# CS 279 - Homework 2

## Deadline:

Due by 11:59 pm on **FRIDAY**, September 5.

## How to submit:

Submit your files using `~ah270/279submit` on nrs-labs, with a homework number of `2`, by the deadline shown above.

## Purpose

To practice some basic UNIX/Linux commands and to practice writing a small `bash` shell script

## Important notes:

- If I don't specify which mode -- octal or symbolic -- to use in specifying a file's permissions, then you may use either mode, your choice.

## The Problems:

### *Problem 1:*

You will practice with some common UNIX/Linux commands in the context of a simple `bash` shell script.

Write a `bash` shell script named `backup.sh` that meets the following specifications:

- include a descriptive comment at its beginning that also includes your name and the last-modified date
- it should create a new directory named `BACKUP` in the current working directory
  - (it is OK that you'll get a complaint if this directory happens to already exist -- we haven't yet covered the shell programming features needed to prevent that...)
- it should set `BACKUP`'s permissions so that the owner/user has read, write, and execute permissions on it, but the group and the world/other have no permissions
- it should copy all of the non-directory files in the current working directory into the directory `BACKUP`
  - (and it is OK if it prints a message complaining about being unable to copy `BACKUP` itself over -- likewise, it is OK if it complains about being unable to copy over any other directory files that happen to be in the current working directory)
- it should echo to the screen a descriptive message indicating that it is about to show the current

contents of `BACKUP`,

- ...and then it should output to the screen the current contents of `BACKUP`.

Also perform at least the following test of `backup.sh`:

- do this test within a directory containing at least 3 non-directory files
- list the current contents of this directory, redirecting the results into a file `backup-test.txt`
- then run `backup.sh` in this directory, appending the results to the file `backup-test.txt`

Submit your resulting `backup.sh` and `backup-test.txt` .

## *Problem 2:*

Run the `history` command -- see how it shows a listing of commands that you have done.

It turns out that following `history` with an integer <number> results in your seeing the last <number> of commands that you have done -- that is,

```
history 3
```

...shows just your last 3 commands.

You'll use this to create part of your output for this problem.

- Make a directory `279prob2`, and set its permissions so that you (the owner/user) has read, write, and execute permissions on it, but the group and the world/other have no permissions
- Demonstrate `279prob2`'s permissions with the following command:

  ```
  ls -ld 279prob2 > 279prob2/prob2-perms.txt
  ```

  - (you use the `-d` option of `ls` when you want to see the name of the directory, not a listing of its contents. Using `-ld` lets us get the directory's permissions, not the permissions of each file within that directory.)
- Change to `279prob2` (make it your current working directory).
- Create a file named `prob2play` within `279prob2`, that contains any contents you would like.
- Set `prob2play`'s permissions so that you (the owner/user), the group, and the world/others have read and write permissions only.
- Demonstrate `prob2play`'s permissions within the following command:

  ```
  ls -l prob2play >> prob2-perms.txt
  ```
- Now change `prob2play`'s permissions so that you (the owner/user), the group, and the world/others each have a different set of permissions of your choice (any set is fine, as long as each level has a *different* set)
- Demonstrate `prob2play`'s modified permissions within the following command:

  ```
  ls -l prob2play >> prob2-perms.txt
  ```
- Now change `prob2play`'s permissions such that you are **adding** the same permission to all 3 levels

(it is OK if one of the levels already had that permission, have your command attempt to add it, anyway)

- Demonstrate `prob2play`'s modified permissions within the following command:

```
ls -l prob2play >> prob2-perms.txt
```

- Now change `prob2play`'s permissions such that you are **removing** the same permission from all 3 levels (it is OK if one of the levels didn't have that permission, have your command attempt to remove it, anyway)
- Demonstrate `prob2play`'s modified permissions within the following command:

```
ls -l prob2play >> prob2-perms.txt
```

- You've done at least 13 UNIX commands at this point (and possibly more). Use the `history` command to figure out how many commands it has been since the command creating the directory `279prob2` (remember to count your `history` commands(s) as you determine this)!

  Then call `history` with an appropriate number argument, redirecting the output into `prob2-commands.txt`, such that all of your commands done for this problem, since the command creating the directory `279prob2`, are saved into `prob2-commands.txt`.

Submit your resulting `prob2-perms.txt` and `prob2-commands.txt` .

## *Problem 3:*

Soon, the collected answers to Homework 1 will be posted to the course Moodle site.

Consider the responses to the latter part of Homework 1 Problem 4, in which you gave at least one command for which you learned something interesting using `man`, and what you learned about that command. Note that each of these has an ID number.

Find one of these (besides your own! 8-) ) which you would like to try out, and create a small `bash` shell script named `prob3-play.sh` that meets the following specifications:

- include a descriptive comment at its beginning that also includes your name and the last-modified date
- include another comment that gives the ID number of the posted Problem 4 response you are going to try out, and why you chose it
- echo to the screen a descriptive message describing what you are about to try
- then, put the command trying that out

Try out your script until you are satisfied with it.

Submit your resulting `prob3-play.sh` .