Exercise 02 - Creating Local Users - 01

Goal:

The goal of this exercise is to create a shell script that adds users to the same Linux system as the script is executed on.

Scenario:

Imagine that you're working as a Linux System Administrator for a fast growing company. The latest company initiative requires you to build and deploy dozens of servers. You're falling behind schedule and are going to miss your deadline for these new server deployments because you are constantly being interrupted by the help desk calling you to create new Linux accounts for all the people in the company who have been recruited to test out the company's newest Linux-based application.

In order to meet your deadline and keep your sanity, you decide to write a shell script that will create new user accounts. Once you're done with the shell script you can put the help desk in charge of creating new accounts which will finally allow you to work uninterrupted and complete your server deployments on time.

Shell Script Requirements:

You think about what the shell script must do and how you would like it operate. You come up with the following list.

The script:

- Is named "`add-local-user.sh`".
- Enforces that it be executed with superuser (root) privileges. If the script is not executed with superuser privileges it will not attempt to create a user and return an exit status of 1.
- Prompts the person who executed the script to enter the username (login), the name for person who will be using the account, and the initial password for the account.
- Creates a new user on the local system with the input provided by the user.
- Informs the user if the account was not able to be created for some reason. It the account is not created, the script is to return and exit status of 1.
- Displays the username, password, and host where the account was created. This way the help desk staff can copy the output of the script in order to easily deliver the information to the new account holder.

After coming up with your list, you realize that you're not sure where to get the hostname information. You decide to wait until you start writing your shell script to check the bash man page to see if there are any builtin commands or variables that could provide this information.
You decide to develop your script on a Linux virtual machine running on your local operating system. This way you can test the script's functionality before uploading it to the server where it will be used by the help desk staff.

Create a Virtual Machine:

First, start a command line session on your local machine. Next, move into the working folder you created for this course.

```
cd shellclass
```

Initialize the vagrant project using the usual process of creating a directory, changing into that directory, and running "vagrant init". We'll name this vagrant project "localusers".

```
mkdir localusers
cd localusers
vagrant init jasonc/centos7
```

Configure the Virtual Machine

Edit the Vagrantfile and set the hostname of the virtual machine to "localusers".

```
config.vm.hostname = "localusers"
```

Start the Virtual Machine and Log into It

Now you're ready to start the VM and connect to it.

```
vagrant up
vagrant ssh
```

Navigate to the `/vagrant` Directory

```
cd /vagrant
```

Write the Shell Script

At this point you can either create the script inside the virtual machine using the `vim`, `nano`, or `emacs` text editors or you can create the file using your favorite text editor on your local operating system. (Atom from https://atom.io/ is a good choice.)

When creating your script refer back to the shell script requirements. If you want or need more detailed steps to help you write your script, refer to the pseudocode at the end of this document. It was intentionally placed at the end of the document because I want to encourage you to write the script on your own. It's fine if you need the pseudocode. As you get more scripting practice you'll be able to script without any additional aids.

Test Your Script

Once you've finished writing the script, test it by creating the following accounts:

- Username: einstein
  - Real Name: Albert Einstein
  - Password: E=mc2theory$
- Username: isaac
  - Real Name: Isaac Newton
  - Password: @ppleFell1666
- Username: tedison
  - Real Name: Thomas Edison
  - Password: Light.Bulb1

Remember that the first time you execute the script you'll need to make sure it has executable permissions.

```
chmod 755 add-local-user.sh
```

Here is an example run of the script. (Portions typed are in bold.)

```
sudo ./add-local-user.sh
Enter the username to create: einstein
Enter the name of the person or application that will be using this
account: Albert Einstein
Enter the password to use for the account: E=mc2theory$
Changing password for user einstein.
passwd: all authentication tokens updated successfully.
Expiring password for user einstein.
passwd: Success
```

```

username:
einstein

password:
E=mc2theory$

host:
localusers
```

Make sure the accounts have been created by examining the the last 3 lines of the `/etc/passwd` file.

```
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
... # additional accounts will be displayed
einstein:x:1001:1001:Albert Einstein:/home/einstein:/bin/bash
isaac:x:1002:1002:Isaac Newton:/home/isaac:/bin/bash
tedison:x:1003:1003:Thomas Edison:/home/tedison:/bin/bash
```

(NOTE: You could have also used `tail -3 /etc/passwd` to display just the last 3 lines of the file.)

Switch to the `einstein` user. Because the script forces a password change upon first login, create a new password for the `einstein` user. (Suggested password: "`CrazyHair!`")

```
su - einstein
```

Once you've changed the password of the user, exit out of the session to return to the vagrant user.

```
exit
```

Test to make sure that the script exits with a non-zero exit status if the user does not use superuser (root) privileges. (Portions typed are in bold.)

```
./add-local-user.sh
Please run with sudo or as root.
echo ${?}
1
```

Reference Material:

Vagrantfile for localusers

Here are the contents of the `shellclass/localusers/Vagrantfile` file with all the comments removed.

```
Vagrant.configure(2) do |config|
  config.vm.box = "jasonc/centos7"
  config.vm.hostname = "localusers"
end
```

Pseudocode

You can use the following pseudocode to help you with the logic and flow of your script.

```
# Make sure the script is being executed with superuser privileges.

# Get the username (login).

# Get the real name (contents for the description field).

# Get the password.

# Create the user with the password.

# Check to see if the useradd command succeeded.

# Set the password.

# Check to see if the passwd command succeeded.

# Force password change on first login.

# Display the username, password, and the host where the user was
created.
```