SE 2XA3 (2019/20, Term I) Minor Lab 2 -- lab section L03

Back To Lab Menu Back To Main Menu Log Out

If you are in the correct lab room and during the SCHEDULED time slot for your lab section for Minor Lab 2 and working on one of the lab's workstations, go into Marks+Comments to see if your attendance was registered. If you signed on the course website too early (the most common problem), your attendance would not be recorded; in such case log out and sign on again. If your attendance is still not recorded, ask your TA to record your attendance and email Prof. Franek right away. If you are not using one of the lab's workstations and want to have your attendance recorded, tell the TA.

The Minor Labs are open book labs, you can bring and use any notes etc., you can access any website, Google, Wikipedia etc. You are allowed to cooperate with other people, inside or outside the lab. The TA can provide extensive individual help as long as this would not deprive other students of the TA's help.

More advanced Bash scripts

Create a bash script named `script1`

The script creates in the current directory a subdirectory named **XXX** and in this subdirectory a few files. The contents of the files are not important (in the sample solution provided, three files named **file1**, **fi1e2**, and **file3** are created, each containing just 1 empty byte). The script then checks all files in the current directory and all files in its subdirectories and displays their names. If a file is in a subdirectory, the file's name is followed by the name of the subdirectory in parenthesis. After that, the subdirectory **XXX** and all its files are removed.
Hints: *use* ***find .*** *to list all files, have a look how the output looks like -- it gives the relative pathname for each file. Use the command* ***dirname <pathname>*** *to find the directory of each file. If the directory name is dot, it means that the file is in the current directory, so just display it as the plain file using the command* ***basename <pathname>*** *. If it is not dot, remember the name of the directory and strip* ***./*** *from the beginning of the pathname, then display the name of the file using* **basename <pathname>** and then the remembered directory.

A sample run, the current directory contains just five files named **script1**, **script2**, **script3**, **script4**, and **script5**:

```
script1
script3
script4
script5
script2
script1
file1 (XXX)
file2 (XXX)
file3 (XXX)
```

see a sample solution for the above script

Create a bash script named `script2`

Write a bash script that creates in a loop 10 different files in the current directory named `lab2_file_1`, `lab2_file_2`, ..., `lab2_file_10`. The file `lab2_file_1` contains a single line saying `123`, file `lab2_file_2` contains a single line saying `234`, ..., file `lab2_file_10` contains a single line saying `101112`. Then it gets names of all files in the directory and in a loop it displays the names and contents of the created files (ignoring all other files that there may be). Then it removes all the created files, leaving all other files intact. Should produce an output similar to the sample run below.

Hints: *read section 3.1.2 of the bash manual (found in Help) to understand the use of* **`'`** *(single quote) and* **`"`** *(double quote). Read section 3.5.5 to understand how to do arithmetic, in particular expressions like* **`$(($i+1))`** *and section 3.5.4 to understand how to use* **`` ` ``** *(back quote). Read section 6.4 to learn about bash conditional expressions.*

A sample run:

```
lab2_file_1:
123
lab2_file_10:
101112
lab2_file_2:
234
lab2_file_3:
345
lab2_file_4:
456
lab2_file_5:
567
lab2_file_6:
678
lab2_file_7:
789
lab2_file_8:
8910
lab2_file_9:
91011
```

see a sample solution for the above script

Create a bash script named `script3`

Write a bash script that creates in the current directory a file named `s1` . The file `s1` contains a single line saying `I am s1` . Then it creates a file named `s1a`. The file `s1a` contains a single line saying `I am s1a` . Then it creates a file named `s2`. The file `s2` contains a single line saying `I am s2` . Then it creates a file named `s2a`. The file `s2a` contains a single line saying `I am s2a` . Then it creates a file named `s3`. The file `s3` contains a single line saying `I am s3` . Then in a loop it tests which of the files `s1`, `s1a`, `s2`, `s2a`, and `s3` exist in the current directory. It is achieved by having 5 variables `x1`, `x1a`, `x2`, `x2a`, and `x3` initialized to 0. Then the script goes through the list of files checking whether the file is one of the files `s1`, `s1a`, `s2`, `s2a`, and `s3`, and if so, the corresponding variable is set to 1. Then the script goes through the variables and based on their values, it displays a message like `s1 not found in current directory` or `s1` found in current directory. Then all the variables are summed up and if the sum is smaller than 5, a message `not all files found in current directory` is displayed, otherwise a message `all files found in current directory` is displayed. After, the script removes only the created files and terminates.

When the script performs well and it produces an output like the sample run below, block out the creation of one of the files `s1`, `s1a`, `s2`, `s2a`, *and* `s3` *and run the script again, you should get output similar to the second sample run below.*

A sample run when all files are created:

```
s1 found in current directory
s1a found in current directory
s2 found in current directory
s2a found in current directory
s3 found in current directory
all files found in current directory
```

A sample run when all files except **`s3`** are created:

```
s1 found in current directory
s1a found in current directory
s2 found in current directory
s2a found in current directory
s3 not found in current directory
not all files found in current directory
```

see a sample solution for the above script

Create a bash script named `script4`

This is an extension of `script3` . A subdirectory `XXX` in the current directory is created and the files `s1`, `s1a`, `s2`, `s2a`, and `s3` are moved to `XXX` while renamed to `t1`, `t1a`, `t2`, `t2a`, and `t3` . All the files in the current directory are checked to see if the files `s1`, `s1a`, `s2`, `s2a`, and `s3` disappeared from the current directory. For appropriate messages see the sample runs below.

When the script performs well and it produces an output like the sample run below, block out the moving of one of the files `s1`, `s1a`, `s2`, `s2a`, *and* `s3` *and run the script again, you should get output similar to the second sample run below.*

A sample run when all files are created:

```
s1 found in current directory
s1a found in current directory
s2 found in current directory
s2a found in current directory
s3 found in current directory
all files found in current directory
XXX created
all files moved to XXX
all files disappeared from current directory
```

A sample run when all files except **`s3`** are moved to **`XXX`** :

```
s1 found in current directory
s1a found in current directory
s2 found in current directory
s2a found in current directory
s3 found in current directory
all files found in current directory
XXX created
all files moved to XXX
not all files disappeared from current director
```

see a sample solution for the above script