tar.gz**, where mm, dd, yy, hh and nn are the current month, day, year, hour and minute. Then the .tar.gz file is moved into a backup directory ~/BACKUP. For example, at 20:35 on Aug. 01, 2003, you run the command "backup hw6" (assuming you have a subdirectory named hw6 in the current directory), the hw6 subdirectory is packed into a file named hw6-080103-2035.tar.gz and stored in ~/BACKUP.

You are supposed to handle all valid UNIX pathname as the command line parameter, for example:

```
backup mydoc
backup mydoc/
backup /home/mike/mydoc
backup /home/mike/mydoc/
```

The first 2 commands above do the same thing – backup the directory "mydoc" under the current working directory.
The last 2 commands above do the same thing – backup the directory "mydoc" under "/home/mike" by the following tar command for example:

```
tar cf mydoc-xxxxxx-xxxx.tar /home/mike/mydoc
```

Notice that the filename of the resulting tar file begins with "mydoc", which is the name of the directory you are actually backing-up instead of the full pathname.

To make it more convenient to do the compare during grading, you need to use the exact version of the tar and gzip utility: /usr/local/bin/tar and

/usr/local/bin/gzip.

You may assume all the command line arguments are valid. So no error checking is necessary.

2. Regression Test Utility

You are to implement a regression test utility to help with your program test. The command line syntax of the utility is like this:

regtest prg testcase_dir num

"prg" is the name of the program you are testing.
"testcase_dir" is the directory where all testcases are stored.
"num" is the number of testcases we are going to test.

We have the following assumptions:

- the program you are testing is in the current working directory
- the program you are testing always reads the input from **stdin** and gives out the result by printing into **stdout**
- we have at most 99 test cases and they are all named testXX, where XX is the number of the testcases. The numbers are always assigned in the lower end of whole numbering space. (For example, if we have 8 testcases, their names are test01, test02, …, test08, no numbers greater than 08 will be used)
- for each testcase testXX, the expected output of the testcase is in a file named testXX.out (also in directory **testcase_dir**).

Basically what you do in the **regtest** script is to run the program against each of the testcase in **testcase_dir** and redirect the output into a temporary file and diff it with the expected output of the testcase. If no difference is reported by diff, your script keeps silent (without printing anything), otherwise print out the number of the testcase failed. The numbers are separated by spaces. Only when all testcases pass, you return 0 from the script; otherwise return 1 showing the failure of the regression test.

You may assume all the command line arguments are valid. So no error checking is necessary.

3. Prime Factors

You are to write a C Shell script named **factors**. The command line syntax is like this:

factors num

"num" is an integer. If it is less than 2, you should print out nothing, otherwise print out the prime factors (print only one when there are more than one occurrence of the factor) each per line in ascending order. For example, "factors 405" will produce the following result:

3
5

In the above example, 405 = 3*3*3*3*5, but we only print “3” once.

You may assume all the command line arguments are valid. So no error checking is necessary.

PS:

For detailed information of the output format, please refer to some sample testcases and expected results in /home/cs352/summer03/assignments/hw7.