Exercise 20: Writing Scripts - Part 1

I. Prepare the environment

II. Understanding variables and shell script arguments

1. Login to the CentOS server with student
2. Write the hello.sh script, that running on bash shell, to print the “Hello World!” string to the screen. Run it after you completed.
3. Write another script to display some information of the user running the script as the following format:
   User name: <user name>
   User ID: <user id>
   Shell: <Shell of user>
4. Write the shell script to display the user information of user whose user ID is passed to the script as an argument when executing. Display the output as the following format:
   User ID input: <user ID that is passed to script>
   The result looked up from /etc/password for the UID above:
   <User information got from /etc/password for that UID>
5. Write another script to allow input of 2 user ID and return the information of those 2 user with the following format:
   USER1 INFOMATION:
   <tab>User ID: <user id of the first user>
   <tab>User Name: <user name of the user>
   <tab>Home directory: <home dir of the user>
   <tab>Default shell: <default shell of the user>
   USER2 INFOMATION:
   <tab>User ID: <user id of the second user>
   <tab>User Name: <user name of the user>
   <tab>Home directory: <home dir of the user>
   <tab>Default shell: <default shell of the user>

Exercise Instructions

I. **Prepare the environment**

II. **Understanding variables and shell script arguments**

1. Login to the CentOS server with student
2. Write the hello.sh script, that running on bash shell, to print the “Hello World!” string to the screen. Run it after you completed.

   Script detailed:

   ```
   #!/bin/bash
   echo “Hello World!”
   ```

   Run the script

   ```
   $ <path to script>
   Or
   $ ./hello.sh
   ```

3. Write another script to display some information of the user running the script as the following format:
   User name: <user name>
   User ID: <user id>
   Shell: <Shell of user>

   Script detailed:

   ```
   #!/bin/bash
   echo “User name: $USER”
   echo “User ID: $UID”
   echo “Shell: $SHELL”
   ```

4. Write the shell script to display the user information of user whose user ID is passed to the script as an argument when executing. Display the output as the following format:
   User ID input: <user ID that is passed to script>
   The result looked up from /etc/password for the UID above:
   <User information got from /etc/password for that UID>

   Script detailed:

   ```
   #!/bin/bash
   echo “User ID input: $1”
   echo “The result looked up from /etc/password for the UID above:”
   grep $1 /etc/passwd
   ```

   Run the script

   ```
   $ ./<your script name> <UID of the user that you want to lookup>
   ```

5. Write another script to allow input of 2 user ID and return the information of those 2 user with the following format:
   USER1 INFOMATION:
   <tab>User ID: <user id of the first user>
   <tab>User Name: <user name of the user>
   <tab>Home directory: <home dir of the user>
   <tab>Default shell: <default shell of the user>
   USER2 INFOMATION:
   <tab>User ID: <user id of the second user>
   <tab>User Name: <user name of the user>
   <tab>Home directory: <home dir of the user>
   <tab>Default shell: <default shell of the user>

   Script detailed:

```
#!/bin/bash
echo “USER1 INFORMATION:”
echo -n “User ID:”
grep $1 /etc/passwd | cut -d: -f 3
echo -n “User Name:”
grep $1 /etc/passwd | cut -d: -f 1
echo -n “Home directory:”
grep $1 /etc/passwd | cut -d: -f 6
echo -n “Default shell:”
grep $1 /etc/passwd | cut -d: -f 7
echo “USER2 INFORMATION:”
echo -n “User ID:”
grep $2 /etc/passwd | cut -d: -f 3
echo -n “User Name:”
grep $2 /etc/passwd | cut -d: -f 1
echo -n “Home directory:”
grep $2 /etc/passwd | cut -d: -f 6
echo -n “Default shell:”
grep $2 /etc/passwd | cut -d: -f 7
```

   Run the script

```
$ ./<your script name> <UID of user 1> <UID of user 2>
```