|  |
| --- |
| Name: Phạm Mai Dung  ID: 19520477  Class: IT007.L21.1 |

OPERATING SYSTEM  
LAB X’S REPORT

**SUMMARY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | | **Status** | **Page** |
| Section 1.5 | Task name 1 | Done | 2 - 9 |
| Task name 2 | Done | 2 – 11 |
| Task name 3 | Done | 11 – 13 |
| Task name 4 | Done | 13 – 16 |
|  |  |  |
|  |  |  |
| … | … |  |  |
| … |  |  |

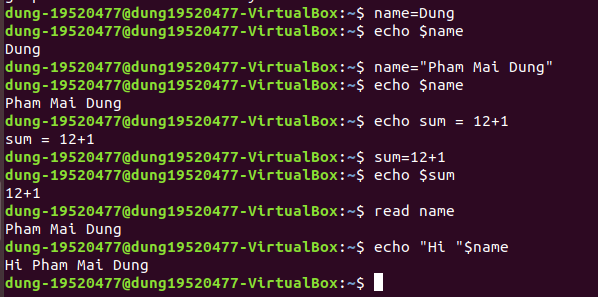
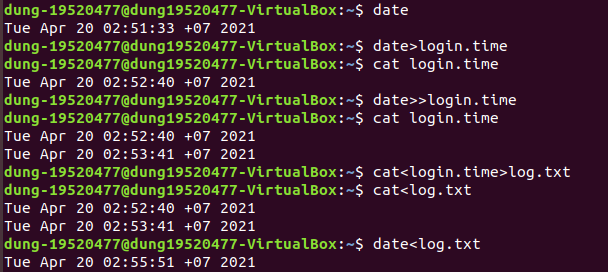
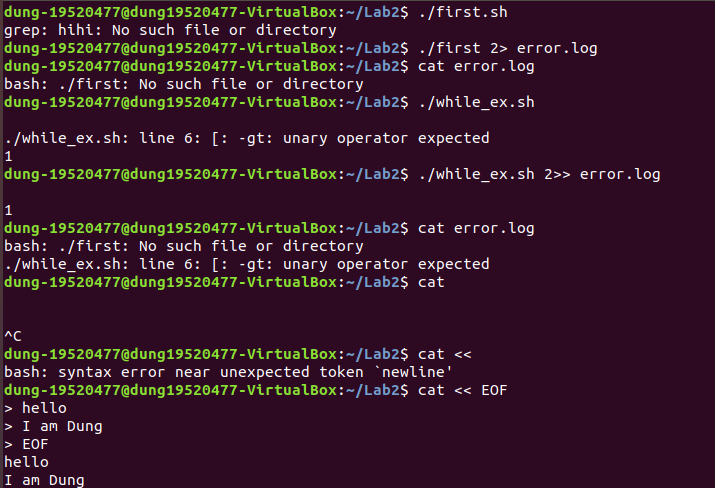
**Self-scrores: 8**

*\*Note: Export file to* ***PDF*** *and name the file by following format:* ***LAB X – <Student ID>.pdf***

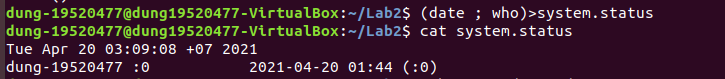
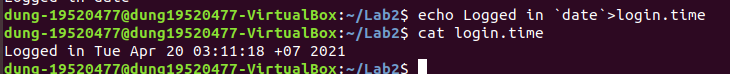
# Section 1.5

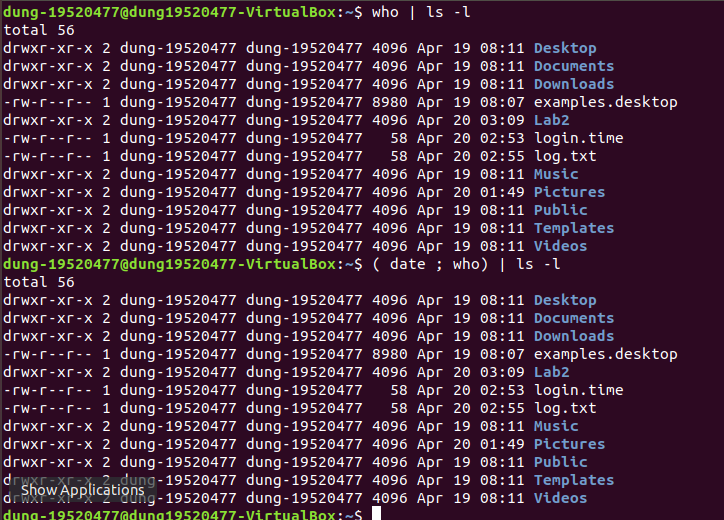
## Chạy tất cả các đoạn lệnh ví dụ ở phân 2.4. Chụp hình kết quả chạy các file script và lưu vào báo cáo

*Hình 1.1\_ Truy tìm và hiển thị nội dung của các tệp nguồn chứa chuỗi main()*

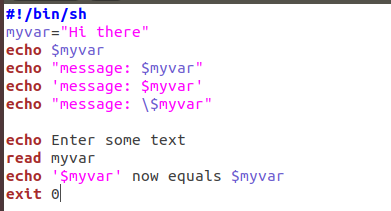
*Hình 1.2\_ Sử dụng biến*

*Hình 1.3\_ các ký tự chuyển hướng vào/ra*

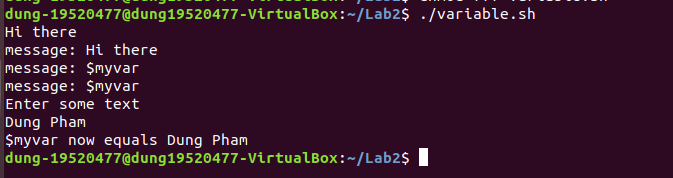
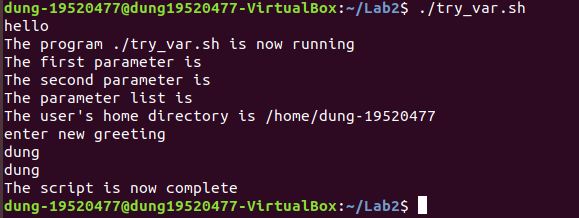
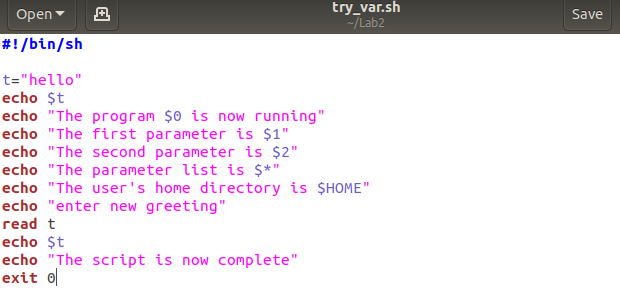
*Hình 1.4\_ dấu ;*

*Hình 1.5\_ dấu ` `*

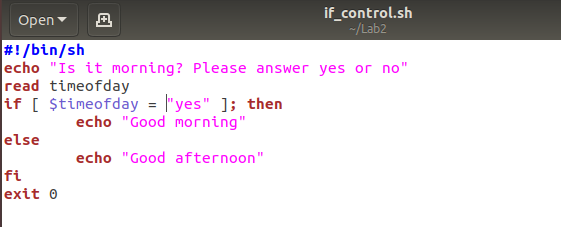
*Hình 1.6\_ pipelines*

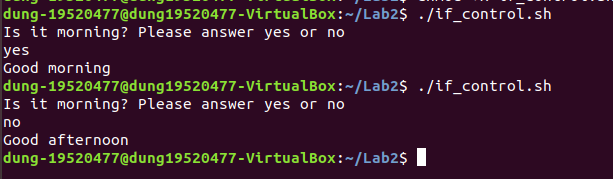
**

*variable.sh*

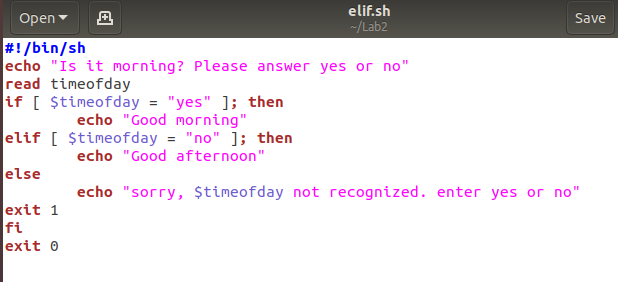
*Hình 1.7\_ thực thi file variable.sh*

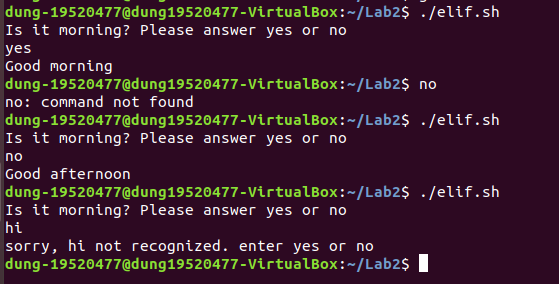
*Hình 1.8\_ thực thi file try\_var.sh*

**

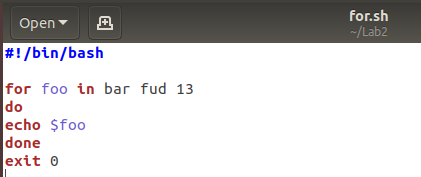
**

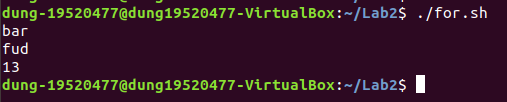
*Hình 1.9\_ thực thi file if\_control.sh (cấu trúc if-else)*

**

**

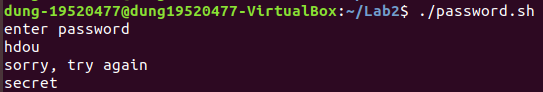
*Hình 1.10\_ thực thi file elif.sh*

**

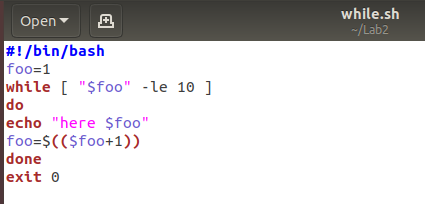
**

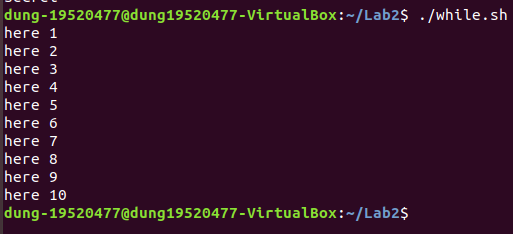
*Hình 1.11\_ thực thi file for.sh*

## 

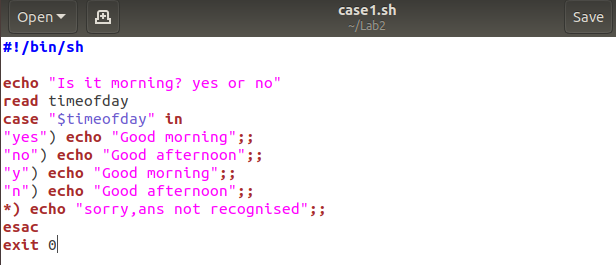
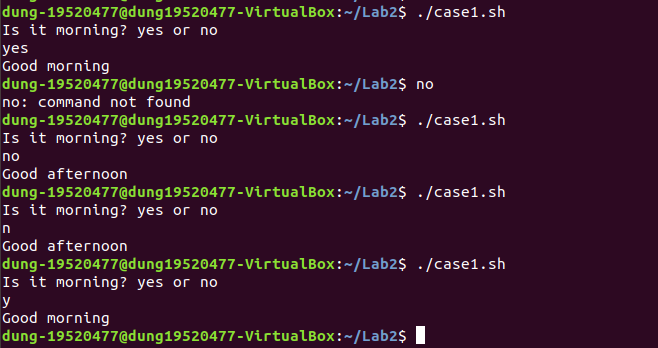
**

*Hình 1.12\_ thực thi file pasword.sh*

**

**

*Hình 1.13\_ thực thi file while.sh*

*Hình 1.14\_ thực thi file case1.sh*

## Viết chương trình cho phép nhập tên và MSSV. Kiểm tra và yêu cầu nhập lại khi MSSV không trùng với MSSV của mình. In kết quả ra màn hình.

* Text

  Description automatically generatedTại thư mục Lab2, ta sử dụng lệnh ***gedit bai2.sh*** tạo một file kịch bản để viết chương trình, dùng lệnh ***chmod +x bai2.sh*** để cấp quyền thực thi cho file này.

*Hình 2.1\_ Tạo file kịch bản bai2.sh và cấp quyền thực thi*

* Trong file bai2.sh:
* Lệnh ***#!/bin/sh:*** file dùng sh shell để thực thi
* Lệnh ***echo “Nhap...”*** : in ra màn hình (dễ nhìn hơn)
* Lệnh ***read tenSV*** (tương tự ***read mssv***): cho phép nhập chuỗi vào và gán chuỗi đó vào biến tenSV (mssv)
* Gán giá trị là MSSV của mình vào biến MyID: ***MyID=19520477***
* Sử dụng vòng lặp while để kiểm tra xem MSSV đã nhập vào có trùng với giá trị của biến MyID hay không:

+ Điều kiện của while là nếu giá trị của biến mssv không bằng với giá trị của biến MyID là đúng (***while [ $mssv -ne $MyID ]*** ) thì vòng lặp while thực hiện nhập lại MSSV:

do

...

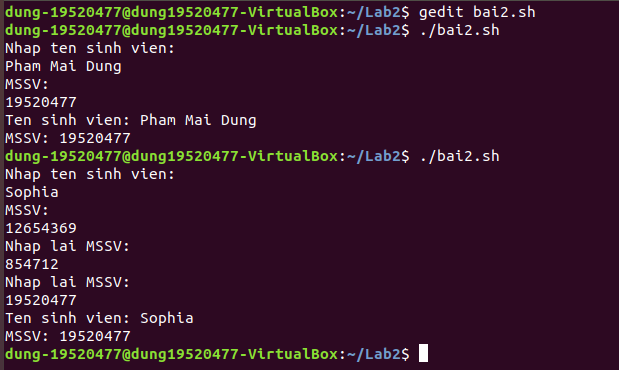
done (như trong hình 2.2 dưới)

* Sau khi đã nhập theo đúng yêu cầu (hay thoát khỏi vòng lặp while), ta sẽ in ra màn hình giá trị của 2 biến tenSV và mssv bằng lệnh echo (hình 2.2).
* *Note: tenSV, mssv hay MyID là các biến, khi tiền tố có dấu $ là giá trị của biến, ví dụ $mssv là giá trị của biến mssv.*

Graphical user interface, text, application, chat or text message

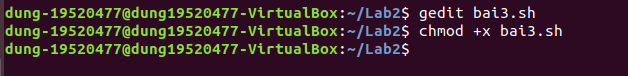
Description automatically generated

*Hình 2.2\_ chương trình nhập vào thông tin sinh viên (tên và MSSV) và kiểm tra yêu cầu (được viết trong file bai2.sh)*

* Hình 2.3: in ra màn hình kết quả thực thi file bai2.sh:
* Kết quả thực thi cho thấy, khi ta nhập MSSV không trùng khớp thì vòng lặp while sẽ tiếp tục thực hiện đến khi ta nhập MSSV trùng khớp mới dừng. Kết quả in ra màn hình gồm tên (“Ten sinh vien:” + $tenSV) và mã số sinh viên (“MSSV:” + $mssv).

*Hình 2.3\_ Kết quả thực thi chương trình nhập vào thông tin sinh viên đã được viết trong bai2.sh*

## Viết chương trình cho phép nhập vào một số n. Kiểm tra nếu n<10 thì nhập lại. Tính tổng các số từ 1 đến n. In kết quả ra màn hình.

* Tại thư mục Lab2, ta sử dụng lệnh ***gedit bai3.sh*** tạo một file kịch bản để viết chương trình, dùng lệnh ***chmod +x bai3.sh*** để cấp quyền thực thi cho file này.

*Hình 3.1\_ Tạo file kịch bản bai3.sh và cấp quyền thực thi*

* Trong file bai3.sh:
* Lệnh ***#!/bin/sh:*** file dùng sh shell để thực thi
* Lệnh ***echo “Nhap...”*** : in ra màn hình (dễ nhìn hơn)
* ***read num***: Nhập vào một số bất kỳ (nhưng kiểu chuỗi) và gán vào biến num
* Sử dụng vòng lặp while xét điều kiện: nếu giá trị biến num nhỏ hơn 10, thực hiện nhập lại số n

*//-lt: so sánh bé hơn; nếu giá trị của num < 10, thực thi lệnh trong vòng lặp*

***while [ $num -lt 10 ] do...done*** (line 5 – 9 trong hình 3.2)

* Tính tổng từ 1 đến n:

+ Khởi tạo 2 biến sum và i: ***sum=0, i=0***

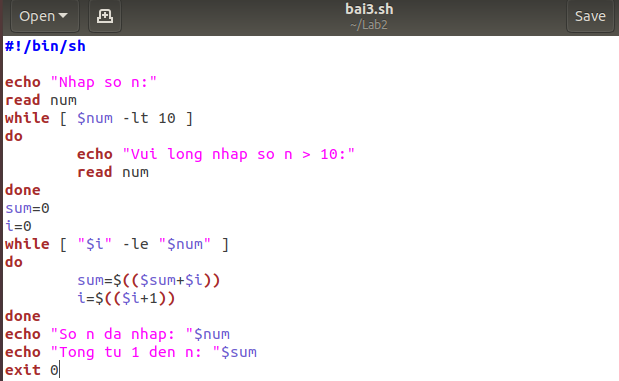
+ Dùng vòng lặp while xét điều kiện các giá trị của biến i ($i) nếu nhỏ hơn hoặc bằng giá trị biến num ($num), thực hiện cộng vào sum các giá trị của biến i, sau đó tăng i lên 1 đơn vị.

*//-le: so sánh bé hơn hoặc bằng; nếu $i <= $num, thực thi lệnh trong vòng lặp*

***while [ “$i” -le “$num” ] do...done*** (line 12 – 16 trong hình 3.2)

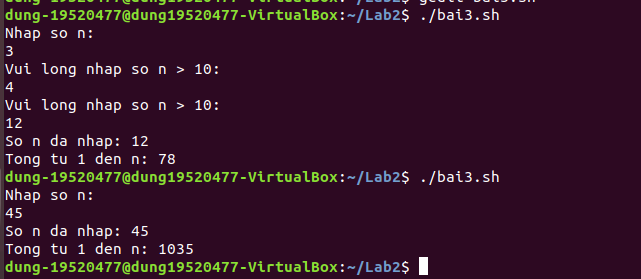
+ Vì giá trị của các biến là kiểu chuỗi , nên khi thực hiện sum+i và i+1 lệnh được viết như sau: ***sum=$(($sum+$i)), i=$(($i+1))*** (“ép kiểu”)

* Lệnh ***echo “So n da nhap: ”$num***: in ra màn hình số đã nhập (được gán trong biến num)
* Lệnh ***echo “Tong tu 1 den n: ”$sum***: in ra màn hình tổng của các số từ 1 đến n (được gán trong biến sum)



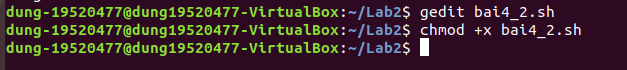
*Hình 3.2\_ chương trình nhập vào một số n, kiểm tra yêu cầu và tính tổng 1,2,3,...,n (được viết trong file bai3.sh)*

* Hình 3.3: in ra màn hình kết quả thực thi file bai3.sh:

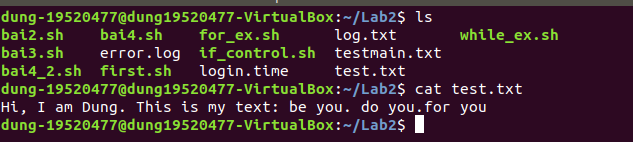
Kết quả thực thi cho thấy các vòng lặp while đã thực thi đúng, kết quả in ra màn hình theo yêu cầu

*Hình 3.3\_ Kết quả thực thi chương trình nhập vào một số đã được viết trong bai3.sh*

## Viết chương trình cho phép nhập vào một chuỗi. Kiểm tra chuỗi đó có tồn tại trong một file text cùng thư mục hay không.

* Tại thư mục Lab2, ta sử dụng lệnh ***gedit bai4\_2.sh*** tạo một file kịch bản để viết chương trình, dùng lệnh ***chmod +x bai4\_2.sh*** để cấp quyền thực thi cho file này.

*Hình 4.1\_ Tạo file kịch bản bai4\_2.sh và cấp quyền thực thi*

* Tạo file test.txt trong cùng thư mục (em cũng dùng lệnh gedit) và có nội dung như hình 4.2:

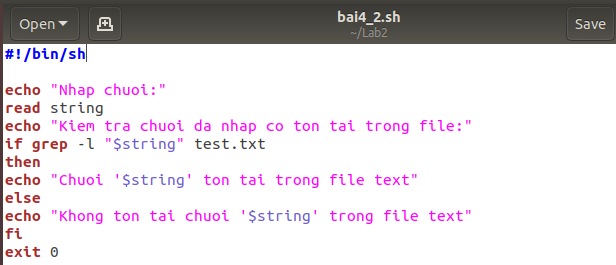
*Hình 4.2\_ Hiển thị file test.txt trong thư mục Lab2 và nội dung của nó.*

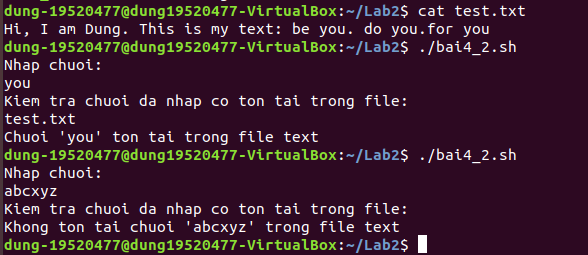
* Trong file bai4\_2.sh: (Hình 4.3)
* Lệnh ***#!/bin/sh:*** file dùng sh shell để thực thi
* Lệnh ***echo “Nhap chuoi:”*** : in ra màn hình yêu cầu nhập chuỗi
* ***read string***: Nhập vào một chuỗi bất kỳ và gán vào biến string
* Dùng cấu trúc if else để kiểm tra xem chuỗi vừa nhập có tồn tại tong file text hay không:

+ ***grep -l ‘$string’ test.txt***: tìm chuỗi đã nhập có tồn tại trong file test.txt hay không

+ Nếu có, in ra màn hình có tồn tại bằng lệnh: ***echo “Chuoi ‘$string’ ton tai trong file text”***

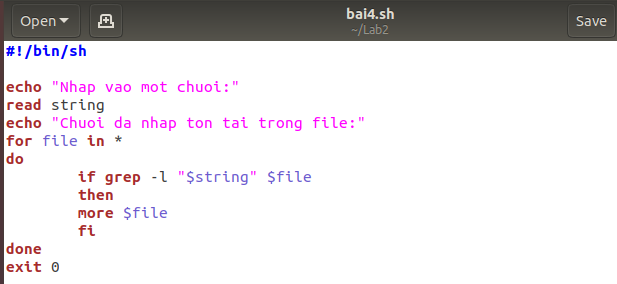
+ Nếu không, ỉn ra màn hình không tồn tại bằng lệnh: ***echo “Khong ton tai chuoi ‘$string’ trong file text”***

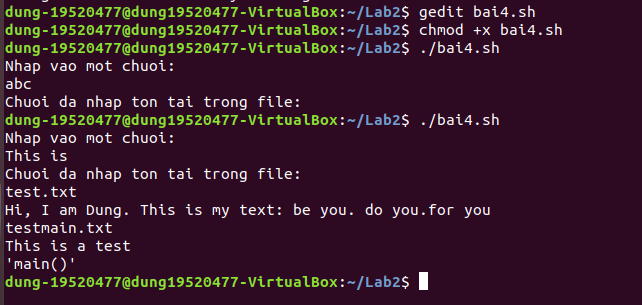
*Hình 4.3\_ chương trình nhập vào một chuỗi, kiểm tra chuỗi có trong file text cùng thư mục hay không (được viết trong file bai4\_2.sh)*

* Hình 4.4: in ra màn hình kết quả thực thi file bai4\_2.sh:

*Hình 4.5\_ Kết quả thực thi chương trình nhập vào một chuỗi, kiểm tra tồn tại hay không đã được viết trong bai4\_2.sh*

* Không chỉ kiểm tra xem chuỗi có tồn tại trong 1 file hay không, ta còn có thể tìm xem chuỗi có tồn tại trong nhiều file của thư mục hay không.
* Tạo một file kịch bản bai4.sh có nội dung như sau: (hình 4.6)
* Sử dụng vòng lặp for để duyệt các file trong thư mục hiện hành (***for file in \****), tìm chuỗi có tồn tại trong các file bằng lệnh ***grep -l “$string” $file***

*Hình 4.6\_ chương trình nhập vào một chuỗi, kiểm tra chuỗi có trong nhiều file cùng thư mục hay không (được viết trong file bai4.sh)*

* Hiển thị kết quả chương trình thực thi file bai4.sh:

*Hình 4.7\_ Kết quả thực thi chương trình nhập vào một chuỗi, kiểm tra chuỗi tồn tại trong nhiều file hay không đã được viết trong bai4.sh*

+ Khi nhập vào chuỗi ‘abc’, vì nó không tồn tại trong file nào thuộc thư mục Lab2 nên kết quả không có trả về file nào

+ Khi nhập vào chuỗi ‘This is’, chuỗi này tồn tại trong 2 file test.txt và testmain.txt nên kết quả trả về 2 file này và nội dung của chúng.