

Bash Scripting

Due: October 1, 2019 on myCourses at 23:55

Lab C and Mini 1 will provide some background help for this mini assignment.

QUESTION 1: A Project Setup Script

Software developers use scripting to support good software engineering practices. For this question we will explore a standard way that software developers organize their home directory when writing software.

Write a script called: `mkproj` `ARGUMENT` `OWNER`

Example: `./mkproj Ass2 jvybihal`

Where:

- The script's name is `mkproj`
- `ARGUMENT` is the name of the software project the developer wants to initialize.
- `OWNER` is the username of the person whom the project belongs to.

Use `vi` to create the script.

The script must do the following in the order presented:

1. Make sure the script is running in Bash
2. If the user did not provide the correct number of arguments, then display the following error message and then terminate the script: "Error: Wrong number of arguments. Please type `./mkproj ARGUMENT OWNER`. The argument is the name of the project. The `OWNER` is the user name the project belongs to."
3. If the directory "Projects" does not exist, then the script creates that directory. If the directory already exists then nothing is done.
4. Change the working directory to Projects
5. If the directory `ARGUMENT` (i.e. the project name) already exists within the directory Projects, then terminate the script with the following error and change directory to the parent directory (where we came from): "Error: The project name has already been used. Please select another project name and try again."
6. Create the directory `ARGUMENT` (i.e. the project name).
7. Change the current directory to the directory `ARGUMENT` (i.e. the project name)
8. Create the subdirectories: `archive`, `backup`, `docs`, `assets`, `database`, `source`. Then use the `ls` command to display them as verification to the user.
9. Create a `readme.txt` file in the `docs` directory with the following default information: "Created by:" and then insert the `OWNER` name. Display the file as verification to the user.

This script automates the creation of projects (each with its own directory name) within a directory called Projects. You can use this script for future assignments in this course as well as other courses.

BONUS: (for no extra points) you may want the script to divide projects by course number or some other designator (we saw this in mini1 where Projects and then COMP206 was created before creating the directories). This is not required in this assignment, but if you want to add it, then change the command-line arguments to:

```
mkproj ARGUMENT SUB OWNER
```

Where SUB is the subdirectory (like COMP206) where all the directories will be created under Projects. Therefore: Projects/SUB/ARGUMENT/ all the directories for the project. **Make sure to tell your TA that you have done this.**

QUESTION 2: A Backup Script

Write a script called: backup SWITCH FILES DEST MESSAGE

Example: ./backup -D ~/source/*.c ~/backup Initial backup example

Where:

- SWITCH is optional: -D for delete original files. If not present the files to be backed-up are not deleted.
- FILES is the path, using wild cards, of the files to be backed-up.
- DEST is the path to the backup directory.
- MESSAGE is a multiword description about the backup. It will be added to a log file.

Use vi to create the script.

The script must do the following in the order presented:

1. Make sure the script is running under Bash.
2. Verify that the script has the correct number of arguments. If it does not the script must terminate with the following error message: "You have the incorrect number of arguments. The correct syntax is: ./backup SWITCH FILES DEST MESSAGE. Please try again."
3. Verify that the DEST directory exists. If it does exist, then do nothing. If it does not exist, then create the directory.
4. Using the TAR command, backup all the FILES to DEST. Call the backup file name: "backupDATE.tar". Where DATE is today's date (not time).
5. CHMOD the tar file as RW for only the owner.
6. Append to a logfile called "backups.log" the MESSAGE preceded by today's date and time. This logfile is within the backup directory. The format of this log message is: DATE TIME MESSAGE. It is all on one line. Each time the backup script is called a new log is appended to this file.

You must use **mimi.cs.mcgill.ca** to create the solutions to this assignment. You cannot use your Mac command-line, Windows command-line, nor a Linux distro installed locally on your laptop. You can `ssh` or `putty` from your laptop to **mimi**, or you can go to the third floor of Trotter and use any of those labs, to complete this assignment.

WHAT TO HAND IN

Everything must be submitted to My Courses before the due date. Remember that you can hand in your assignment up to two days late but there will be a penalty of 5% each day. After that, your assignment will not be accepted. Please hand in the following:

- The bash script file: `mkproj`
- The bash script file: `backup`
- The above two files do not have a file extension.
- Make sure to add comments to your script
- **Add you name and student ID number as a comment.**

HOW IT WILL BE GRADED

THE TESTING SCRIPT

The TA will run a testing script when evaluating your assignment. This test script is included in the assignment. You can use this testing script as you write your solution to verify that your program runs correctly. The script file is called `mini2tester.sh`.

POINTS AWARDED

The assignment is worth a total of 20 points.

- 10 points for Question 1
 - 2 point – following script instructions
 - 1 point – sha-bang
 - 2 point – testing number of arguments
 - 1 point – creating directories
 - 2 point – testing project directory
 - 2 point – create the `readme.txt` file
- 10 points for Question 2
 - 2 points – following script instructions
 - 2 points – testing for variable number of arguments
 - 2 points – the `TAR` command
 - 2 points - `CHMOD` command
 - 2 points - Writing the log file

GRADING RULES

The following rules are followed by the TA when grading assignments:

- A program must run in order to get a grade (even if it does not run well). If it does not run (does not compile) it will receive a zero. (Make sure to run your programs from `mimi.cs.mcgill.ca`).
- The TA will grade using the `mimi.cs.mcgill.ca` server.
- All questions are graded proportionally. This means that if 40% of the question is correct, you will receive 40% of the grade.