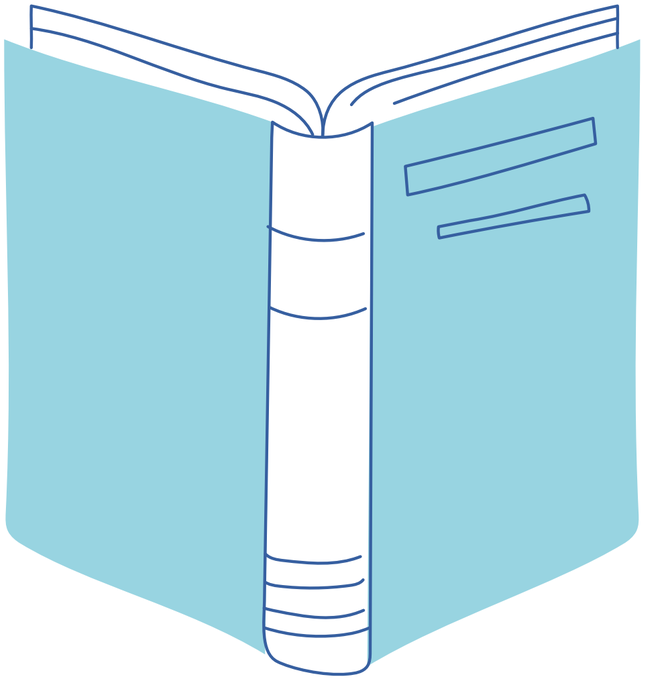
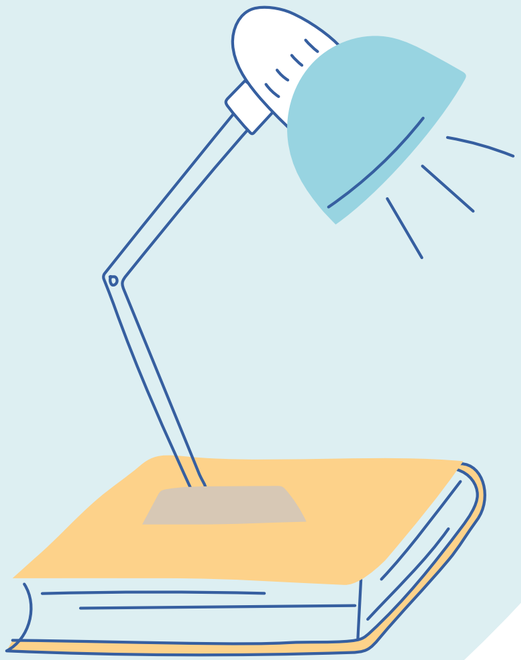


Assignment

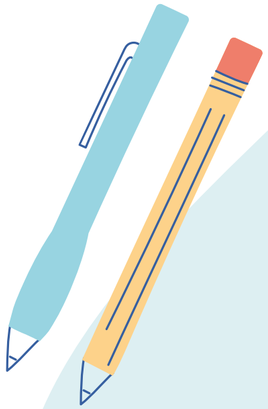
Lab07-OS

By Prak Pychey



Overview

- *Task 1: Answering Questions*
- *Task 2: Let's create your first script!*
- *Task 3: Let's make your script easier to execute and more advanced!*
- *Task 4: Let's have some fun with your previous script!*
- *Task 5: Write a script named check_number.sh*



Task 1

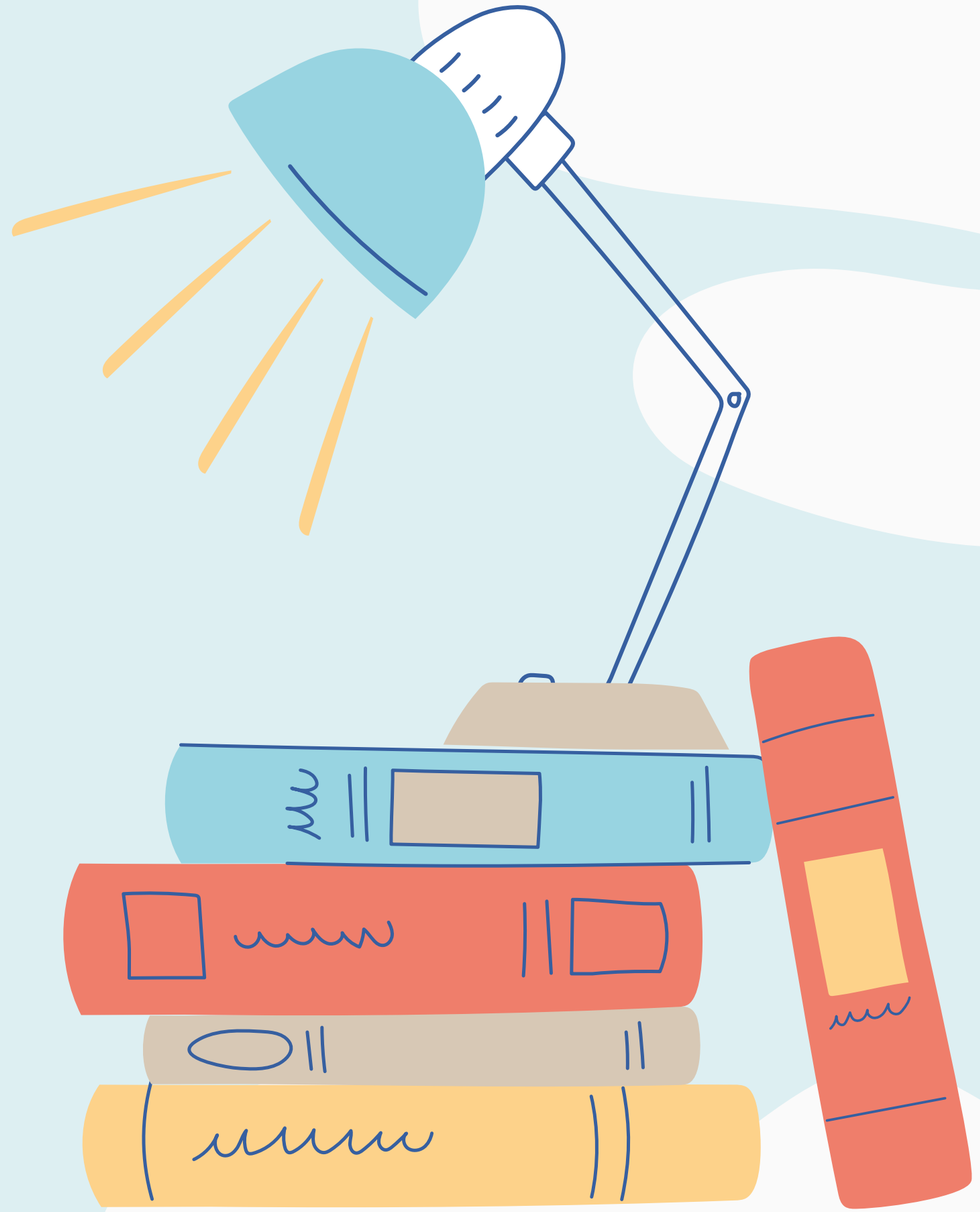
Practical Exercises

1. Read "**Part 4 - Writing Shell Scripts**" in "**The Linux Command Line 2nd Edition**" by William E. Shotts, Jr, and answer the following questions:
   a. By using your own words, what are shell scripts?
   b. What is the format or header of a script file?
   c. Explain the differences between *sh* and *bash*.
   d. Explain the differences between running script using *bash script.sh* and *./script.sh*.

Answer

a. Shell scripts are script files that contains shell commands (both basic command line and more advanced) which we can use to run the commands written in the file.
b. Format or header of a script file starts with shebang (#!) followed by path to the shell interpreter.
Example #!/bin/bash or #!/bin/sh.
c. The different between sh and bash:

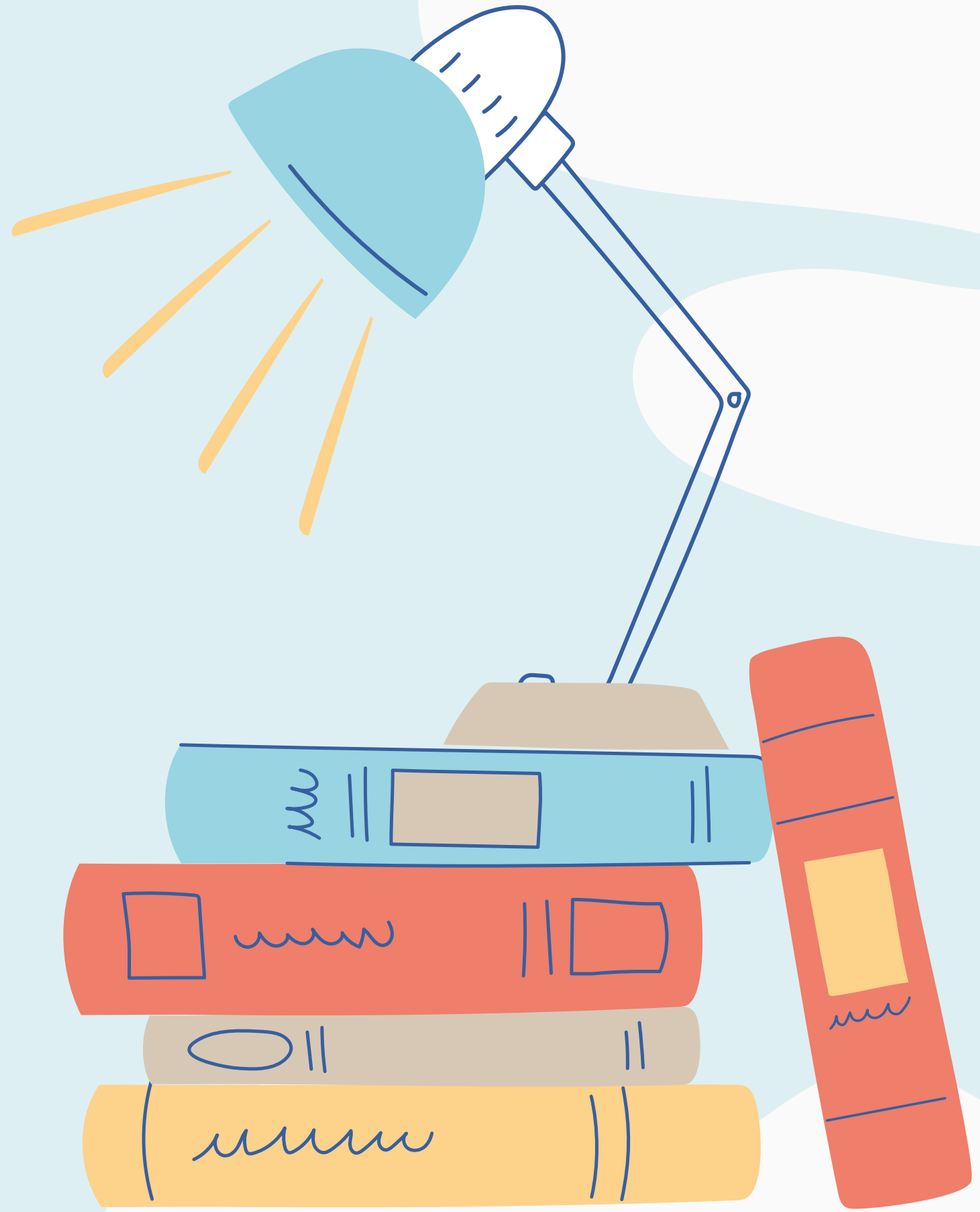
- sh: is the command-line interpreters. it is the original unix shell and available in most unix-like OS.
- bash is the enhanced version of of sh, it is backward-compatible with sh while also providing additional features and improvements. it is common in linux but not that available in other unix-like OS.



Answer

d. the difference between running ./ and bash:

- ./script.sh: this will execute the script and the system will use the #! to know which shell interpreter to use.
- bash script.sh: this takes the file as an argument for the bash interpreter to execute, so even if the shebang specify to use sh interpreter, it doesn't care. also it runs regardless of the execute permission whereas ./script.sh will need execute permission.



Task 2

2. Let's create your first script!
   a. Write your first script named as "**hello_ubuntu**" which displays "**Hello! This is Ubuntu.**"
   b. What command do you use to set permission for your script in order to make it executable for everyone?
   c. What command do you use to execute your script above? (*Hint*: Page 356)
   d. Modify your script by adding $*name* as a variable and display output input name

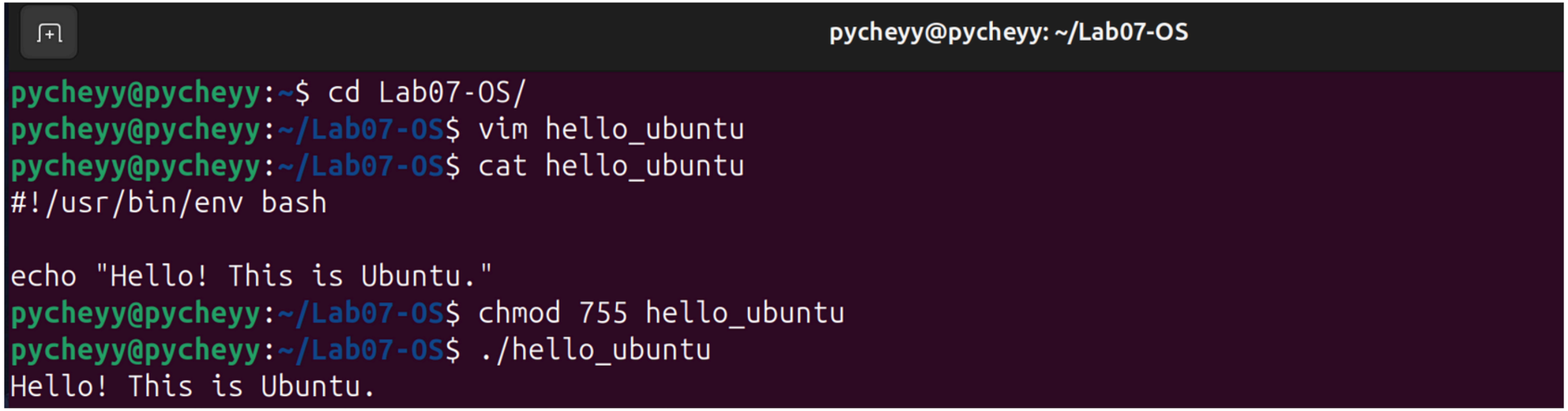
      $ Enter your name:

      $ John *(your input)*

      $ Hello, John!

   e. Take a screenshot of content of the script and result from running it.

Answer



make file, permission, excuting file

Continue

```
pycheyy@pycheyy:~/Lab07-OS$ vim hello_ubuntu
pycheyy@pycheyy:~/Lab07-OS$ cat hello_ubuntu
#!/usr/bin/env bash

echo "Enter your name:"
read name
echo "Hello, $name";
pycheyy@pycheyy:~/Lab07-OS$ ./hello_ubuntu
Enter your name:
John
Hello, John
pycheyy@pycheyy:~/Lab07-OS$
```

modify hello_ubuntu and excuting it

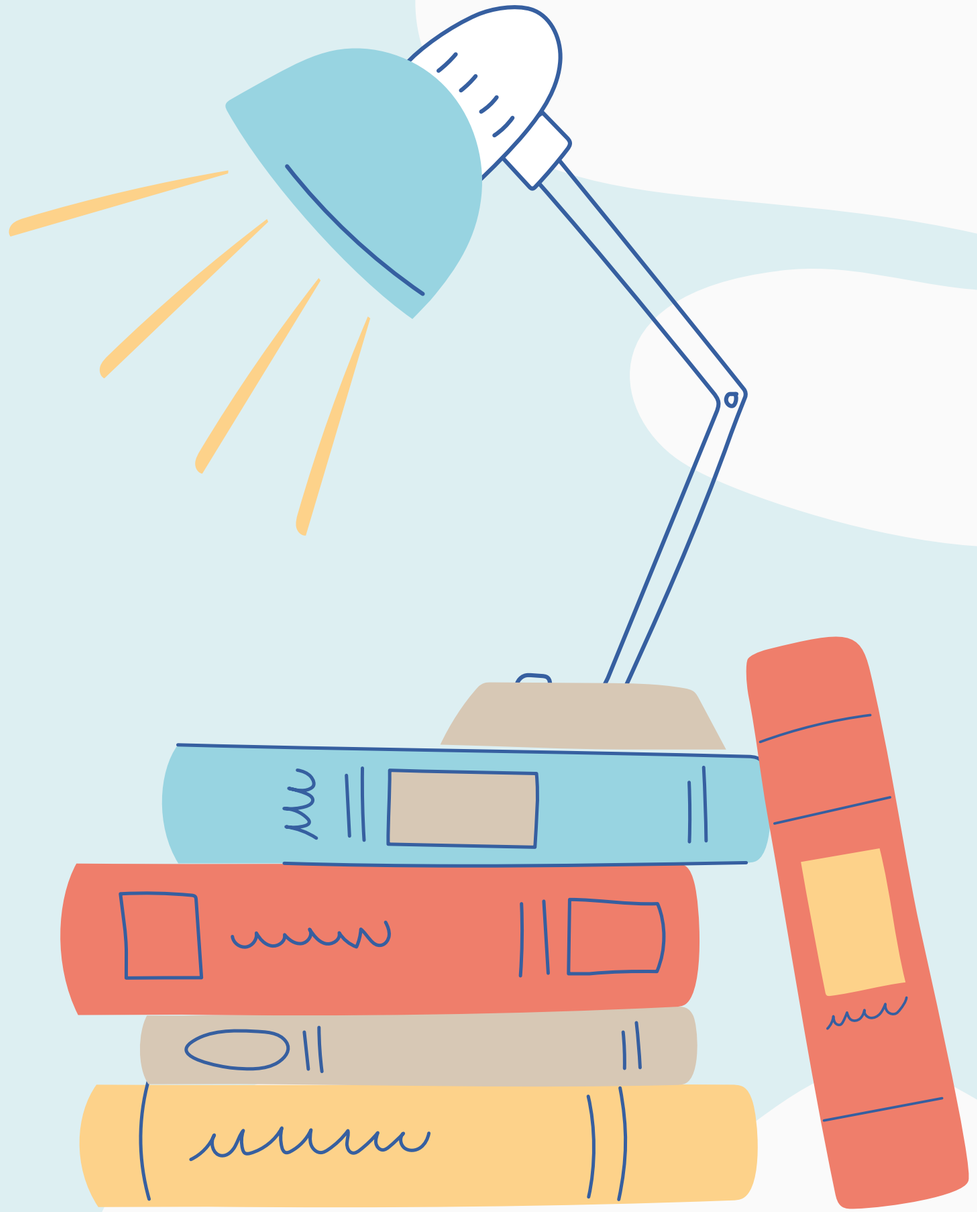
Task 3

3. Let's make your script easier to execute and more advanced!
   a. Don't you know that you can run your script above just like you run the other commands!? There are many options which can make your script executable like a command. List down two options which makes your script executable by using this command *hello_ubuntu*. (*Hint*: Page 356)
   b. Command *whoami* displays the name of current user. Now, let's make a new script named as "***hello_there***" and commanded by *hello_there* which can recognize the current username, and it will display, for example, "**Hello ratha! How are you today?**", where "**ratha**" is the current user for this example. (*Hint*: In a script, you can get the result from any command by putting that command inside Left Single Quotation Mark *`command`.)*

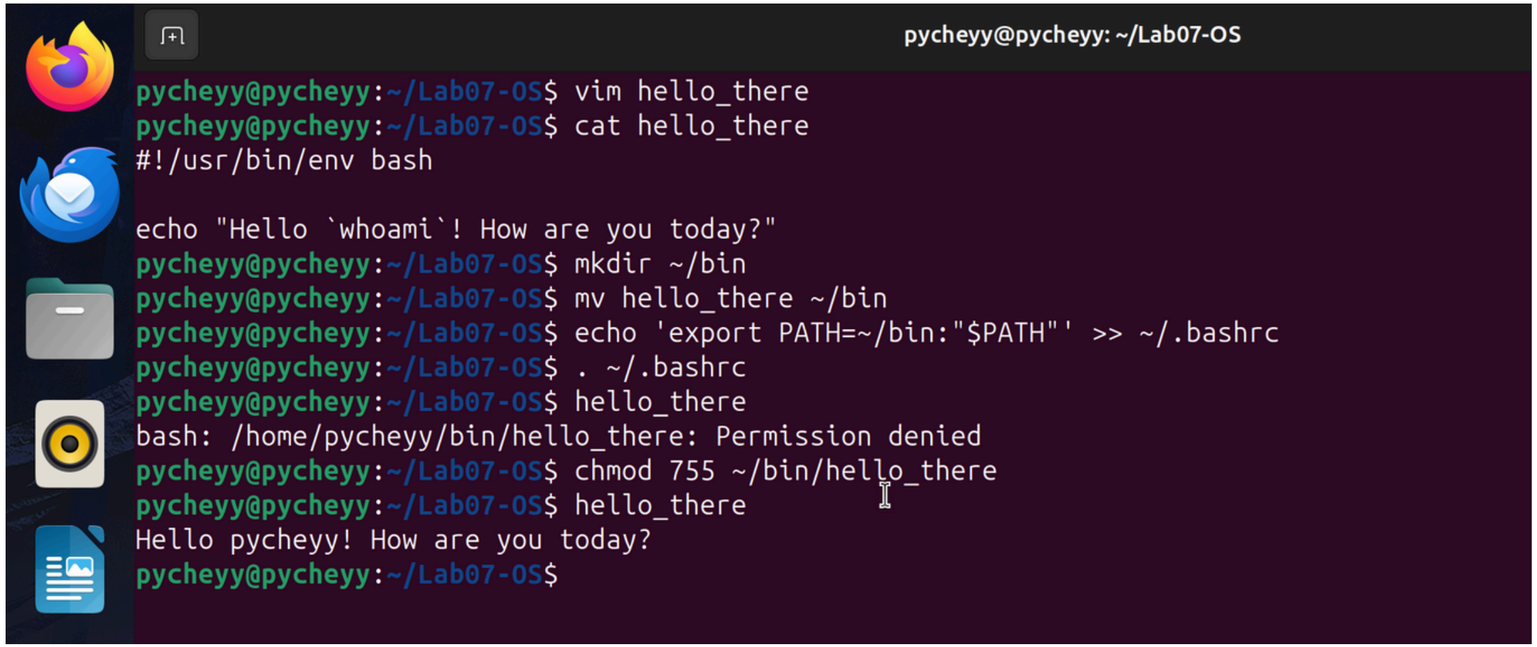
Answer

a.Two options to make hello_ubuntu run like command:

- moving hello_ubuntu to /usr/local/bin using sudo.
- making our own bin dir in home directory then move hello_ubuntu to bin then add ~/bin to PATH variable



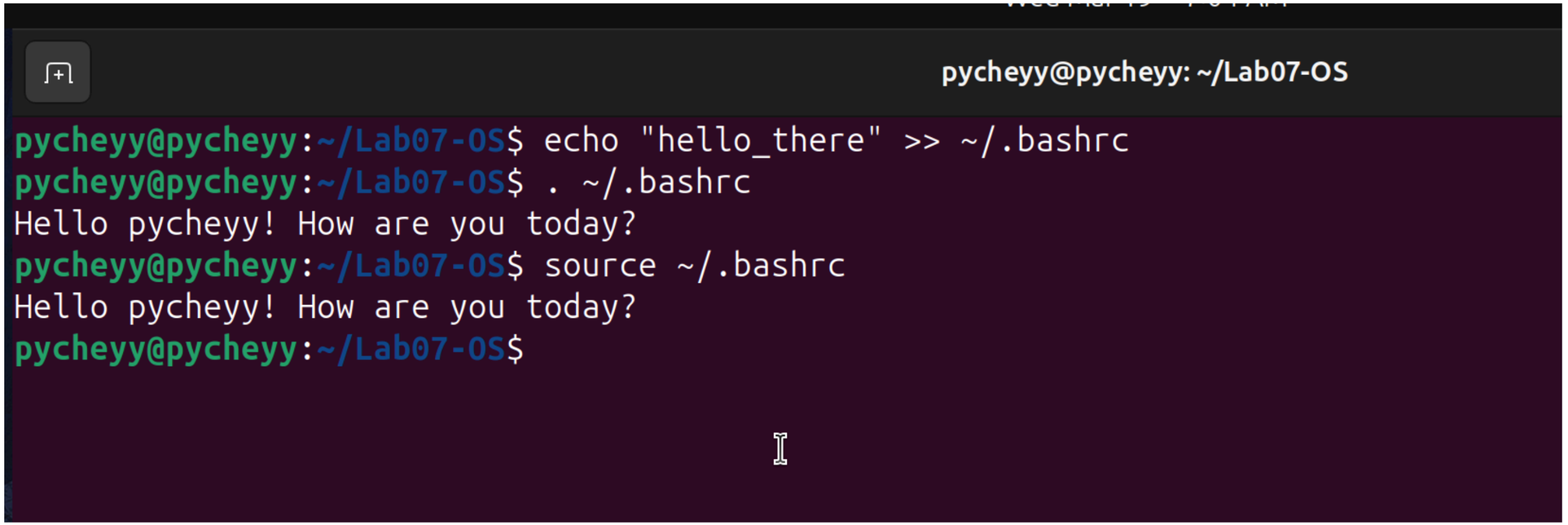
b.


pycheyy@pycheyy: ~/Lab07-OS
pycheyy@pycheyy:~/Lab07-OS$ vim hello_there
pycheyy@pycheyy:~/Lab07-OS$ cat hello_there
#!/usr/bin/env bash
echo "Hello `whoami`! How are you today?"
pycheyy@pycheyy:~/Lab07-OS$ mkdir ~/bin
pycheyy@pycheyy:~/Lab07-OS$ mv hello_there ~/bin
pycheyy@pycheyy:~/Lab07-OS$ echo 'export PATH=~/bin:"$PATH"' >> ~/.bashrc
pycheyy@pycheyy:~/Lab07-OS$. ~/.bashrc
pycheyy@pycheyy:~/Lab07-OS$ hello_there
bash: /home/pycheyy/bin/hello_there: Permission denied
pycheyy@pycheyy:~/Lab07-OS$ chmod 755 ~/bin/hello_there
pycheyy@pycheyy:~/Lab07-OS$ hello_there
Hello pycheyy! How are you today?
pycheyy@pycheyy:~/Lab07-OS$

Task 4

4. Let's have some fun with your previous script! Do research on how to make a script run automatically whenever the terminal is opened, and then make your terminal run your last script "**hello_there**" every single time you open your terminal. After everything is set, take a screenshot to proof your success. (Hint: file "**.bashrc**")

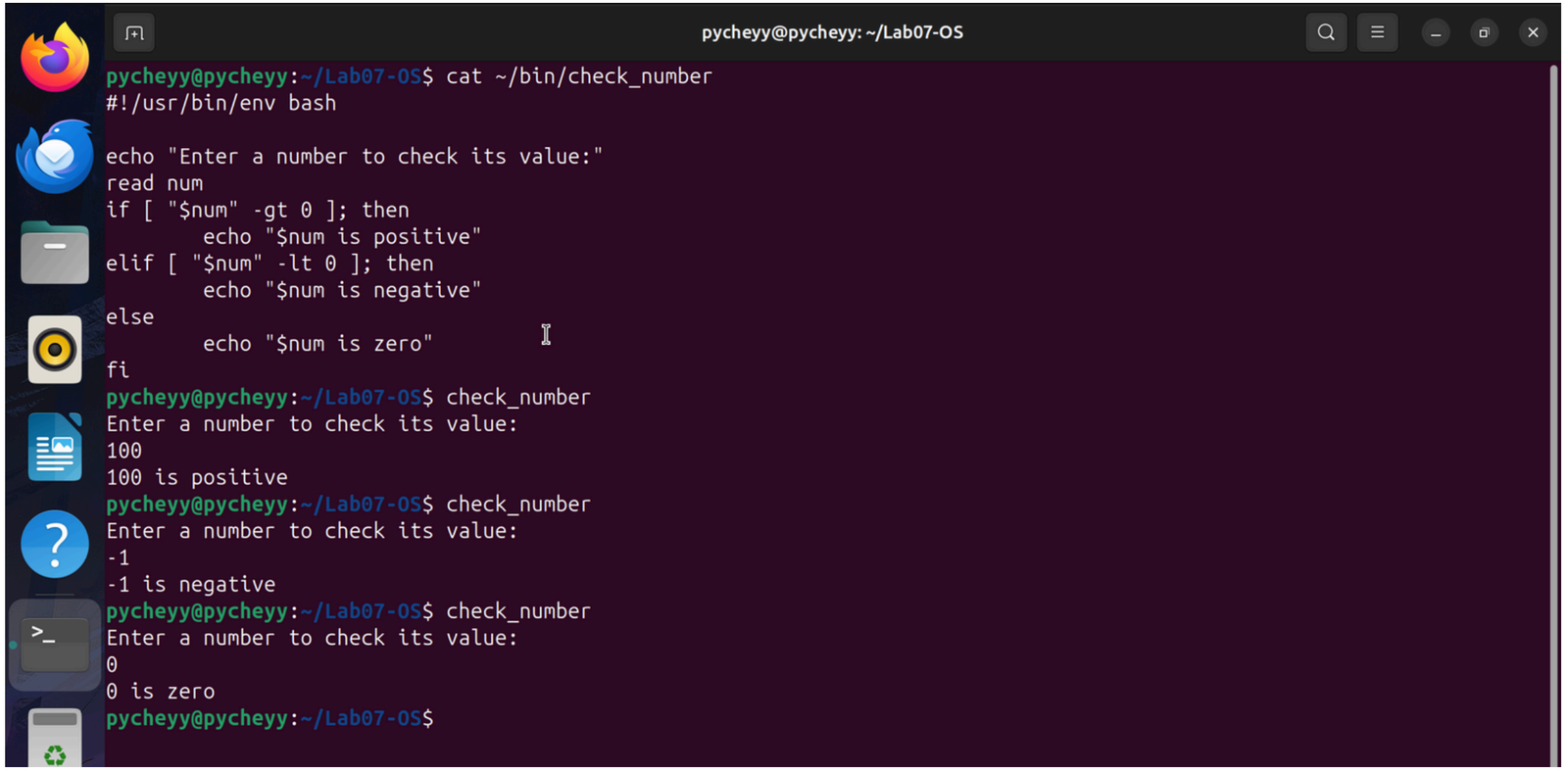
Answer


pycheyy@pycheyy: ~/Lab07-OS
pycheyy@pycheyy:~/Lab07-OS$ echo "hello_there" >> ~/.bashrc
pycheyy@pycheyy:~/Lab07-OS$. ~/.bashrc
Hello pycheyy! How are you today?
pycheyy@pycheyy:~/Lab07-OS$ source ~/.bashrc
Hello pycheyy! How are you today?
pycheyy@pycheyy:~/Lab07-OS$

Task 5

5. Write a script named *check_number.sh* that prompts the user to enter a number. The script should then check if the given number is **positive**, **negative**, or **zero**, and display the appropriate message. Hint: Use -gt for greater than, -lt for less than, and -eq for equal comparison.

Answer


pycheyy@pycheyy: ~/Lab07-OS
pycheyy@pycheyy:~/Lab07-OS$ cat ~/bin/check_number
#!/usr/bin/env bash
echo "Enter a number to check its value:"
read num
if ["$num" -gt 0]; then
echo "$num is positive"
elif ["$num" -lt 0]; then
echo "$num is negative"
else
echo "$num is zero"
fi
pycheyy@pycheyy:~/Lab07-OS$ check_number
Enter a number to check its value:
100
100 is positive
pycheyy@pycheyy:~/Lab07-OS$ check_number
Enter a number to check its value:
-1
-1 is negative
pycheyy@pycheyy:~/Lab07-OS$ check_number
Enter a number to check its value:
0
0 is zero
pycheyy@pycheyy:~/Lab07-OS$

Thank you

Prak Pychey

prakpycheyy@gmail.com

Resources