C CS 2XA3/SE 2XA3 (2015/16, Term I) Proj 1 -- lab section L03

Back To Lab Menu | Back To Main Menu | Submissions | Log Out

Sample solution

In this project, there is just one deliverable, i.e. one file to be created and submitted either via the Submission button at the top of this page, or using the `2xa3submit` command (see Lab 1). The file contains a bash script and must be called `proj1`. The description of what this script is supposed to do and contain is given below.

What should the bash script `proj1` do:

1. Displays `Creating SAVE` and creates in the current directory a subdirectory named `SAVE`.
2. It sets the permissions for the directory `SAVE` to readable, writeable, and executable (searchable) by user only.
3. Displays `Content of current directory with permissions` and displays all items in the current directory so that the permission shows.
4. In the current directory it creates 10 ASCII text files named `spam1`, ..., `spam10`.. The content of `spam1` consists of two lines: the first saying `spam1` and the second saying `Next file is spam2`. The content of `spam2` consists of two lines `spam2` and `Next file is spam3` etc. Note that the last file `spam10` contains two lines, the first saying `spam10` and the second saying `No next file`.
   (*this part ought to be done by a loop, not by 10 separate commands*)
5. Displays `Regular files in current directory` and then displays names of all regular files in the current directory.
6. Displays `Regular files of SAVE` and then displays names of all regular files in `SAVE`.
7. Then it moves all `spam` files containing a character `2` or a character `3` or a character `5` from the current directory to `SAVE`.
8. Displays `Regular files of SAVE` and then displays names of all regular files in `SAVE`.
9. Displays `Regular files in current directory` and then displays names of all regular files in the current directory.
10. Then it creates a single file in the current directory named `BIGFILE` that contains all the lines containing a character 7 from all the `spam*` files in the current directory.
    (** this was omitted on the day of project, so both approaches are accepted as correct -- if you displayed or files or if you displayed just the* spam *files*)
11. Displays `The content of BIGFILE` and then displays all the lines of `BIGFILE`.
12. Displays `Regular files in current directory with permissions` and then displays names of all regular files in the current directory so that permissions show.
13. Then it changes the permissions for `BIGFILE` so that it is readable by all, executable by owner+group and wrieable by owner only.
14. Displays `Regular files in current directory with permissions` and then displays names of all regular files in the current directory so that permissions show.
15. Then it removes the directory `SAVE` with all its contents, and from the current directory all the created files (i.e. BIGFILE, spam..). Now the current directory should contain exactly the same files and subdirectories as just before this script was executed.

What commands you might need: `cd mkdir echo` for loop `> >> ls grep mv cat chmod`

A few useful hints:

- current directory is referred to as `.` , the parent directory as `..`
  For instance, `ls .` will show all files/subdirectories in the current directory, while `ls ..` will show all files/subdirectories in the parent directory
- a range from 1 to 15 can be expressed as `{1..10}`, for instance
  `for i in {1..10}`
- to concatenate a string with a number (`x` contains a string, `i` contains a number), use `$x$i`
- Substring notation -- if a variable `x` contains a string, then `${x:e1:e2}` is the substring of `x` consisting from position `e1` to position `e2`. The first letter of a string has position 0. Thus, if `$x` is `abcdef`, then `${x:0:5}` is `abcdef`, while `${x:2:3}` is `cd`, `${x:1:3}` is `bcd`, etc.
- to increment a variable `i` containing a number, use `i = $(($i+1))`
- to test if a name stored in a variable `f` is a name of a regular file, use `[ -f = $f ]`
- to extract into fout lines from a file fin that conation a symbol `2`, use `grep '2\|3' $f` ` and then test whether `x` is empty (note the ' quotes inside and the ` quotes around the whole expression).
- In Bash you quite often need to check to see if a variable has been set or has a value other than an empty string. This can be done using the `-n` or `-z` string comparison operators.
  The `-n` operator checks whether the string is not null. Effectively, this will return true for every case except where the string contains no characters. ie:

  ```
  VAR="hello"
  if [ -n "$VAR" ]
  then
      echo "VAR is not empty"
  fi
  ```

  Similarly, the `-z` operator checks whether the string is null. ie:

  ```
  VAR=""
  if [ -z "$VAR" ]
  then
      echo "VAR is empty"
  fi
  ```

  Note the spaces around the square brackets. Bash will complain if the spaces are not there.

When executed, the script should give an output similar to this:

```
Creating SAVE
Content of current directory with permissions:
total 20
drwxr-xr-x 3 franek faculty 4096 Sep  2 12:37 .
drwx------ 6 franek faculty 4096 Sep  2 11:25 ..
-rw-r--r-- 1 franek faculty   61 Sep  2 12:37 ooo
-rwxr--r-- 1 franek faculty 2115 Sep  2 12:37 proj1
drwxr-xr-x 2 franek faculty 4096 Sep  2 12:37 SAVE
Regular files in current directory:
proj1
spam1
spam10
spam2
spam3
spam4
spam5
spam6
spam7
```

```
spam8
spam9
Regular files of SAVE:
Regular files of SAVE:
spam1
spam2
spam3
spam4
spam5
Regular files in current directory:
proj1
spam10
spam6
spam7
spam8
spam9
The content of BIGFILE:
Next file is spam7
spam7
Regular files in current directory with permissions:
-rw-r--r-- 1 franek faculty 25 Sep  2 12:37 BIGFILE
-rwxr--r-- 1 franek faculty 2115 Sep  2 12:37 proj1
-rw-r--r-- 1 franek faculty 20 Sep  2 12:37 spam10
-rw-r--r-- 1 franek faculty 25 Sep  2 12:37 spam6
-rw-r--r-- 1 franek faculty 25 Sep  2 12:37 spam7
-rw-r--r-- 1 franek faculty 25 Sep  2 12:37 spam8
-rw-r--r-- 1 franek faculty 26 Sep  2 12:37 spam9
Regular files in current directory with permissions:
-rwxrw-r-- 1 franek faculty 25 Sep  2 12:37 BIGFILE
-rwxr--r-- 1 franek faculty 2115 Sep  2 12:37 proj1
-rw-r--r-- 1 franek faculty 20 Sep  2 12:37 spam10
-rw-r--r-- 1 franek faculty 25 Sep  2 12:37 spam6
-rw-r--r-- 1 franek faculty 25 Sep  2 12:37 spam7
-rw-r--r-- 1 franek faculty 25 Sep  2 12:37 spam8
-rw-r--r-- 1 franek faculty 26 Sep  2 12:37 spam9
```