CS 2301-01 Operating Systems 1

Instructor: Professor Ravichandran Purushothaman

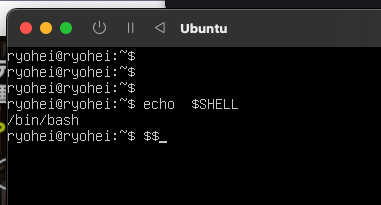
Name: Ryohei Hayashi

Learning Journal 8

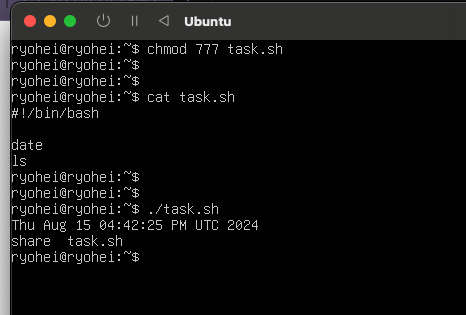
In this learning journal, I utilized the Bash scripting tutorial from LinuxConfig.org to learn the basics of Bash scripting. I will report on the content learned and challenges faced during the process. Specifically, I will cover script creation and execution, file names and permissions, the use of variables, and the definition and use of functions. Through this process, I deepened my foundational knowledge of Bash scripting, which I can apply in future shell script development.

1. What is Bash

I checked the definition of Bash scripting in my environment using the command echo $SHELL.



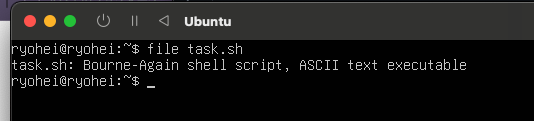
The result confirmed that the Bash script in my environment actually uses /bin/bash, and not other shells like the Korn shell or C shell. I also created a task.sh script and checked its execution result.



From this result, I confirmed that the outputs of the date command and the ls command were combined into one output as the result of executing task.sh.

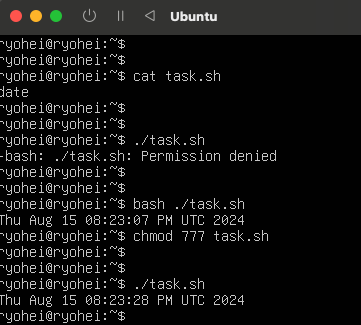
2. File Names and Permissions

To make task.sh executable, I used the command chmod 777 task.sh in the "What is Bash" section. This was necessary because files created by default cannot be executed regardless of the file extension, so it was necessary to make the file executable. Here, the file command can identify the type of a file. This allows you to check the content of the file rather than relying on the file extension.



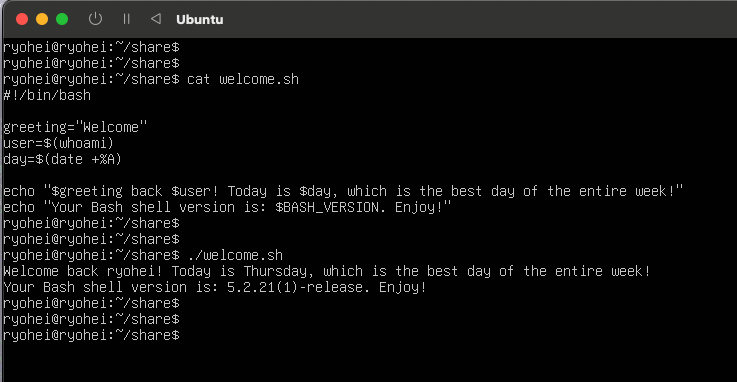
3. Script Execution

I executed the file using ./task.sh, which was possible because of the shebang included at the beginning of the file. Files without a shebang at the beginning cannot be executed, as shown in the following image.



In this example, I used two methods to execute the file: one by explicitly specifying the Bash interpreter, and the other by using the chmod command to make the file executable.

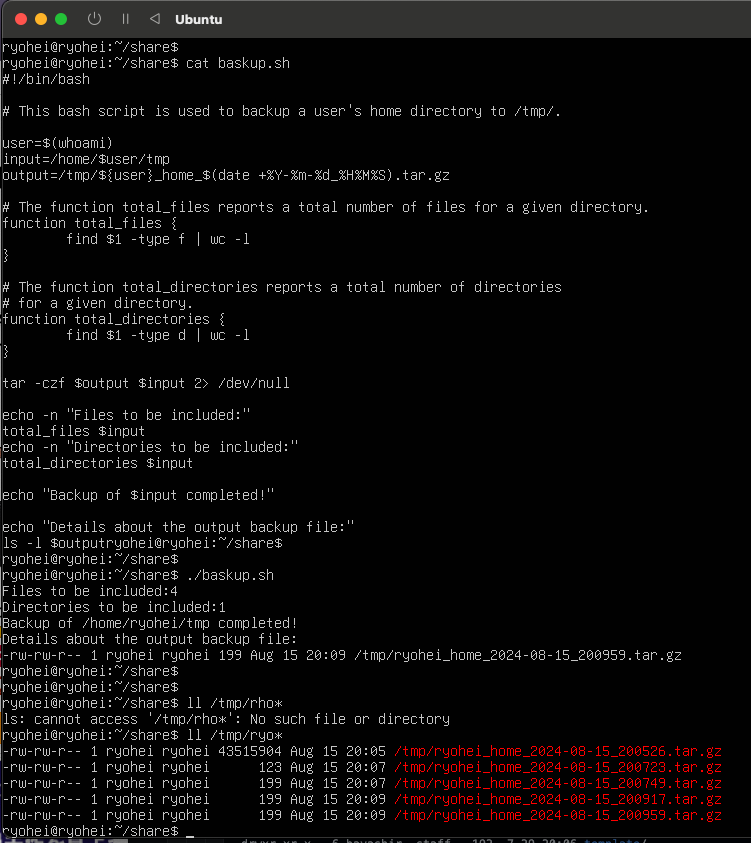
4. Variables

Variables are a very important element in programming. In the shell, variables can store the results of other commands or be used to save and modify data. I created a welcome.sh script and verified its operation.

In welcome.sh, three variables—greeting, user, and day—store the results of commands and strings. These stored strings are called within the echo command for use. Considering the use of these variables, I can explain them based on my own experiences. For example, when implementing processes that span across dates, if the result of the date command is not stored in a variable, the result of the date command at the beginning and end of the process can be completely different. This would prevent you from obtaining the desired result. In such cases, storing the result of the date command in advance as a variable allows you to obtain a consistent value regardless of the execution timing.

5. Functions

Functions enhance the readability of a script by grouping specific code for reuse. While it is possible to avoid using functions and write everything out explicitly, this often results in code that is difficult to read and maintain. In shell scripting, functions are defined using the function keyword, followed by the function body enclosed in curly braces. I created a shell script with a function to back up a specific directory and verified its operation.



In this shell script, I defined and used two functions, total\_files and total\_directories. Each function performs the following tasks:

total\_files: Retrieves the number of files in the specified directory during the function call.

total\_directories: Retrieves the number of directories in the specified directory during the function call.

Conclusion

Through this tutorial, I was able to understand the basics of Bash scripting and experience the flow from script creation to execution. In particular, setting file permissions, creating scripts using variables and functions, and seeing the flexibility and power of Bash scripting firsthand were valuable learning experiences. Moving forward, I feel confident that I can apply this knowledge to create more complex scripts and efficiently automate system management tasks.

Word Count: 648

References

1. Rendek, L. (2020, May 28). *Bash scripting tutorial for beginners.* LinuxConfig. https://linuxconfig.org/bash-scripting-tutorial-for-beginners