Basic Shell Programming Exercises

1 Questions

Make all these scripts executable programs on your PATH.

1. Write a simple shell script that takes *any* number of arguments on the command line, and prints the arguments with "Hello " in front. For example, if the name of the script is `hello`, then you should be able to run it like this:

   ```
   $ hello Nick Urbanik
   Hello Nick Urbanik
   $ hello Edmund
   Hello Edmund
   ```

2. Write a simple shell script that takes two numbers as parameters and uses a `while` loop to print all the numbers from the first to the second inclusive, each number separated only by a space from the previous number. Example, if the script is called `jot`, then

   ```
   $ jot 2 8
   2 3 4 5 6 7 8
   ```

3. Write a script which displays "`Good morning`", "`Good afternoon`" or "`Good evening`", on the monitor, depending on the time of running the script.

4. Write a script which reads a number in units of seconds and converts it to the units hours:minutes:seconds and prints the result to standard output.

   Your script must prompt for re-input if a negative value is input

   ```
   Enter number of seconds: 12345
   Result:
   12345 seconds in hours:minutes:seconds is 3:25:45
   ```

5. Suppose that the script you wrote for question 2 is called `jot`. Then run it calling `sh` yourself. Notice the difference:

   ```
   sh jot 2 5
   sh -v jot 2 5
   sh -x jot 2 5
   ```

   Do you notice any difference in the output from last two?

6. Write a script `calculate`, which accepts 4 arguments a, b, c, d and prints the value of $a \times 20 - b \times 2 + c \div d$ to standard output.

   An example of executing the script:

   ```
   $ calculate 2 12 5 2
   The value of "2*20 - 12*2 + 5/2" is 18
   ```

7. Write a shell script that, for each `.rpm` file in the current directory, prints the name of the package on a line by itself, then runs `rpm -K` on the package, then prints a blank line, using a `for` loop.

   Test your script on the files in `/home/nfs/rh-7.2-updated/RedHat/RPMS`.

   > The option `rpm -K` checks that the software package is not corrupted, and is signed by the author, if you have imported the author's public key with the command:
   > ```
   > $ cd /home/nfs/redhat-8.0
   > $ sudo rpm --import RPM-GPG-KEY
   > ```

8. Modify the script you wrote for the previous question to print the output of `rpm -K` *only* for *all* the files that fail the test. In particular, if the package's GPG signature fails, then your script should display the output of `rpm -K`. There are at least two packages in this directory which do not have a valid GPG signature; one of them is `redhat-release-7.2-1.noarch.rpm`; what is the other?

   Here is output from `rpm -K` for two packages, one with no GPG signature, the other with:

   ```
   $ rpm -K redhat-release-7.2-1.noarch.rpm bash-2.05-8.i386.rpm
   redhat-release-7.2-1.noarch.rpm: md5 OK
   bash-2.05-8.i386.rpm: md5 gpg OK
   ```

   Test it in the same network directory as for the previous question.

9. Write a shell script to add a local group called `administrator` if it does not already exist. Do not execute any external program if the `administrator` group already exists.

10. Download a copy of the bogus student registration data from `http://nicku.org/snm/lab/regular-expressions/artificial-student-data.txt`. Use this for the following exercises, together with the `grep` program:

    (a) Search for all students with the name "CHAN"

    (b) Search for all students whose student number begins and ends with 9, and with any other digits in between.

    (c) Search for all student records where the Hong Kong ID has a letter, not a number, in the parentheses.

    (d) Do the same exercises, but display only the students' names, or student number. You will need a program such as `awk` (or even `cut`) to select the appropriate columns from the output of `grep`.

11. Write a shell script to take a file name on its command line, and edit it with `sed` so that every instance of "`/usr/local/bin`" is changed to "`/usr/bin`"

12. Write a shell script to take a file name on its command line, and edit it using `sed` so that every line that begins with the string `server`:

    `server` *other text*

    is edited so that averything after "`server` " (i.e., the "*other text*") is replaced with the string "`clock.tyict.vtc.edu.hk`", so that the line above looks like this:

    `server clock.tyict.vtc.edu.hk`

    Test this on a copy of the file `/etc/ntp.conf` that is on your computer. (Install the package `ntp` if it is not there).