SFWRENG 2XC3- Major lab 2 - Lab section L02

Rules for Major Labs:

1. For major labs, **students must participate in the registered lab section** and should not move between other lab sections. If you join another lab session, you will not be able to download the major lab questions. Your attendance will be taken during the lab, and your submission will only be marked if you attended the lab session, with the exception of previously-acquired special exemptions.

2. You can download the Major lab questions via AVENUE at the exact time of each lab session **(go to Assessments--> Assignment--> Major Lab).**

3. All major labs are open book; use of all resources printed, written, or from anywhere on the Internet is permitted, but using **ChatGPT** and any **AI-based language models** is not allowed.

4. You are not allowed to solicit help from other persons or use the work of other persons for even part of the solution, or discuss the solution with other persons. Specifically, you are not allowed to discuss the assignment problems with fellow students **in-person or online**. In simple terms, the entire solution must be your own work/findings.

5. You can submit your work (even partial work) as many times as you wish at any time, from the posting of the assignment to the assignment's deadline; all submissions are saved in the repository. For marking, the latest submission is used.

6. An integral part of the submission process of your work is for you to check the submission. **TAs** will no longer assist students in troubleshooting file transfer issues. This was covered during practice labs A & B. If you dislike/have issues with the suggested file-transfer applications, consider the command-line clients *scp* and *sftp*, which work across Mac, Windows, and Linux.

7. Files saved in your **Avenue account (Dropbox)** are considered submitted.

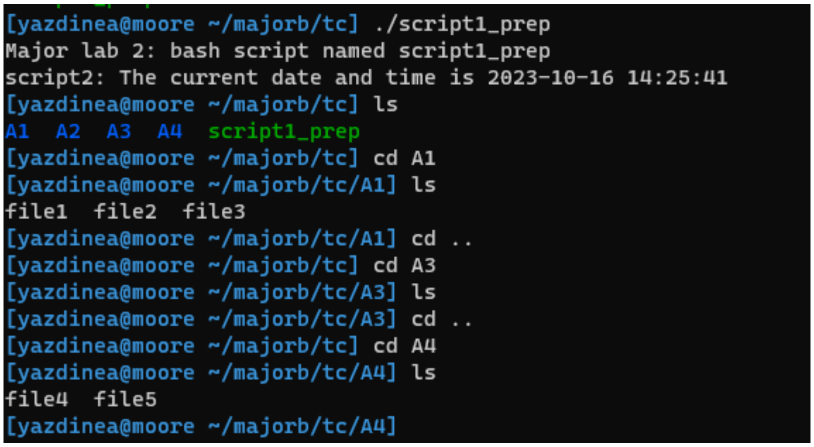
8. **Assignments will not be accepted after the specified lab session without previously-acquired special exception.** Submissions uploaded after the deadline will be disregarded. There will be no more leniency for future Major Labs. If submission is not possible for whatever reason, you could submit it to the TA and CC me, my email address: yazdinea@mcmaster.ca

Note: In this Major lab, you should submit 4 files, **script1-prep, script1**, **Makefile**, and **script2**, while providing a Word or PDF file to show the outputs.

Task 1. bash scripts named script1-prep and script1

Description of the script script1-prep (3 Marks)

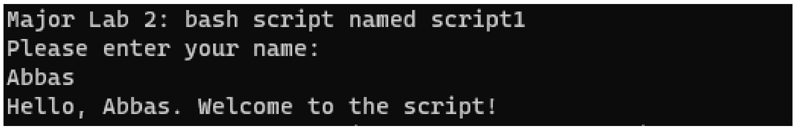
1. The script displays a message Major lab 2: bash script named script1-prep . The name of the script (script1-prep) must not be hard coded in the script, it has to be obtained via basename command using the command line argument $0 . A) . The script should also print the current date and time in the format "YYYY-MM-DD HH:MM:SS". Hit: You can consider this hint ($(date +"%Y-%m-%d %H:%M:%S")) to display.
2. The script then creates four directories named A1 , A2, A3 and a directory named A4 .
3. Then the script cd to the directory A1 and using echo command it creates there a file named file1 with a single line of any text you wish, a file named file2 with a single line of any text you wish, but different from the text of the line of file1 , and a file named file3 with a single line of any text different from the line for file1 and file2 .
4. Then the script cd to the directory A4 and using echo command it creates there a file named file4 with a single line of any text you wish, a file named file5 with a single line of any text you wish, but different from the text of the line of file4.
5. Then the script terminates.
6. After the script terminates, the directories A1 and A4 will contain their respective files, while the empty directories A2 and A3 will also remain.



```
[yazdinea@moore ~/majorb/tc] ./script1_prep
Major lab 2: bash script named script1_prep
script2: The current date and time is 2023-10-16 14:25:41
[yazdinea@moore ~/majorb/tc] ls
A1  A2  A3  A4  script1_prep
[yazdinea@moore ~/majorb/tc] cd A1
[yazdinea@moore ~/majorb/tc/A1] ls
file1  file2  file3
[yazdinea@moore ~/majorb/tc/A1] cd ..
[yazdinea@moore ~/majorb/tc] cd A3
[yazdinea@moore ~/majorb/tc/A3] ls
[yazdinea@moore ~/majorb/tc/A3] cd ..
[yazdinea@moore ~/majorb/tc] cd A4
[yazdinea@moore ~/majorb/tc/A4] ls
file4  file5
[yazdinea@moore ~/majorb/tc/A4]
```

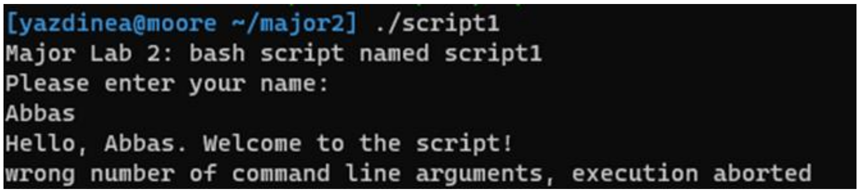
Description of the script script1(7 Marks)

1. The script displays a message Major lab 2: bash script named script1 . The name of the script (script1) must not be hard coded in the script, it has to be obtained via **basename** command using the command line argument $0 .
2. Then the script Prompt the user for the name (user enter XXX)**,** then print "Hello, XXX. Welcome to the script!"**.** Hint: the read command uses to get user_name. like below:



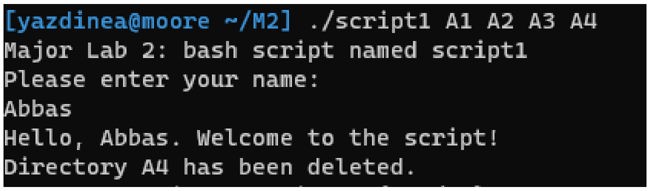
```
Major Lab 2: bash script named script1
Please enter your name:
Abbas
Hello, Abbas. Welcome to the script!
```

3. Then the script checks the number of command line arguments. If it is not 4, it displays an error message wrong number of command line arguments, execution aborted and terminates.



```
[yazdinea@moore ~/major2] ./script1
Major Lab 2: bash script named script1
Please enter your name:
Abbas
Hello, Abbas. Welcome to the script!
wrong number of command line arguments, execution aborted
```

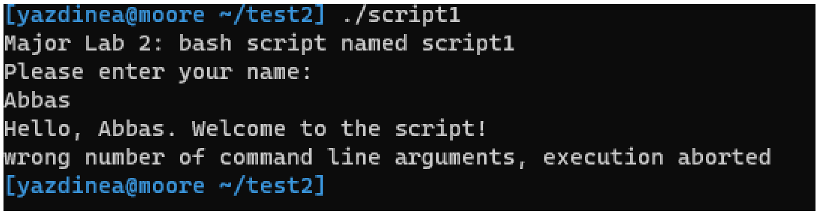
4. Check if directory A4 exists, Remove the directory A4 and all of its contents. Print the "Directory A4 has been deleted", otherwise: "Directory A4 does not exist."



```
[yazdinea@moore ~/M2] ./script1 A1 A2 A3 A4
Major Lab 2: bash script named script1
Please enter your name:
Abbas
Hello, Abbas. Welcome to the script!
Directory A4 has been deleted.
```

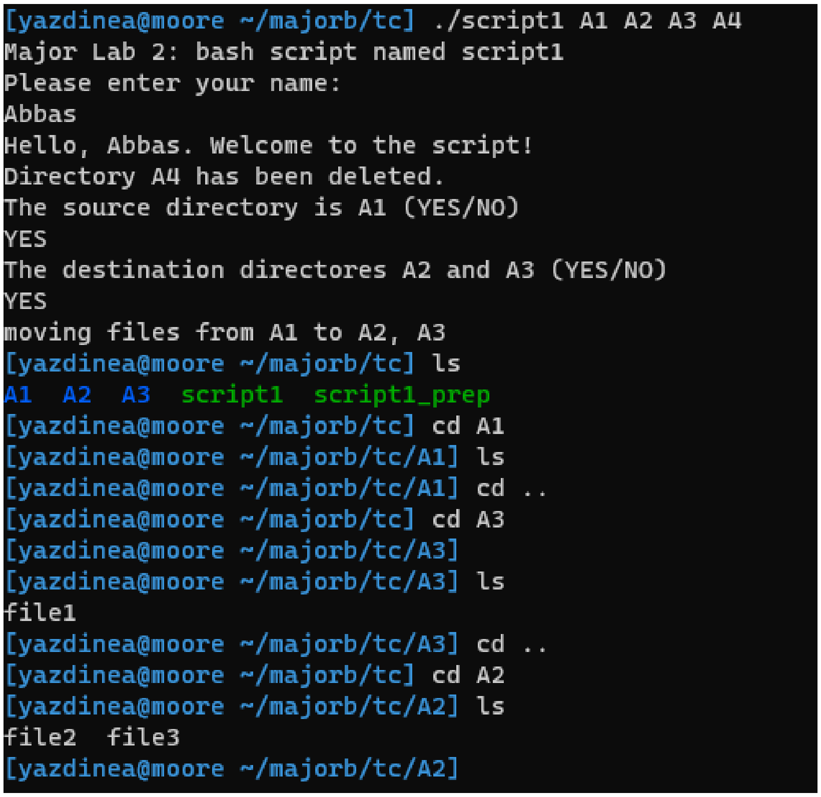
5. Then it displays a message The source directory is XXX answer (YES/NO) where XXX is the value of the first command line argument ($1).
6. It then reads the user's response. If the response is NO , it displays an error message you requested execution abortion and the script terminates. If the response is YES , it continues to step 5. Otherwise it displays an error message incorrect response, redo and the script repeats from step 5.
7. Then it displays a message The destination directories are YYY and ZZZ (YES/NO) where YYY is the value of the second command line argument ($2) and ZZZ is the value of the third command line argument ($3).
8. It then reads the user's response. If the response is NOT , it displays an error message you requested execution abortion and the script terminates. If the response is YES , it continues to step 7. Otherwise it displays an error message incorrect response, redo and the script repeats from step 5.
9. Then it displays a message moving files from XXX to YYY, ZZZ where XXX is the value of the first command line argument ($1), YYY is the value of the second command line argument ($2) and ZZZ is the value of the third command line argument ($3). Then it moves all files from the directory XXX that contain a letter A to the directory YYY and all the files from XXX that do not contain a letter A to the directory ZZZ .
10. Then the script terminates.

If you first run **script1-prep**, then you find 4 new directories A1 , A2, A3 and A4. The directory A1 contains three files file1 , file2 and file3, and A4 contains files file4 and file5. If also run **script1** without argaman, the output is like this:



```
[yazdinea@moore ~/test2] ./script1
Major Lab 2: bash script named script1
Please enter your name:
Abbas
Hello, Abbas. Welcome to the script!
wrong number of command line arguments, execution aborted
[yazdinea@moore ~/test2]
```

Then you run **script1 A1 A2 A3 A4** . After this you find out that the directory **A1** is empty and each of the files **file1 , file2** and **file3** is in one of directories **A2** or **A3** , depending on their contents while **A4** has removed.



```
[yazdinea@moore ~/majorb/tc] ./script1 A1 A2 A3 A4
Major Lab 2: bash script named script1
Please enter your name:
Abbas
Hello, Abbas. Welcome to the script!
Directory A4 has been deleted.
The source directory is A1 (YES/NO)
YES
The destination directores A2 and A3 (YES/NO)
YES
moving files from A1 to A2, A3
[yazdinea@moore ~/majorb/tc] ls
A1  A2  A3  script1  script1_prep
[yazdinea@moore ~/majorb/tc] cd A1
[yazdinea@moore ~/majorb/tc/A1] ls
[yazdinea@moore ~/majorb/tc/A1] cd ..
[yazdinea@moore ~/majorb/tc] cd A3
[yazdinea@moore ~/majorb/tc/A3]
[yazdinea@moore ~/majorb/tc/A3] ls
file1
[yazdinea@moore ~/majorb/tc/A3] cd ..
[yazdinea@moore ~/majorb/tc] cd A2
[yazdinea@moore ~/majorb/tc/A2] ls
file2  file3
[yazdinea@moore ~/majorb/tc/A2]
```

Hints:

- *In order to find if a file* xxx *contains a letter* A *, use the* grep *command and store the result in a variable* x *, i.e.* x=`grep A xxx` *. Then test if the string* x *is empty by* [-z "$x"] *. If it is empty, the file* xxx *does not contain* A *, if it is non-empty, then* xxx *contains* A *.*
- *To read a response of the user, you have to do it in a while loop and use* read *command and then test if the response is correct:*

```
cont=1
while [ $cont -eq 1 ]
do
  read x
  if [ "$x" == "YES" ]
  then
```

```
      cont=0
     else
      if [ "$x" == "NO" ]
      then
        display message about terminating
        exit 1
      else
        display message about redoing
      fi
     fi
done
```

Submit the script source code (script1-pre, and script1) and Word or PDF file which provides screenshots regarding the sample run script to show various cases as follows:

1- Run script1-prep
2- Run script1 without argaman
3- Run script1 A1 A2 A3 A4

Task 2. makefile scripts named makefile (3 Marks) and bash script script2 (6 Marks)

Hint: write and debug the makefile first

Description of the makefile (3 Marks)

1. The makefile expects that in the current directory there are two files, test.c and testA.c
2. The file testB.c is made from the file testA.c by replacing every occurrence of character A by B (use tr command for that).
3. The file testC.c is made from the file testA.c by replacing every occurrence of letter A by C (use tr command for that).
4. The file testA.o is made by simple compilation of testA.c , i.e. by gcc -c testA.c
5. The file testB.o is made by simple compilation of testB.c
6. The file testC.o is made by simple compilation of testC.c
7. The file Complete is made by compilation of test.c and linking it with testA.o , testB.o ,and testC.o, i.e. by gcc -o Complete test.c testA.o testB.o testC.o

Description of the bash script script2

Hint: temporarily put the command exit *after the code for creation of the file* test.c *to make sure that the file is created correctly. After successfully debugging the creation of* test.c,
remove exit *and put it temporarily after the code for creation of the file* testA.c.
Remove exit *completely once you are assured that* script2 *creates both files correctly.*

1. Using the echo command, the script creates a file test.c that contains the following lines:

   ```
   #include <stdio.h>
   extern int layA();
   extern int layB();
   extern int layC();
   int main() {
    printf("The Operating System has three layers: A, B, and C \n");
    printf("Layer A\n");
    layA();
    printf("Layer B\n");
    layB();
    printf("Layer C\n");
    layC();
    return 0;
    }
   ```

   Do not forget to escape the special characters \n and "

2. Using the echo command, the script creates a file testA.c that contains the following 5 lines:

   ```
   #include <stdio.h>
   int layA() {
    printf("Layer A: is running successfully \n");
    return 0;
   }
   ```

   Do not forget to escape the special characters \n and "

3. Then the script creates program (file) Complete by executing make Complete (which by default will use the makefile makefile)
4. Then the script executes the program Complete

5. Then the script cleans up by removing all the files it created,
   i.e. test.c, testA.c, testA.o, testB.c, testB.o, testC.c, testC.o, and Complete

Submit the script source code (makefile and script2) and Word or PDF file which provides screenshots regarding the sample run script.

A sample run of the script script2 (the messages from the execution of makefile were redacted not to reveal the contents of makefile)

```
XXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXX
The Operating System has three layers: A, B, and C
Layer A
Layer A: is running successfully
Layer B
Layer B: is running successfully
Layer C
Layer C: is running successfully
```

Good Luck!