

Exercise 03 - Creating Local Users - Script 2

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:

The help desk team has been using the "add-local-user.sh" script you created. They're really happy that they don't have to wait on you to create new accounts. :-) However, they would like for you to make a couple of changes to the script when you get a chance.

They're tired of coming up with a unique password for each user they create. As a matter of fact, Jim keeps using "password" as the password for every account. They think it would be great if the script automatically generated a password for each new account. That way Jim and the rest of the team won't have to even think about passwords any longer.

Also, they think it would be a little more efficient if they could just specify the account name and account comments on the command line instead of being prompted for them. They already know what they are going to type so they would just rather type it all in at one time.

Since you're happy that you're not the one creating all the new accounts any longer, you decide to accommodate their requests. (You're so nice!)

Shell Script Requirements:

You have your requirements from the "add-local-user.sh" script you created. You decide to use those as the basis for your new requirements. You come up with the following list.

The script:

- Is named "add-new-local-user.sh". (You add the word new to distinguish it from the original script.)
- 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 returns an exit status of 1.
- Provides a usage statement much like you would find in a man page if the user does not supply an account name on the command line and returns an exit status of 1.
- Uses the first argument provided on the command line as the username for the account. Any remaining arguments on the command line will be treated as the comment for the account.

- Automatically generates a password for the new account.
- Informs the user if the account was not able to be created for some reason. If the account is not created, the script is to return an 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.

Start the Virtual Machine and Log into It:

In a previous exercise you created a vagrant project called localusers. Use the VM created in that project for this exercise.

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

```
cd shellclass
```

Chang into the localusers directory, start the virtual machine with "vagrant up", and then connect to it with "vagrant ssh".

```
cd localusers  
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.

NOTE: This script is very similar to the script you created in the previous exercise. You can use it as the starting point for this script and make the required changes or you can start from scratch to give you even more practice.

Test Your Script

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

- Username: jlocke
 - Real Name: John Locke
- Username: brussell
 - Real Name: Bertrand Russell
- Username: philapp
 - Account Description: Philosophy Application User

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

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

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

```
sudo ./add-new-local-user.sh jlocke John Locke
Changing password for user jlocke.
passwd: all authentication tokens updated successfully.
Expiring password for user jlocke.
passwd: Success

username:
jlocke

password:
8794dc4dfb47fe74bf71440da02743a267780a43d60a9054

host:
localusers
```

Make sure the accounts have been created by examining 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
jlocke:x:1004:1004:John Locke:/home/jlocke:/bin/bash
brussell:x:1005:1005:Bertrand Russell:/home/brussell:/bin/bash
philapp:x:1006:1006:Philosophy Application User:/home/philapp:/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 `jlocke` user. Because the script forces a password change upon first login, create a new password for the `jlocke` user. (Suggested password: "Theory-of-mind1632")

```
su - jlocke
```

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

Test to make sure that the script exits with a non-zero exit status if the user does not supply a username/ (Portions typed are in bold.)

```
sudo ./add-new-local-user.sh
Usage: ./add-new-local-user.sh USER_NAME [COMMENT]...
Create an account on the local system with the name of USER_NAME and a
comments field of COMMENT.
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.
# If the user doesn't supply at least one argument, then give them help.
# The first parameter is the user name.
# The rest of the parameters are for the account comments.
# Generate a 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.
```