CT 320: Assignment 2 (A2)

Individual Work

Bash Shell Script

The purpose of this assignment is to write a bash script called `P2.script` that is both practical and uses many features of bash. We are also preparing for Assignment 3 in which you will use the script you have written to implement a periodic process.

Description

This assignment consists of writing a backup and restore program for user data on a Linux system. This assignment will be graded on how you meet the requirements in this document, so please read all parts of the assignment carefully. We have listed the requirements and a list of steps to follow.

Requirements

The manager of system administration at your company is evaluating backup solutions from different companies, and has not yet found one that meets all the requirements. In the meantime he wants an interim solution for backing up user data on the Linux workstations in the engineering lab. Each workstation can have one or more user, each of whom often uses the workstation every day and has large amounts of data. The manager asks you to develop a bash script that will backup data on all workstations nightly. The same script must be capable of allowing you to restore data from the backup in case recovery is needed because files were accidentally deleted.

Steps

Develop a bash script called `P2.script` that takes three command-line arguments. The first argument is an action, which is either `-b` for backup or `-r` for restore. The second is the directory where user accounts reside, which would normally be `/home`. The third argument is the directory where the backup files will reside, for example `/tmp/archive`.

The `-b` action causes the script to enumerate all users in the specified directory, create a tar file that contains the entire contents of each user's home directory, and copy the tar file to the backup directory. The tar file name should be *username*`.tar.gz`, where *username* is, well, the user's name (login). The script must also create a file in the backup directory called `repository.txt` that contains backup status, including the user name, home directory, backup date, number of files, number of directories, and total backup size in bytes.

When the `-r` action is requested, the script will enumerate all users in the specified directory, search the repository file and report each line matching the specified username. If no backup is found, the script should report an error message that says `"No backup found for username"`. For every username that has a backup, the script will find the associated tar file and unpack it into `/tmp/scratch` It will then do a recursive file-by-file comparison of the files in the backup against the files in the home directory. If a file exists in the backup but not the home directory, it will restore it. The scratch area should be deleted before completing this command.

Here is an example of the format of `repository.txt`, in which fields are separated by commas:

```
    smith,/home/smith,Thu Sep 12 10:02:13 MDT 2013,100,10,14598723
    jones,/home/jones,Thu Sep 12 10:02:13 MDT 2013,6,1,192845
```

Testing

To verify your script, you must do the following:

1. Create a directory in your home directory called `~/home`, and create user directories in it called catmull, phong, gouraud, blinn, and kajiya.
2. Populate the user directories. At least one user should have at least two levels of directory hierarchy, and every user should have multiple files.
3. Put files in the hierarchy that have different content, size, and protection, for example you could include audio, video, photo, and text files.
4. Use the `find` command to enumerate all of the files in `~/home` with a full listing including size, date, and protections, and store it in `baseline.txt`.
5. Run the script as `./P2.script -b ~/home /tmp/archive`. Make sure that the tar files are created correctly in `/tmp/archive`, and check the repository file.
6. Delete three or four random files from different user directories at different levels in the hierarchy to simulate files being accidentally lost.
7. Run the script as `./P2.script -r ~/home /tmp/archive` to restore any missing files for all users.
8. Repeat the `find` command to enumerate all of the files in `~/home` with a full listing including size, date, and protections, and store it in `restored.txt`.
9. Verify that `baseline.txt` and `restored.txt` have identical contents, thereby showing that the restore worked correctly.

Hints

No logging of any sort is required for the assignment, but you will be more efficient if you make the script echo commands and/or print out debug information at each step.

Submission

Submit `P2.script` by the due date to RamCT. You may submit the assignment up to 24 hours late for a 20% penalty.