M7 Lab: Shell Scripting

CITA 171: OPERATING SYSTEM USE & ADMIN

Marco Gonzalez
SUNY CANTON | [COMPANY ADDRESS]

TABLE OF CONTENTS

1 TABLE OF FIGURES

2 OBJECTIVES

Learn what scripts are and how to create and execute them.

3 PREPARATION

Start the CITA 171 VM and log in. Start a Terminal program.

4 SCRIPT

A computer program is a list of instructions for a computer to perform tasks. A script is a type of computer program. Human-readable instructions are read, translated into computer-executable instructions, and executed by a computer program called an **interpreter** one instruction at a time.

5 SCRIPT PERMISSIONS

Because a script must be read and executed by the interpreter, the file **read** and **execute** permissions must be set for the user who wants to run the script.

6 SHEBANG

The first line of a Bash script must start with **#!/bin/bash**. The symbol combination #! Is called a **shebang**. This line shows the file path to the bash interpreter (shell program). An incorrect path may cause the script to fall.

7 CREATING A SIMPLE SCRIPT

The simplest way to write a script is to perform the task and capture the commands manually. The captured commands are then edited with a text editor to become a script. See Figure 1, Figure 2 , Figure 3 , and Figure 4. In this example, three commands are executed. The manually-typed commands are saved automatically into a file called **~/.bash_history** (notice the period at the beginning of the name). The **history** command with the **-c** option clears the previously captured commands. The three commands are then executed. The history command without options displays the contents of the ~/.bash_history file. The output is redirected to a script file called **MyFirstScript.sh**. The **.sh** is a common file extension for a Bash script. The nano command is used to edit the script file. Finally, the chmod command is used to set the proper permissions.

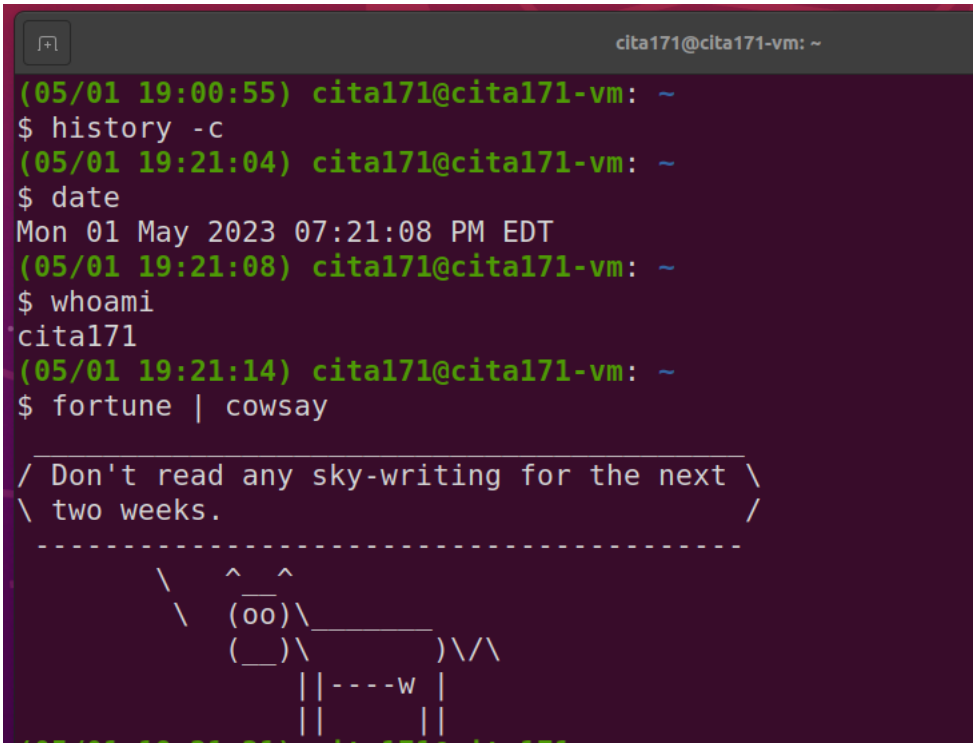

Figure 1. Clearing Command History and Manually Executing Commands

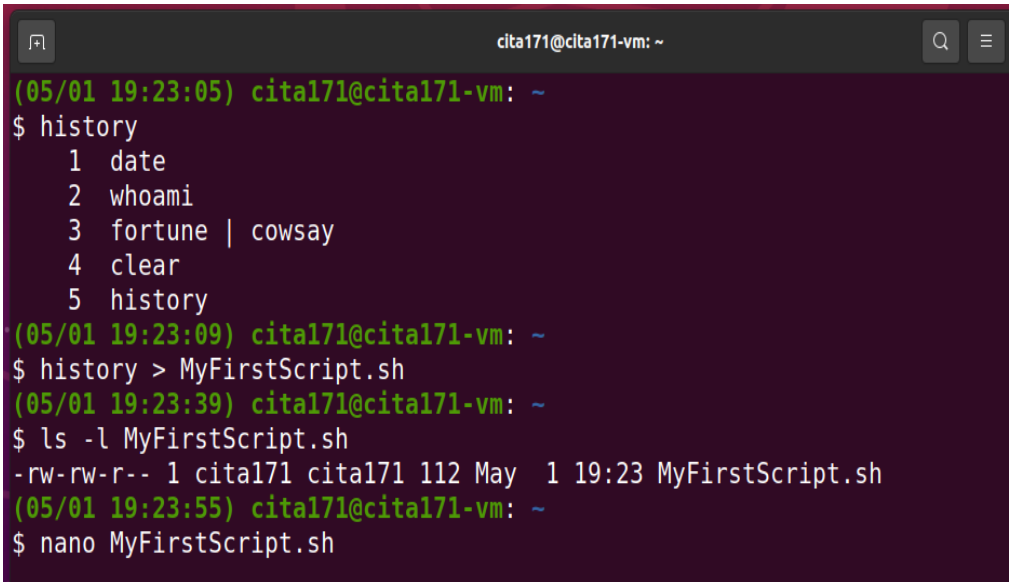

Figure 2. Saving and Editing a Script File

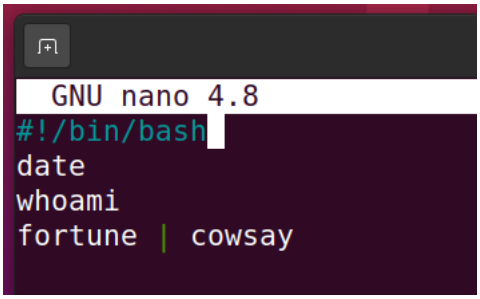

Figure 3. Edited Script

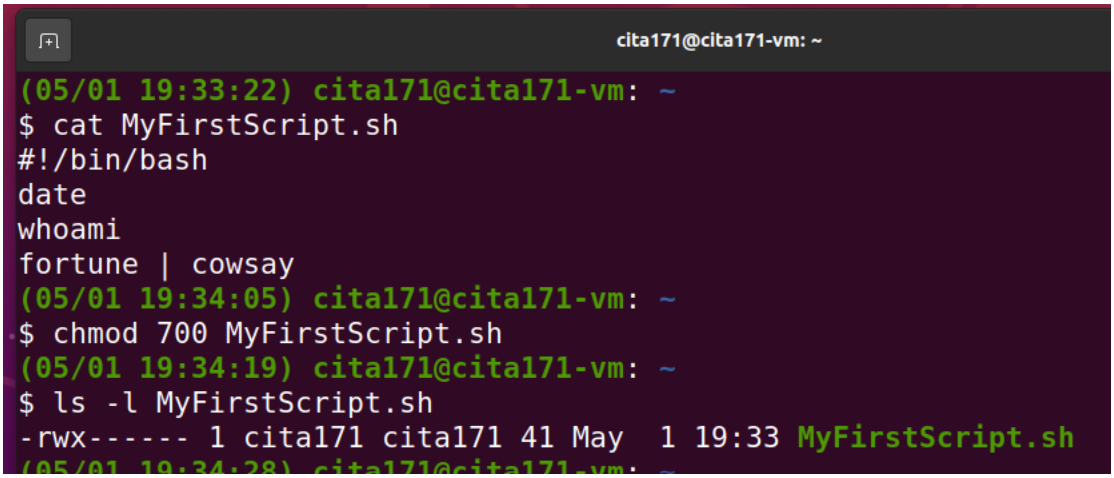

Figure 4. Setting the Proper Script Permissions

8 Executing a Script

There are two ways to execute a script. The **immediate execution** method executes the script immediately, whereas the **scheduled execution** method executes the script at a future time. A scheduled script is sometimes referred to as a **batch job**.

8.1 Immediate Execution

Immediate execution requires the path to the script file is used. The path can be absolute or relative. See Figure 5 and Figure 6.

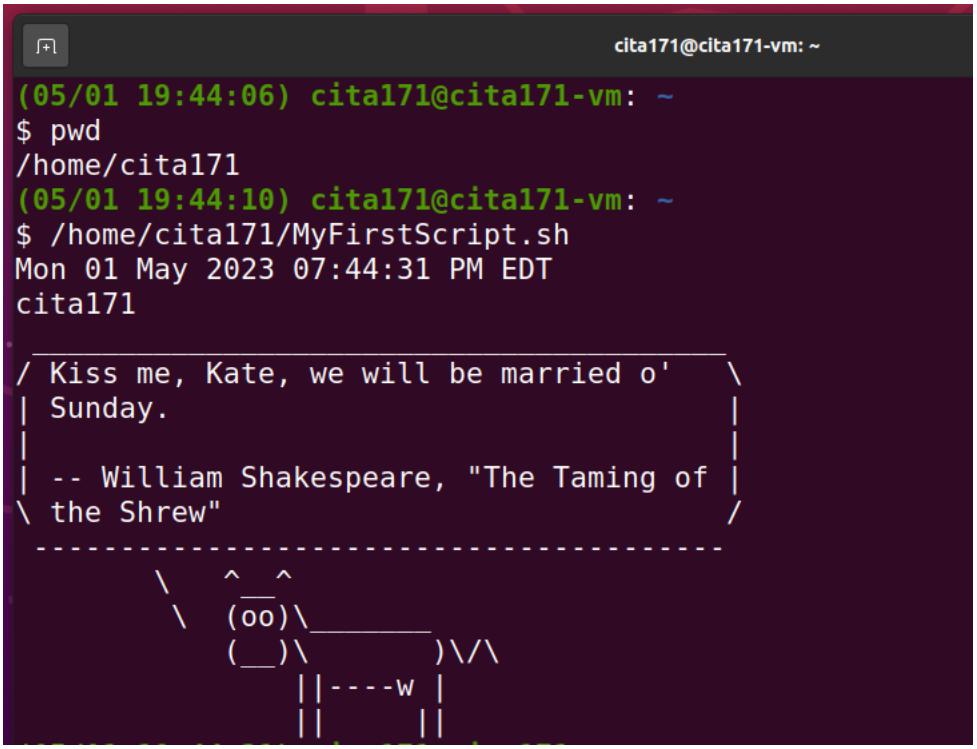

Figure 5. Immediate Execution using an Absolute Path

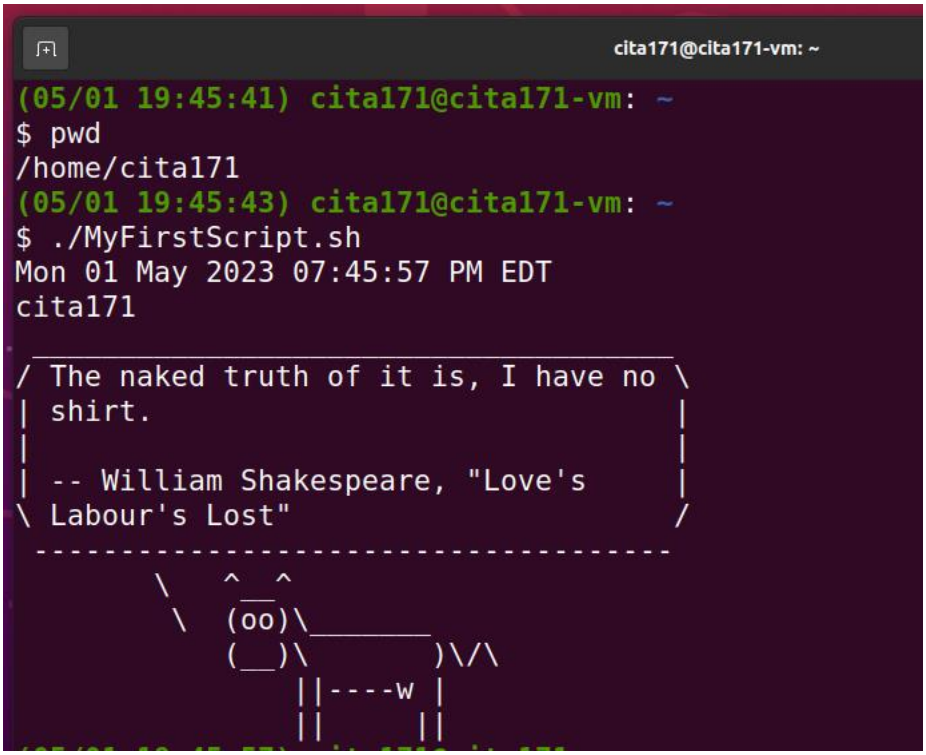

Figure 6. Immediate Execution using a Relative Path

8.2 SCHEDULED EXECUTION

A script can be executed at a future time using the **cron** or the **at** command. See Figure 7. In this example, the MyFirstScript.sh script is scheduled to run at 7:55 PM (Use two minutes from the current time) using the at command, and the result is saved in a file **Result.dat**.

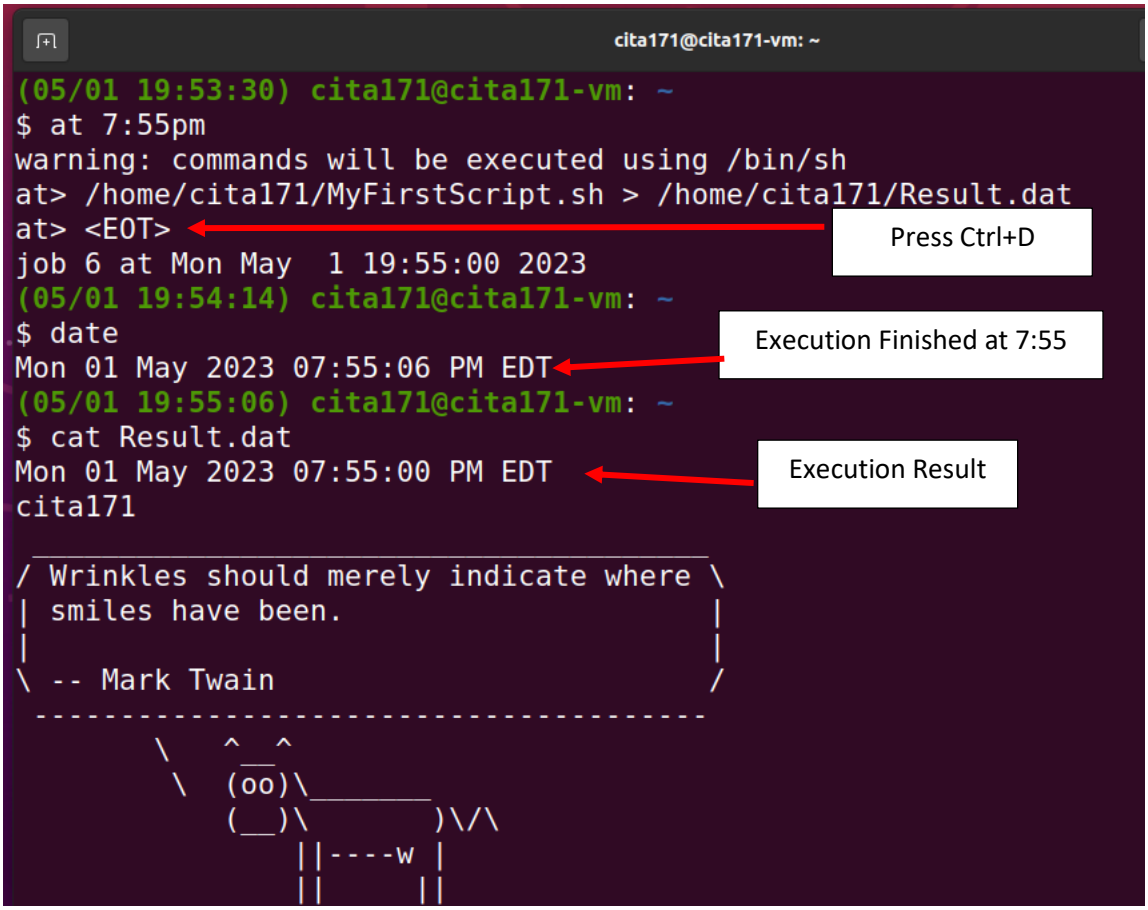

Figure 7. Scheduled Execution using the at Command