

Activity: Examine input and output in the shell

Introduction

In this lab, you'll use the `echo` command to examine how input is received and how output is returned in the shell. You'll also use other Linux commands in the Bash shell to explore more about input and output and other basic functions of the shell.

What you'll do

You have multiple tasks in this lab:

- Generate output in the shell the `echo` command
- Perform basic calculations the `expr` command
- Clear the shell window the `clear` command
- Explore the commands further

Lab instructions

Start the lab

Before you start, you can review the [Resources for completing Linux labs](#). Then from this page, click **Launch App**. A Qwiklabs page will open and from that page, click **Start Lab** to begin the activity! *You may attempt this lab a maximum of 5 times, and you will have 60 minutes to complete this lab during each attempt.*

End the lab

From within the lab, click **End Lab** to end your lab.

Additionally, sometimes you need to refresh your Coursera page in order for your progress to be registered. If you refresh this page after you complete your lab, the green check mark should appear.

Best practices for completing labs:

- Make sure your browser is up to date with the latest version.
- Make sure your internet connection is stable.
- After you complete the lab, leave the lab window open for at least 10 minutes in order to allow the system to record your progress.
- If you run into issues connecting to the lab, try logging into Coursera in an Incognito mode and completing the lab there.

This course uses a third-party app, Activity: Examine input and output in the shell, to enhance your learning experience. The app will reference basic information like your name, email, and Coursera ID.

Activity: Examine input/output in the Linux shell

1 hourFree

Activity overview

Previously, you discussed how the Bash shell helps you communicate with a computer's operating system.

When you communicate with the shell, the commands in the shell can take input and return output or error messages.

In this lab activity, you'll use the `echo` command to examine how input is received and how output is returned in the shell. Next, you'll use the `expr` command to further explore input and output while performing some basic calculations in the shell.

This activity will build foundations in understanding how you communicate with the Linux operating system through the shell. As a security analyst, you'll need to input commands into the shell and recognize when the shell returns either output or an error message.

Next, you'll explore the scenario!

Scenario

As a security professional, it's important to understand the concept of communicating with your computer via the shell.

In this scenario, you have to input a specified string of text that you want the shell to return as output. You'll also need to input a few mathematical calculations so the OS (operating system) can return the result.

Here's how you'll do this: **First**, you'll use the `echo` command to generate some output in the shell. **Second**, you'll use the `expr` command to perform basic mathematical calculations. **Next**, you'll use the `clear` command to clear the Bash shell window. **Finally**, you'll have an opportunity to explore the `echo` and `expr` commands further.

Get ready to examine input and output in the Bash shell!

Note: The lab starts with your user account, called `analyst`, already logged in to the Bash shell. This means you can start with the tasks as soon as you click the **Start Lab** button.

Start your lab

Before you begin, you can review the instructions for using the Qwiklabs platform under the **Resources** tab in Coursera.

If you haven't already done so, click **Start Lab**. This brings up the terminal so that you can begin completing the tasks!

When you have completed all the tasks, refer to the **End your Lab** section that follows the tasks for information on how to end your lab.

Task 1. Generate output with the echo command

The echo command in the Bash shell outputs a specified string of text. In this task, you'll use the echo command to generate output in the Bash shell.

1. Type `echo hello` into the shell and press **ENTER**.

The hello string should be returned:

```
hello
```

The command `echo hello` is the **input** to the shell, and `hello` is the **output** from the shell.

2. Rerun the command, but include quotation marks around the string data. Type `echo "hello"` into the shell and press **ENTER**.

The `hello` string should be returned again:

```
hello
```

Note: The output is the same as before. The quotation marks are **optional** in this case, but they tell the shell to group a series of characters together. This can be useful if you need to pass a string that contains certain characters that might be otherwise misinterpreted by the command.

3. Use the `echo` command to output your name to the shell.

Type `echo "name"` into the shell, replacing `"name"` with your own name, and press **ENTER**.

The name you've entered as the string should return as the output.

Click **Check my progress** to verify that you have completed this task correctly.

Generate output with the `echo` command

Check my progress

Task 2. Generate output with the `expr` command

In this task, you'll use the `expr` command to generate some additional output in the Bash shell. The `expr` command performs basic mathematical calculations and can be useful when you need to quickly perform a calculation.

Imagine that the system has shown you that you have 32 alerts, but only 8 required action. You want to calculate how many alerts are false positives so that you can provide feedback to the team that configures the alerts.

To do this, you need to subtract the number of alerts that required action from the total number of alerts.

1. Calculate the number of false positives using the `expr` command.

Type `expr 32 - 8` into the shell and press **ENTER**.

The following result should be returned:

```
24
```

Note: The `expr` command requires that all terms and operators in an expression are separated by spaces. For example: `expr 32 - 8`, and **not** `expr 32-8`.

Now, you need to calculate the average number of login attempts that are expected over the course of a year. From the information you have, you know that an average of 3500 login attempts have been made each month so far this year.

So, you should be able to calculate the total number of logins expected in a year by multiplying 3500 by 12.

2. Type `expr 3500 * 12` into the shell and press **ENTER**.

The correct result should now be returned:

```
42000
```

Click **Check my progress** to verify that you have completed this task correctly.

Generate output with the `expr` command

Check my progress

Task 3. Clear the Bash shell

In this task, you'll use the `clear` command to clear the Bash shell of all existing output. This allows you to start with the cursor at the top of the Bash shell window.

When you work in a shell environment, the screen can fill with previous input and output data. This can make it difficult to process what you're working on. Clearing the screen allows you to create a clutter-free text environment to allow you to focus on what is important at that point in time.

- Type `clear` into the shell and press **ENTER**.

Note: All previous commands and output will be cleared, and the user prompt and cursor will return to the upper left of the shell window.

Click **Check my progress** to verify that you have completed this task correctly.

Clear the Bash shell

Check my progress

Optional task: Perform more calculations with the `expr` command

You have the opportunity to explore input and output further using the `echo` and `expr` commands.

1. Generate at least one new output using the `echo` command.

(Remember the `echo "hello"` output you generated).

2. Perform at least one new calculation using the `expr` command.

The mathematical operators you can use with the `expr` command for **adding**, **subtracting**, **dividing**, and **multiplying** are `+`, `-`, `/` and `*`.

Note: The `expr` command performs integer mathematical calculations only, so you cannot use the decimal point or expect a fractional result. All results are rounded down to the nearest integer. Also, all terms and operators in an expression need to be separated by spaces. For example: `expr 25 + 15`, and **not** `expr 25+15`.

Conclusion

Great work!

You now have practical experience in using basic Linux Bash shell commands to

- generate output with the `echo` command,
- generate output with the `expr` command, and
- clear the Bash shell with the `clear` command.

Understanding input and output is essential when communicating through the shell. It's important that you're comfortable with these basic concepts before you go on to work with additional commands.

End your lab

Before you end the lab, make sure you're satisfied that you've completed all the tasks, and follow these steps:

1. Click **End Lab**. A pop-up box will appear. Click **Submit** to confirm that you're done. Ending the lab will remove your access to the Bash shell. You won't be able to access the work you've completed in it again.
2. Another pop-up box will ask you to rate the lab and provide feedback comments. You can complete this if you choose to.

3. Close the browser tab containing the lab to return to your course.
4. Refresh the browser tab for the course to mark the lab as complete.