

Name: Nguyễn Đại Hưng

ID: 24520601

Class: IT007.Q15

## OPERATING SYSTEM LAB 2'S REPORT

### SUMMARY

Task		Status	Page
Section 2.5	Ex 1	Hoàn thành	2
	Ex 2	Hoàn thành	14
	Ex 3	Hoàn thành	16
	Ex 4	Hoàn thành	18

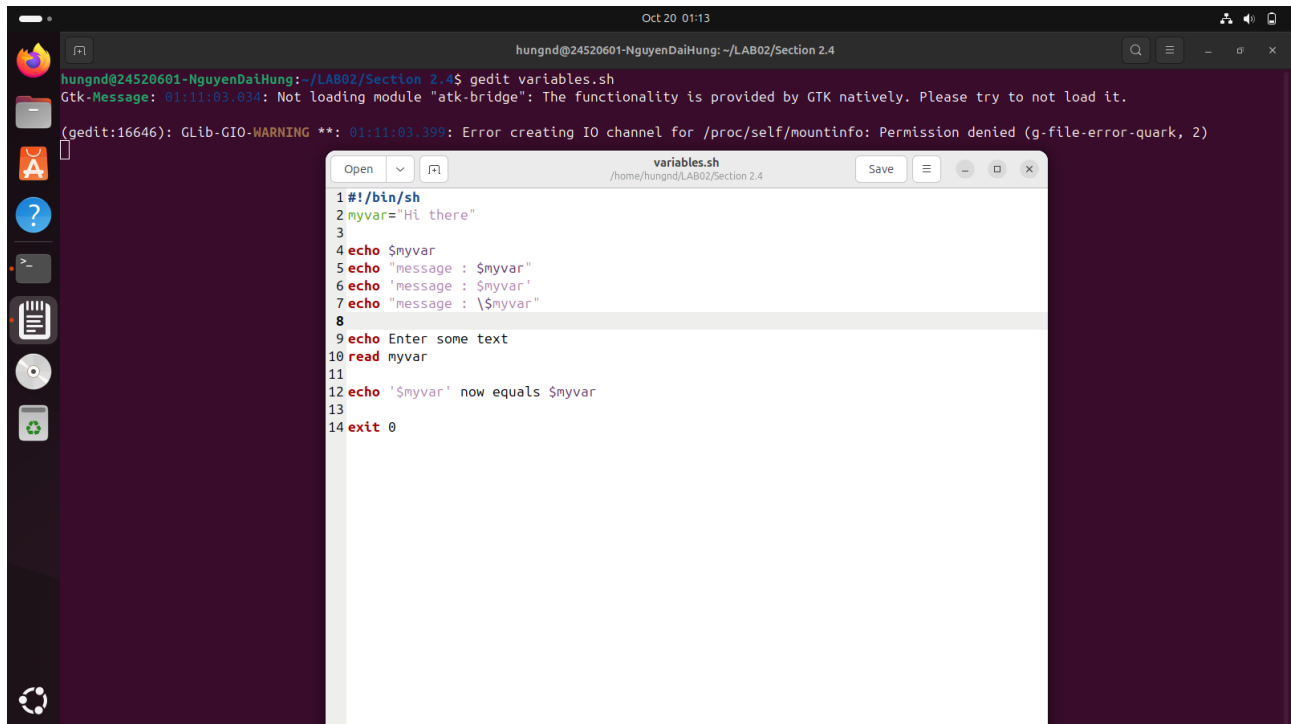
Self-scores: 10/10

*\*Note: Export file to **PDF** and name the file by following format:  
**Student ID\_LABx.pdf***

## Section 2.5

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

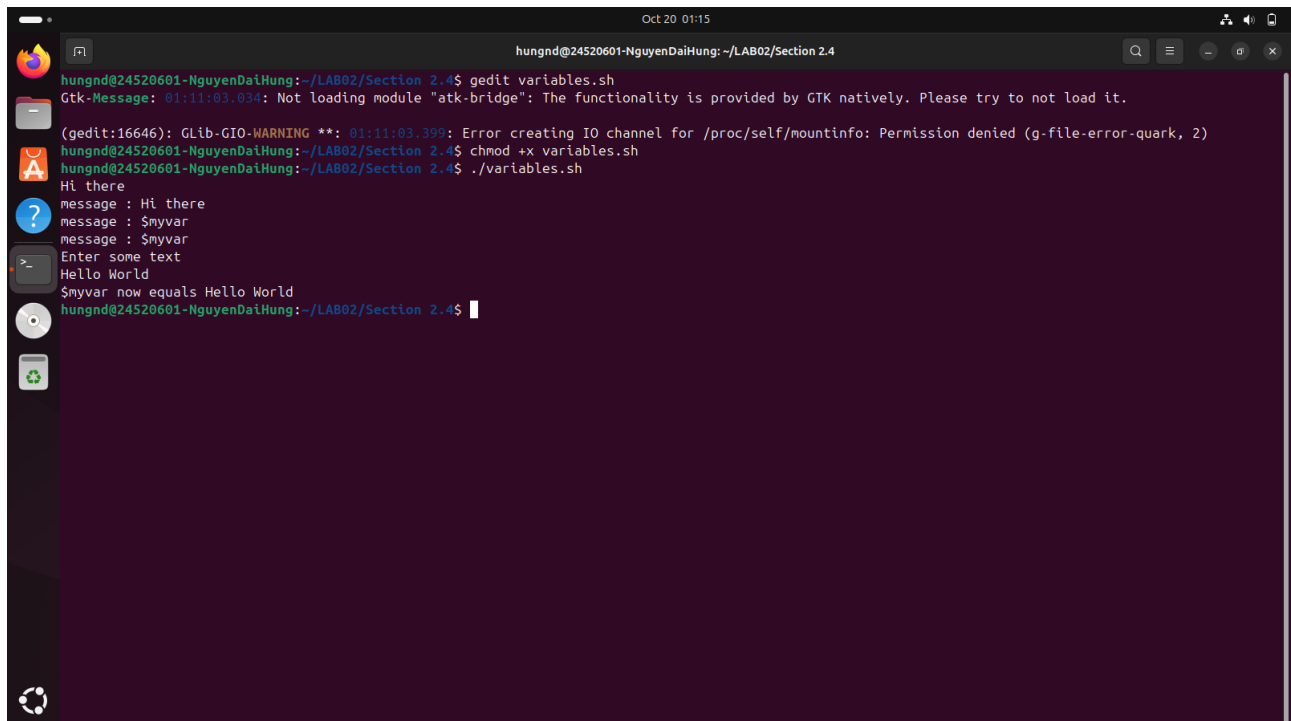
- Ví dụ 2-1: variables.sh



The screenshot shows a terminal window with a dark purple background. A file editor window titled 'variables.sh' is open in the foreground, displaying the following script content:

```
1#!/bin/sh
2myvar="Hi there"
3
4echo $myvar
5echo "message : $myvar"
6echo "message : $myvar"
7echo "message : \ $myvar"
8
9echo Enter some text
10read myvar
11
12echo '$myvar' now equals $myvar
13
14exit 0
```

Hình 1.1: Nội dung script của variables.sh

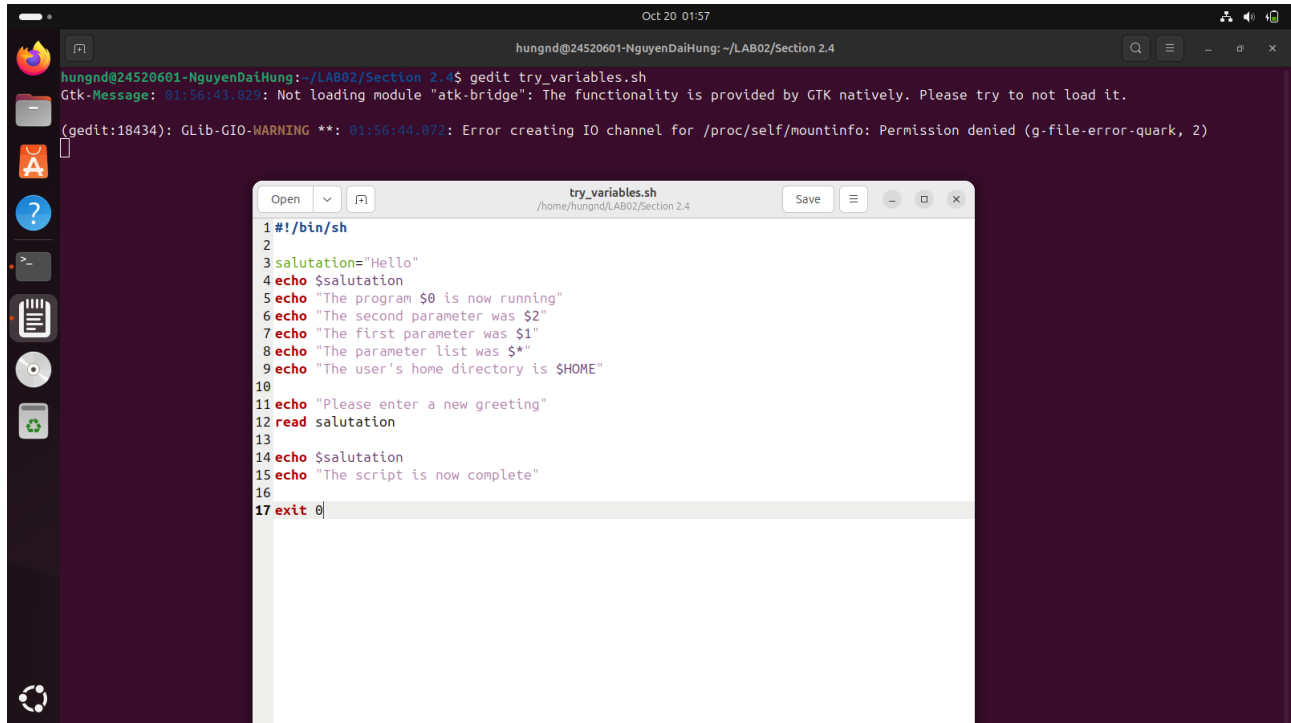


The screenshot shows the same terminal window after running the script. The output is as follows:

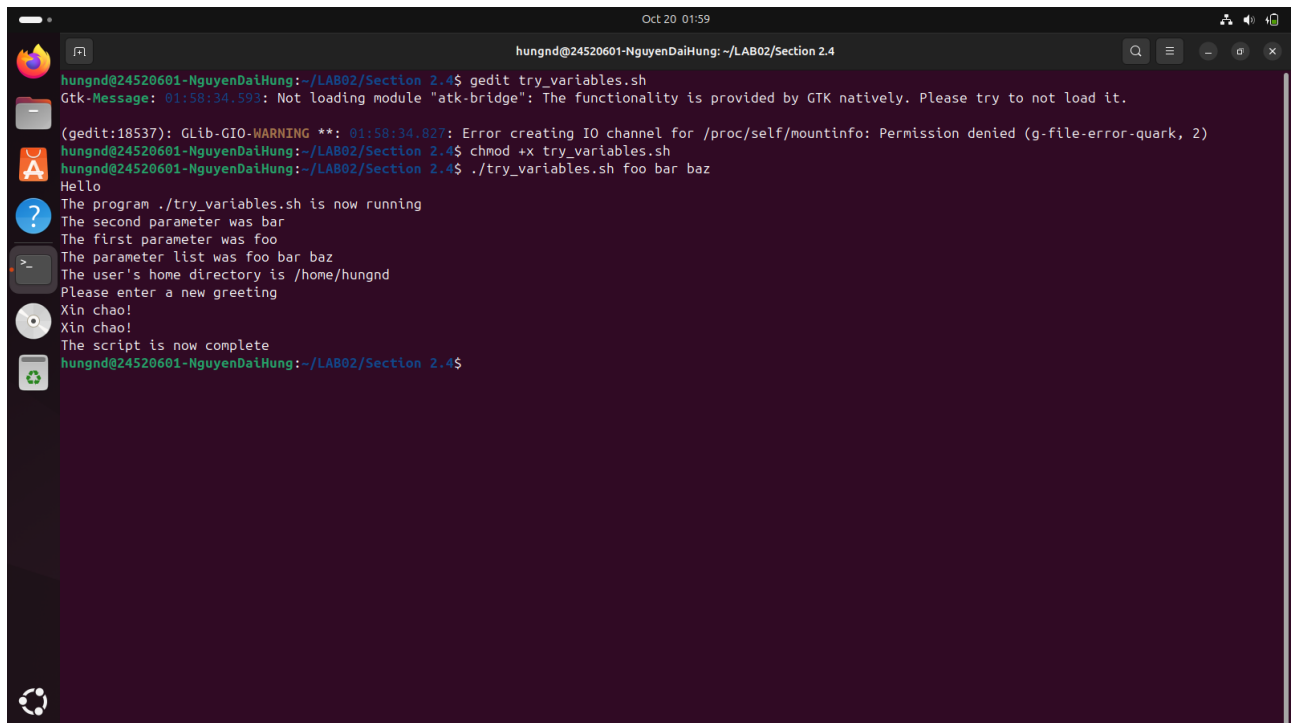
```
hungnd@24520601-NguyenDaiHung: ~/LAB02/Section 2.4$ gedit variables.sh
Gtk-Messsage: 01:11:03.034: Not loading module "atk-bridge": The functionality is provided by GTK natively. Please try to not load it.
(gedit:16646): GLib-GIO-WARNING **: 01:11:03.399: Error creating IO channel for /proc/self/mountinfo: Permission denied (g-file-error-quark, 2)
hungnd@24520601-NguyenDaiHung: ~/LAB02/Section 2.4$ chmod +x variables.sh
hungnd@24520601-NguyenDaiHung: ~/LAB02/Section 2.4$ ./variables.sh
Hi there
message : Hi there
message : $myvar
message : $myvar
Enter some text
Hello World
$myvar now equals Hello World
hungnd@24520601-NguyenDaiHung: ~/LAB02/Section 2.4$
```

Hình 1.2: Kết quả khi thực thi script `variables.sh`

- Ví dụ 2-2: `try_variables.sh`

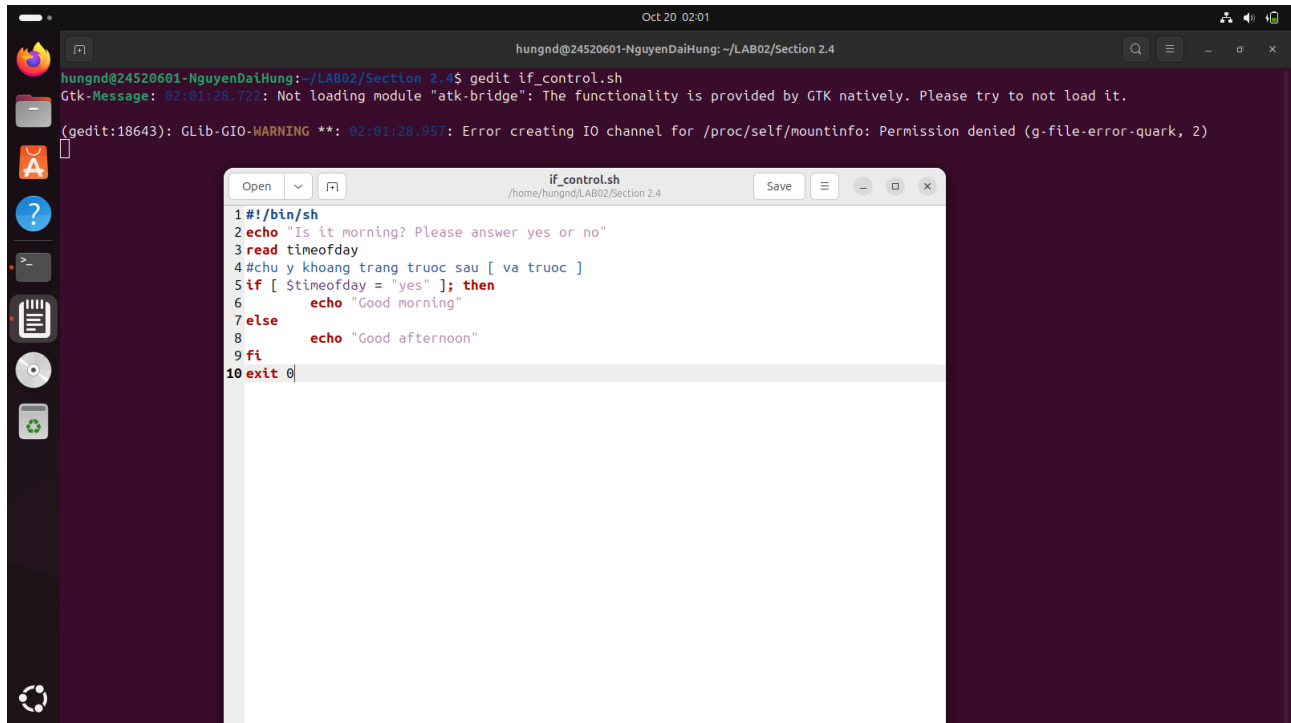


Hình 2.1: Nội dung script của `try_variables.sh`

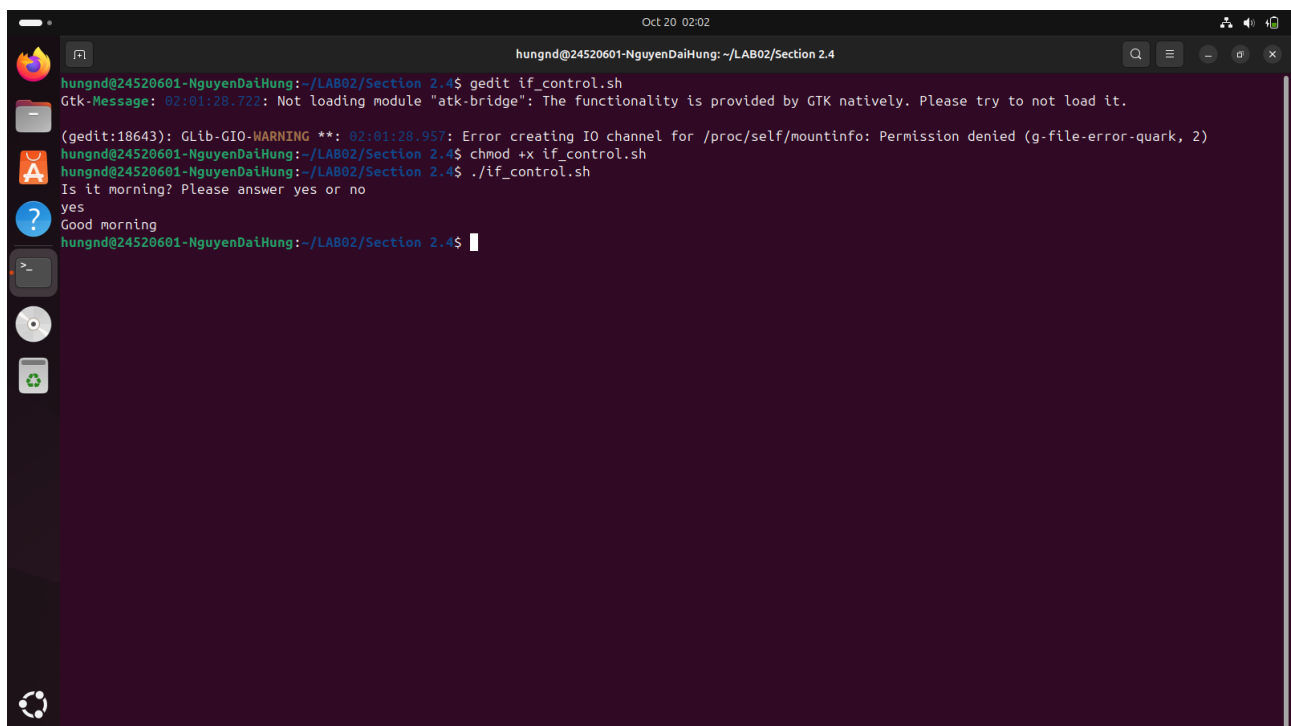


Hình 1.2: Kết quả khi thực thi `try_variables.sh`

- Ví dụ 2-3: if\_control.sh

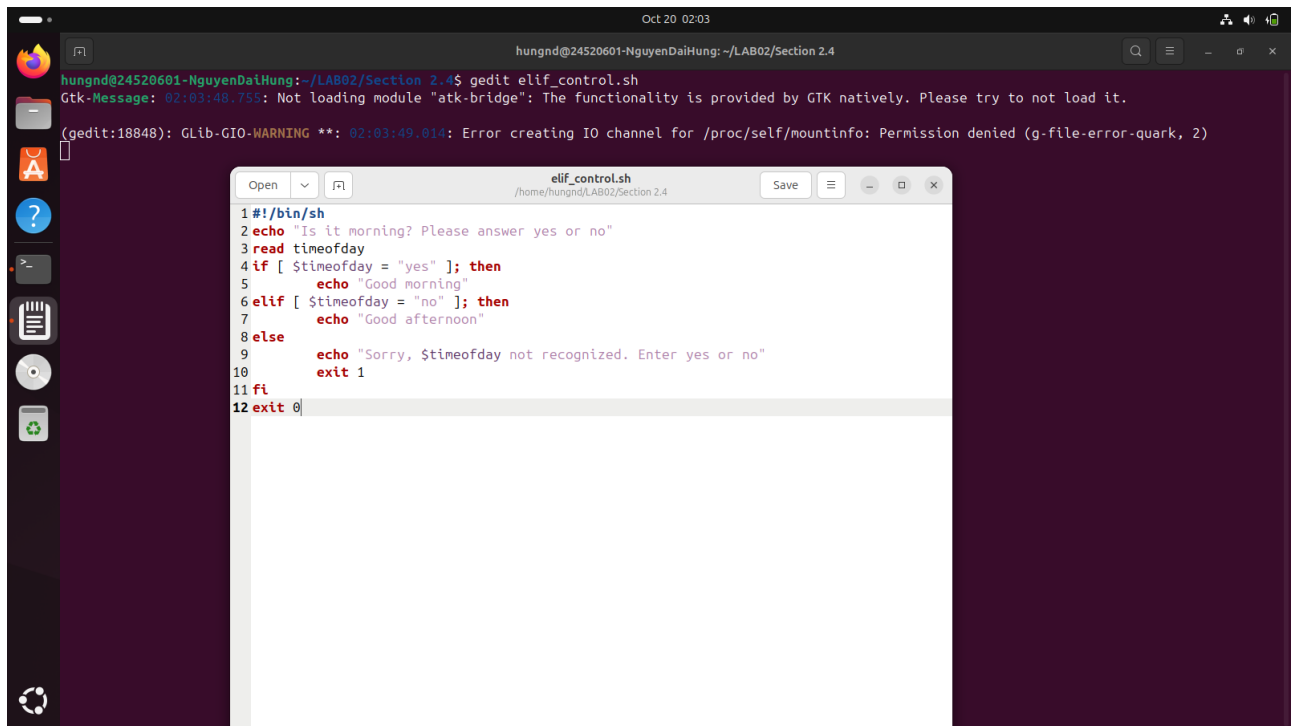


Hình 3.1: Nội dung script của if\_control.sh

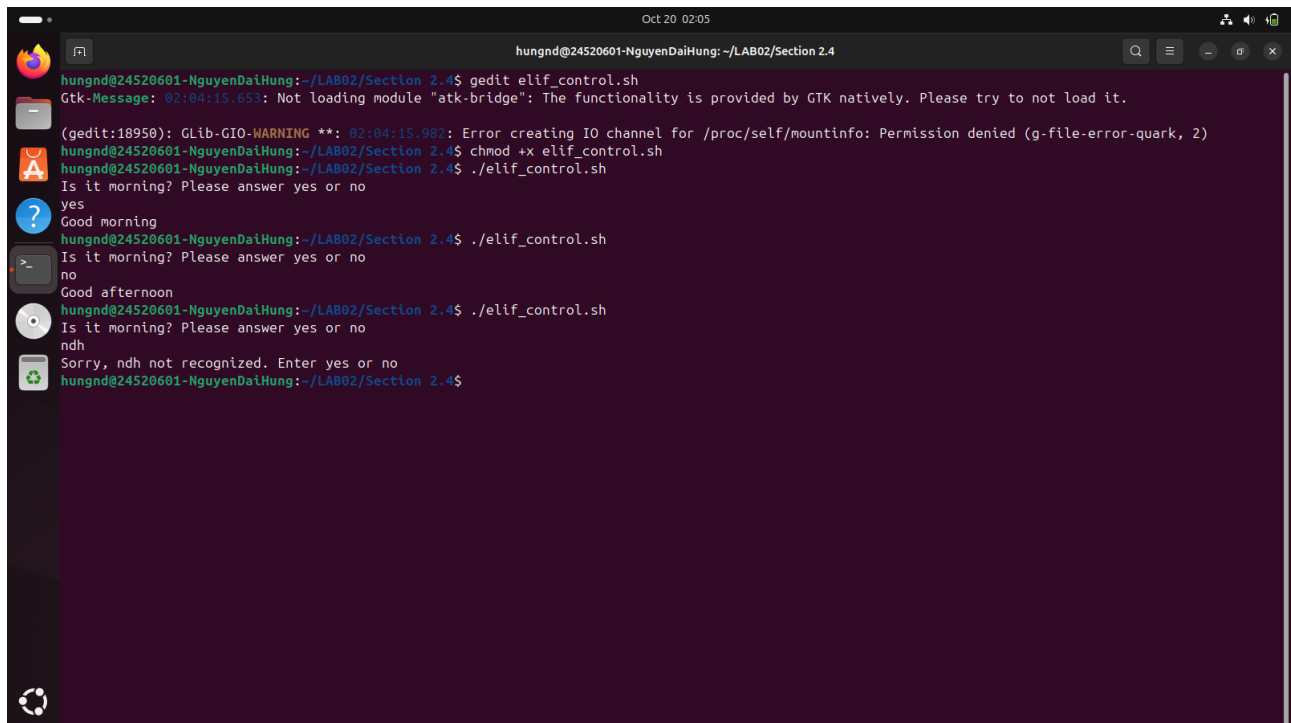


Hình 3.2: Kết quả khi thực thi script if\_control.sh

- Ví dụ 2-4: elif\_control.sh

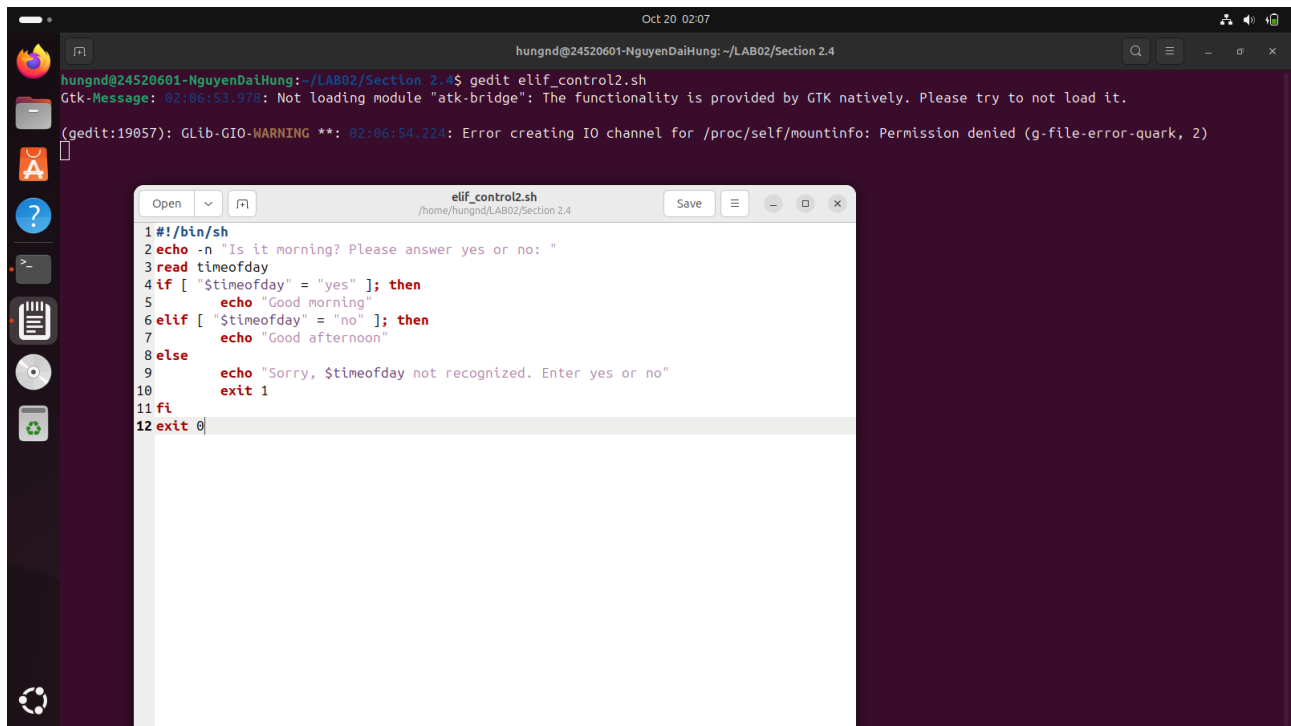


Hình 4.1: Nội dung script của elif\_control.sh

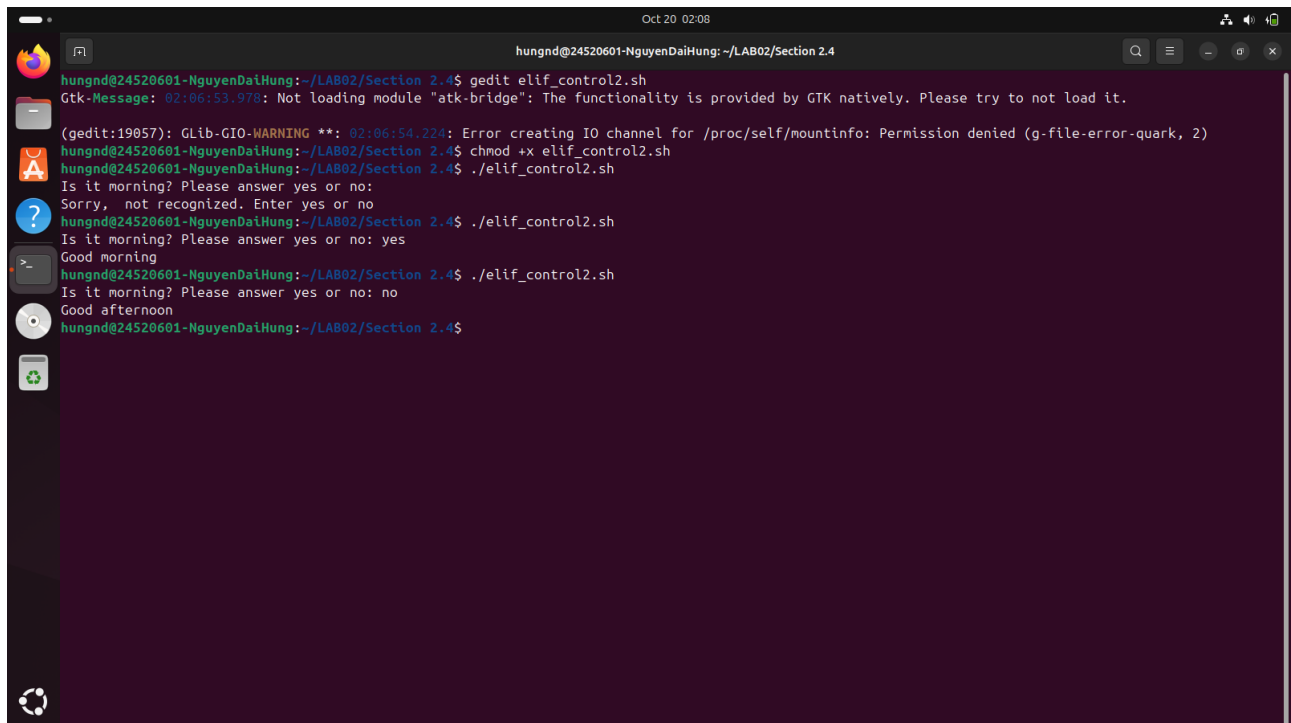


Hình 4.2: Kết quả khi thực thi script elif\_control.sh

- Ví dụ 2-5: elif\_control2.sh

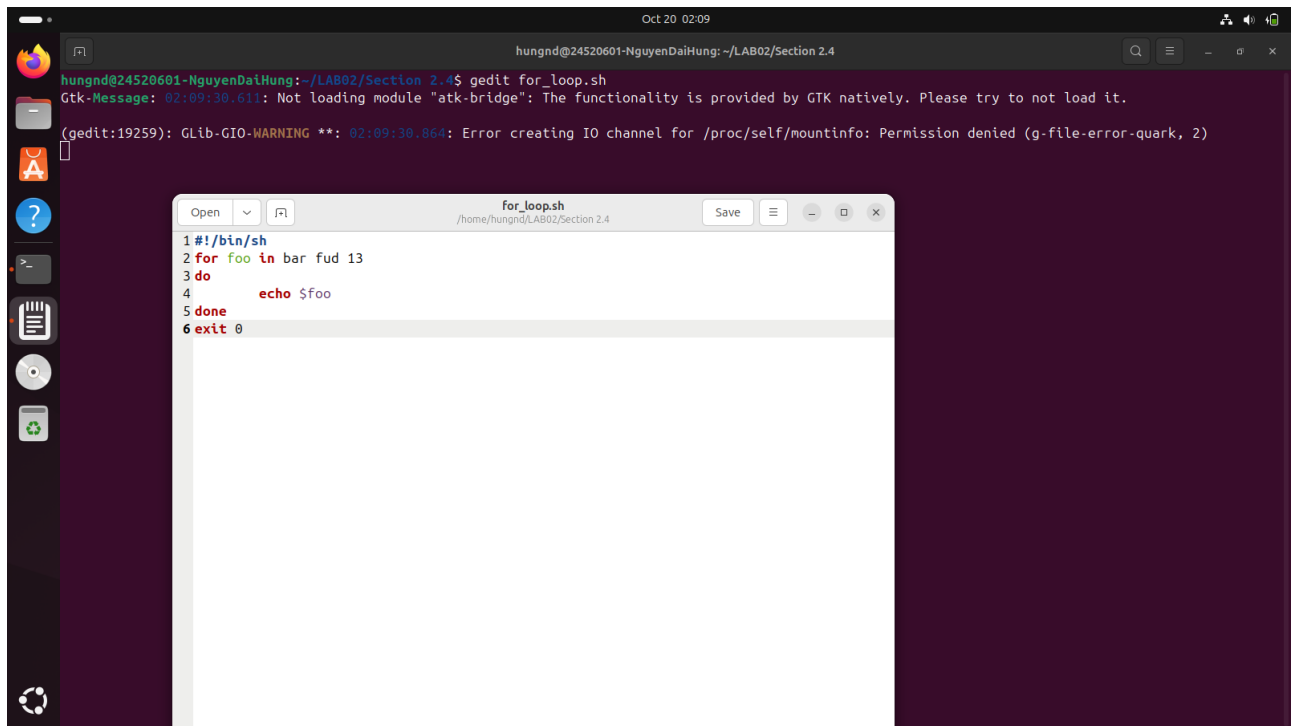


Hình 5.1: Nội dung script của elif\_control2.sh

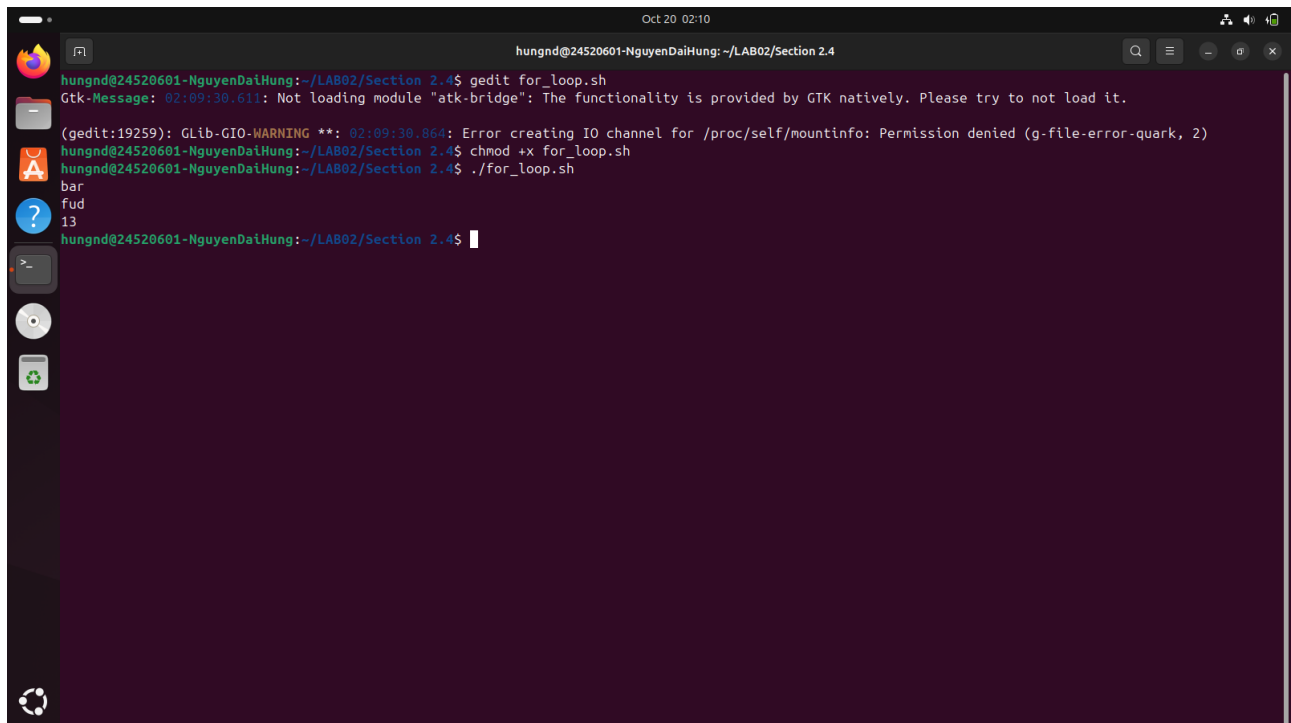


Hình 5.2: Kết quả khi thực thi script elif\_control2.sh

- Ví dụ 2-6: for\_loop.sh

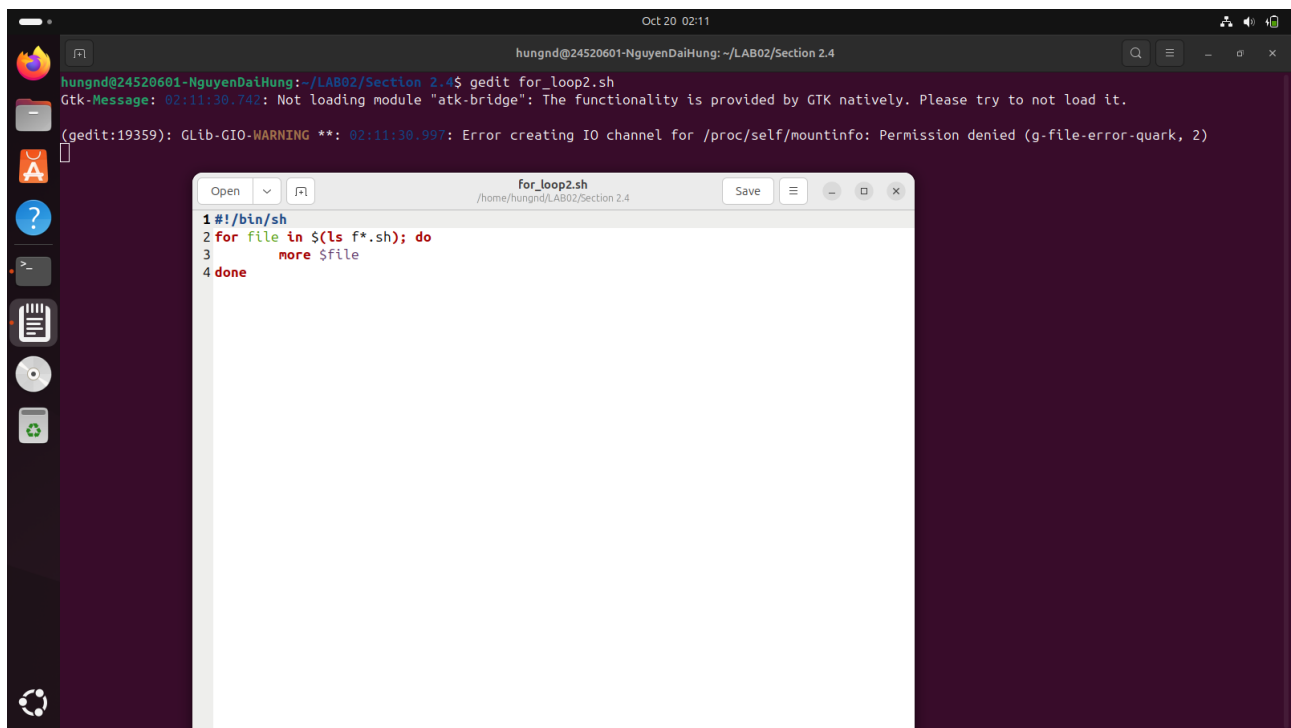


Hình 6.1: Nội dung script của `for_loop.sh`

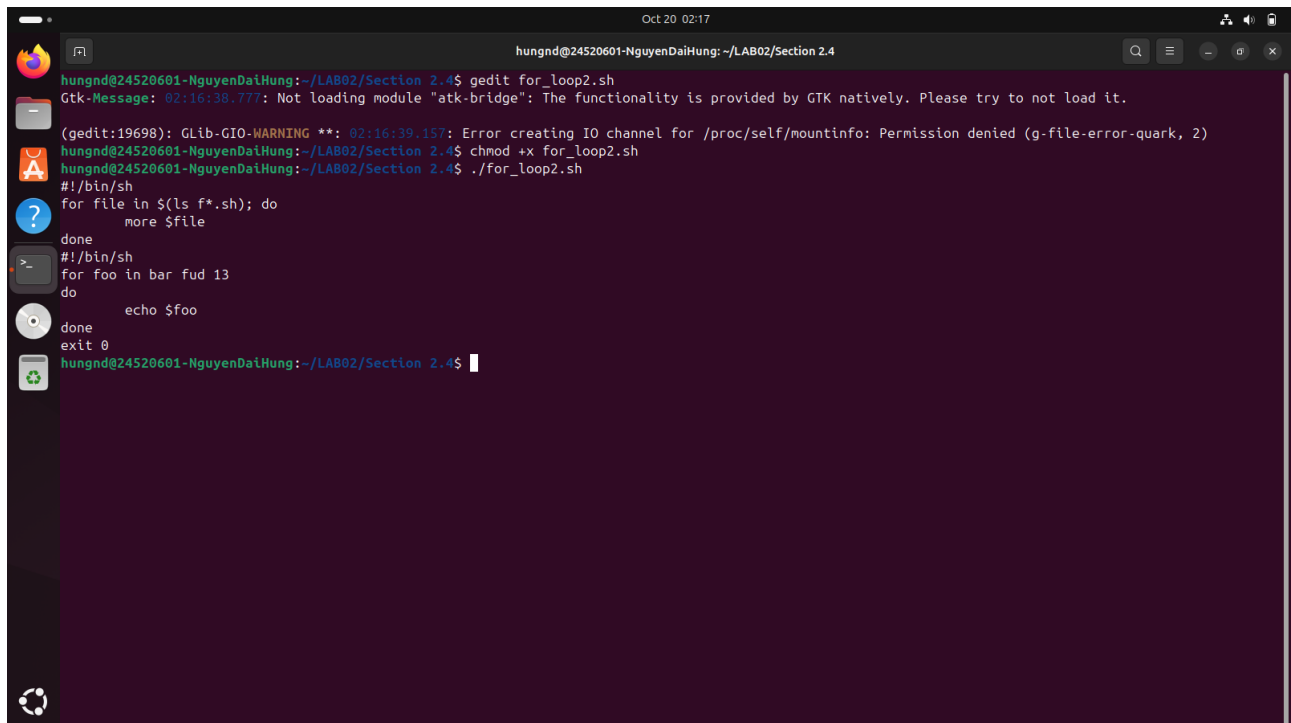


Hình 6.2: Kết quả khi thực thi script `for_loop.sh`

- Ví dụ 2-7: `for_loop2.sh`



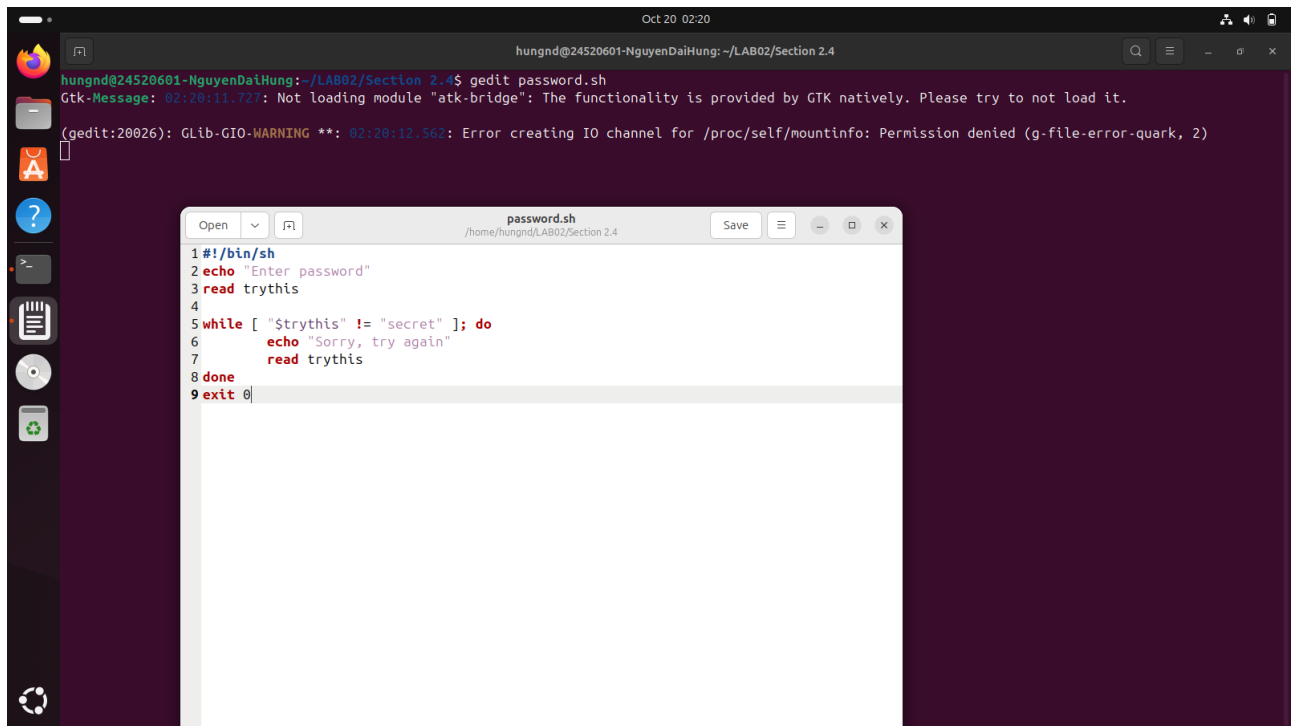
Hình 7.1: Nội dung script của `for_loop2.sh`



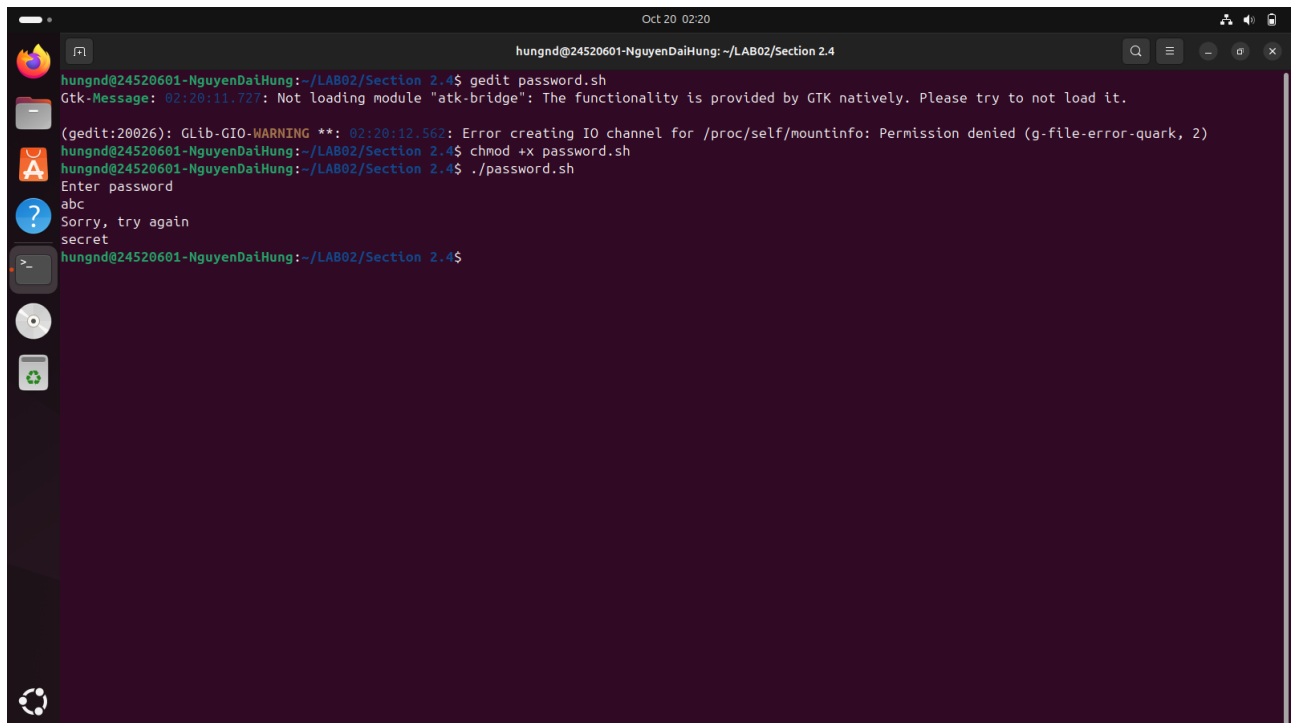
Hình 7.2: Kết quả khi thực thi script `for_loop2.sh`

- Ví dụ 2-8: `password.sh`



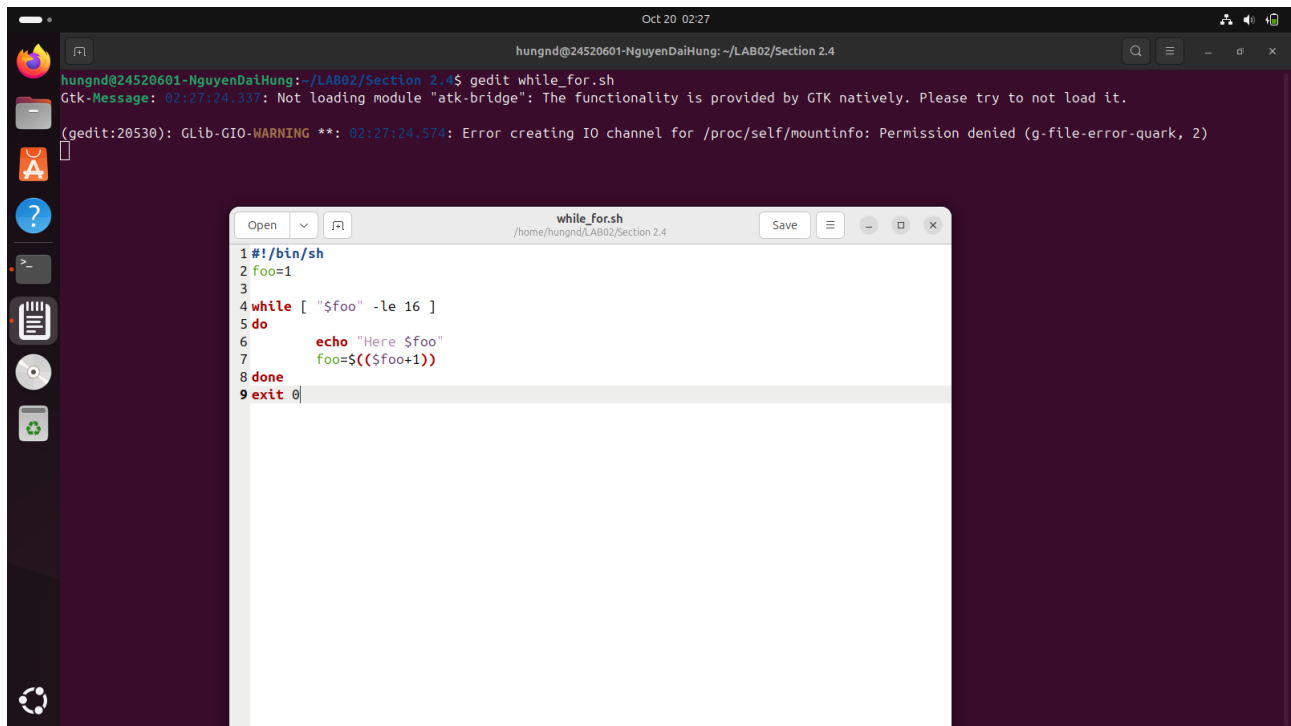


Hình 8.1: Nội dung script của password.sh

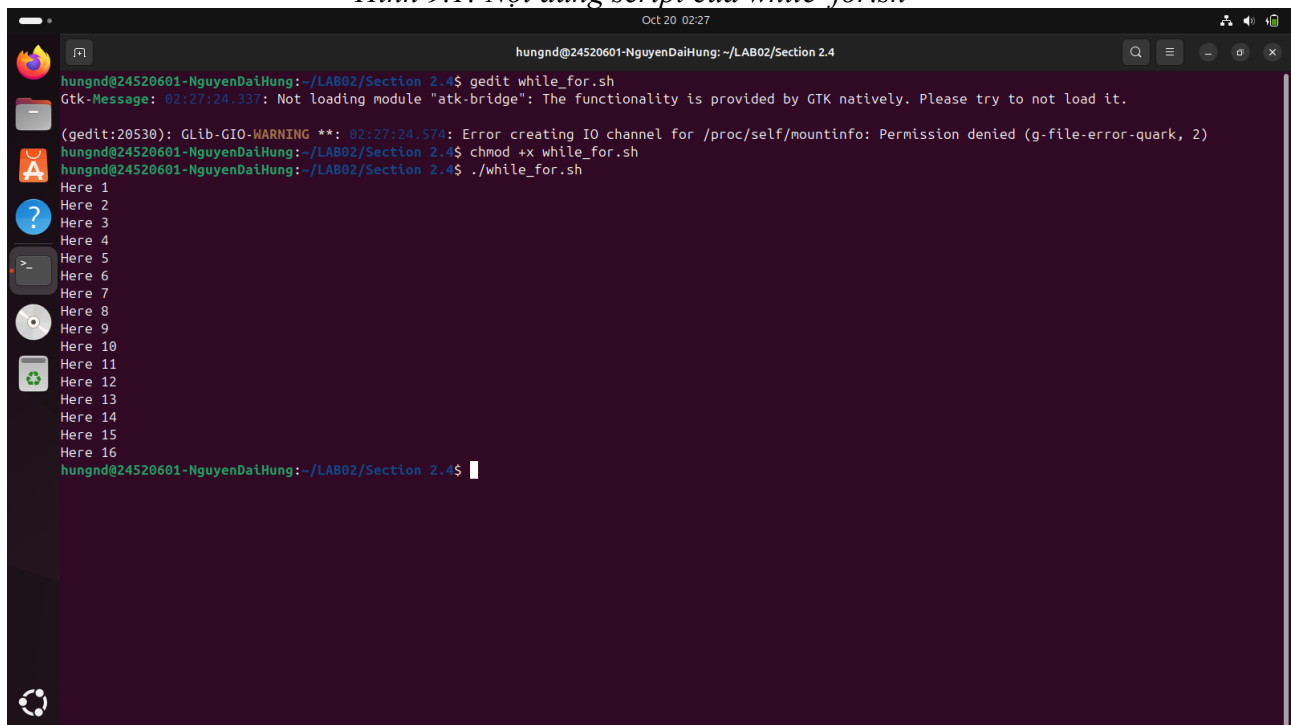


Hình 8.2: Kết quả khi thực thi script password.sh

- Ví dụ 2-9: while\_for.sh

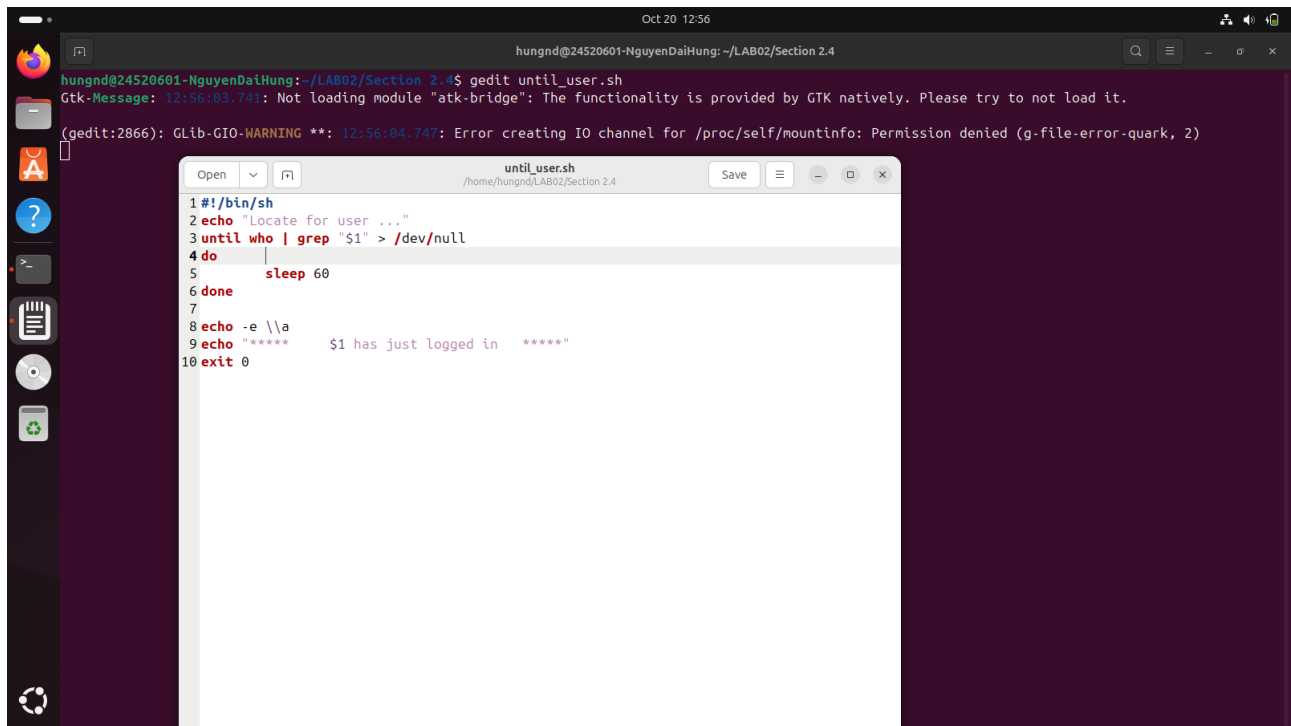


Hình 9.1: Nội dung script của `while_for.sh`

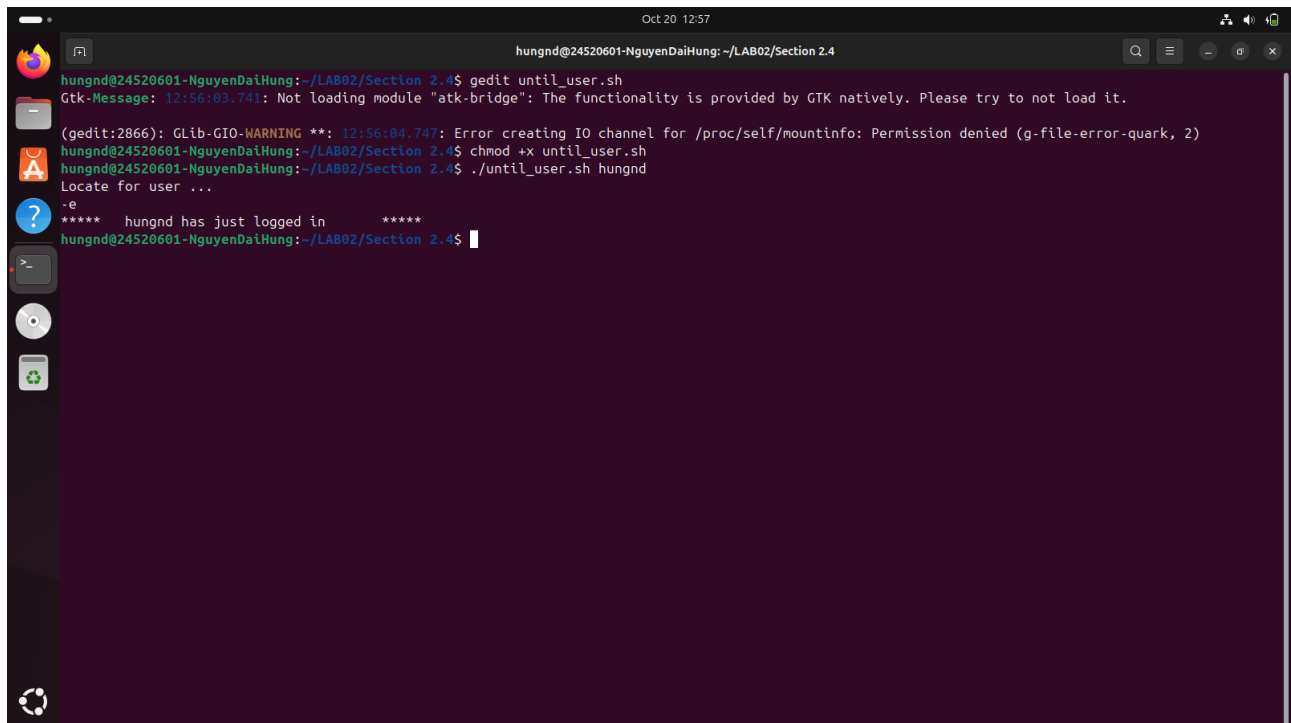


Hình 9.2: Kết quả khi thực thi script `while_for.sh`

- Ví dụ 2-10: `until_user.sh`

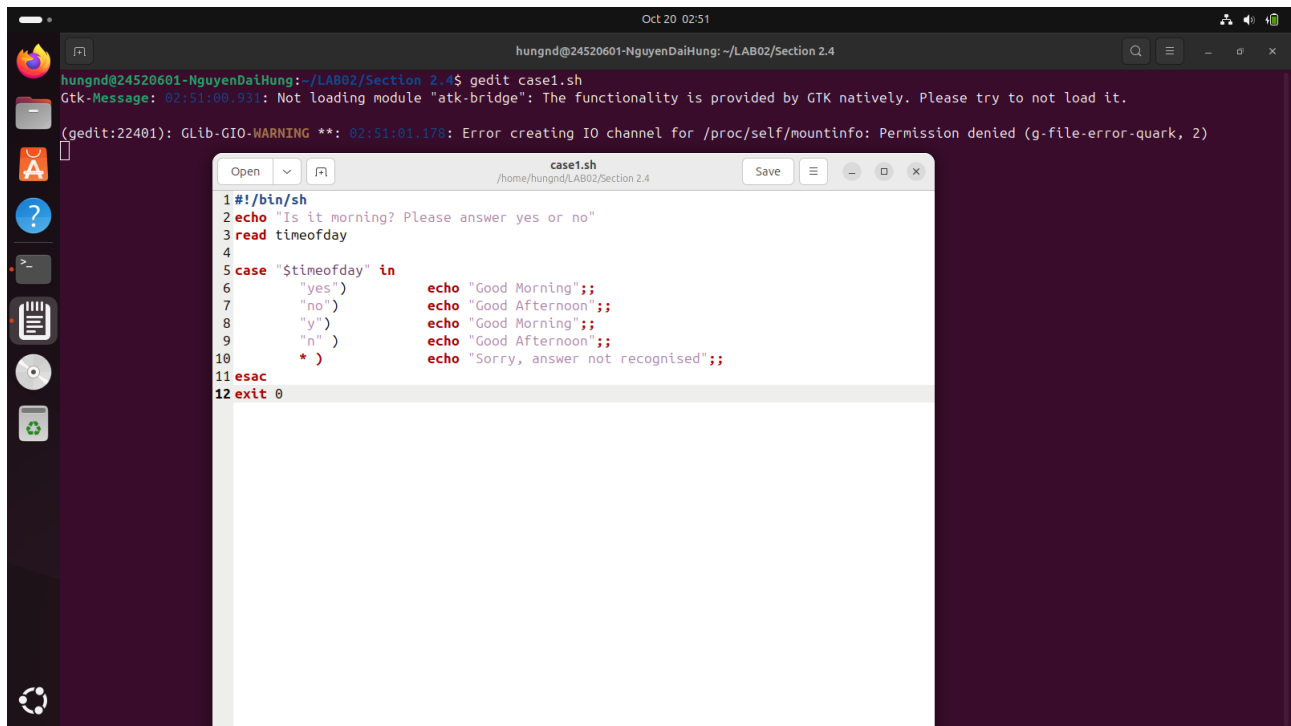


Hình 10.1: Nội dung script của `until_user.sh`

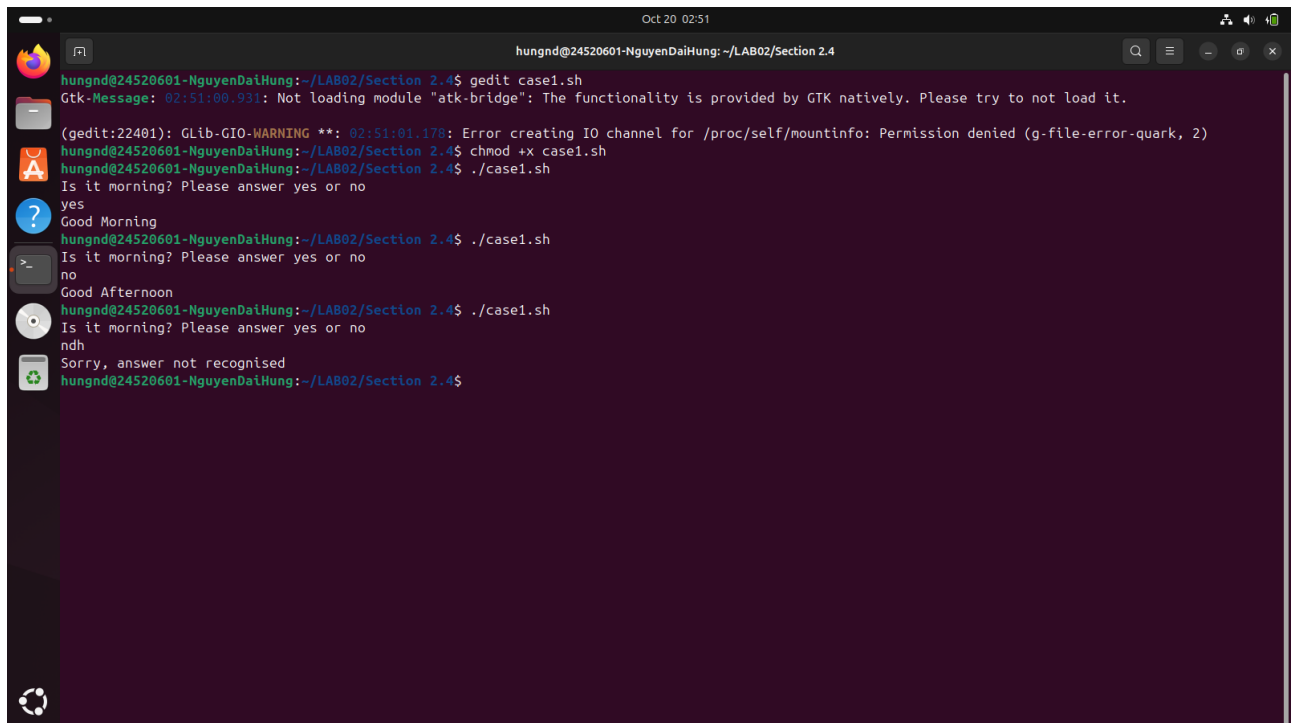


Hình 10.2: Kết quả khi thực thi script `until_user.sh`

- Ví dụ 2-11: `case1.sh`

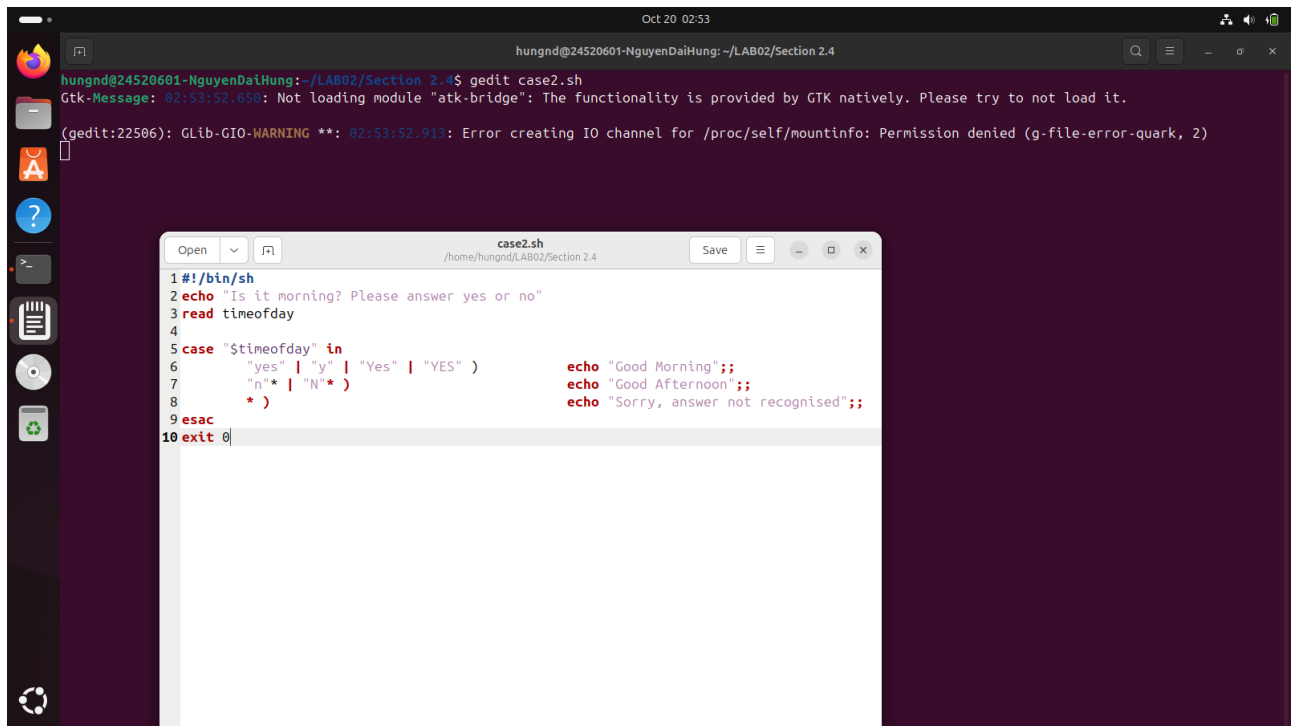


Hình 11.1: Nội dung script của case1.sh

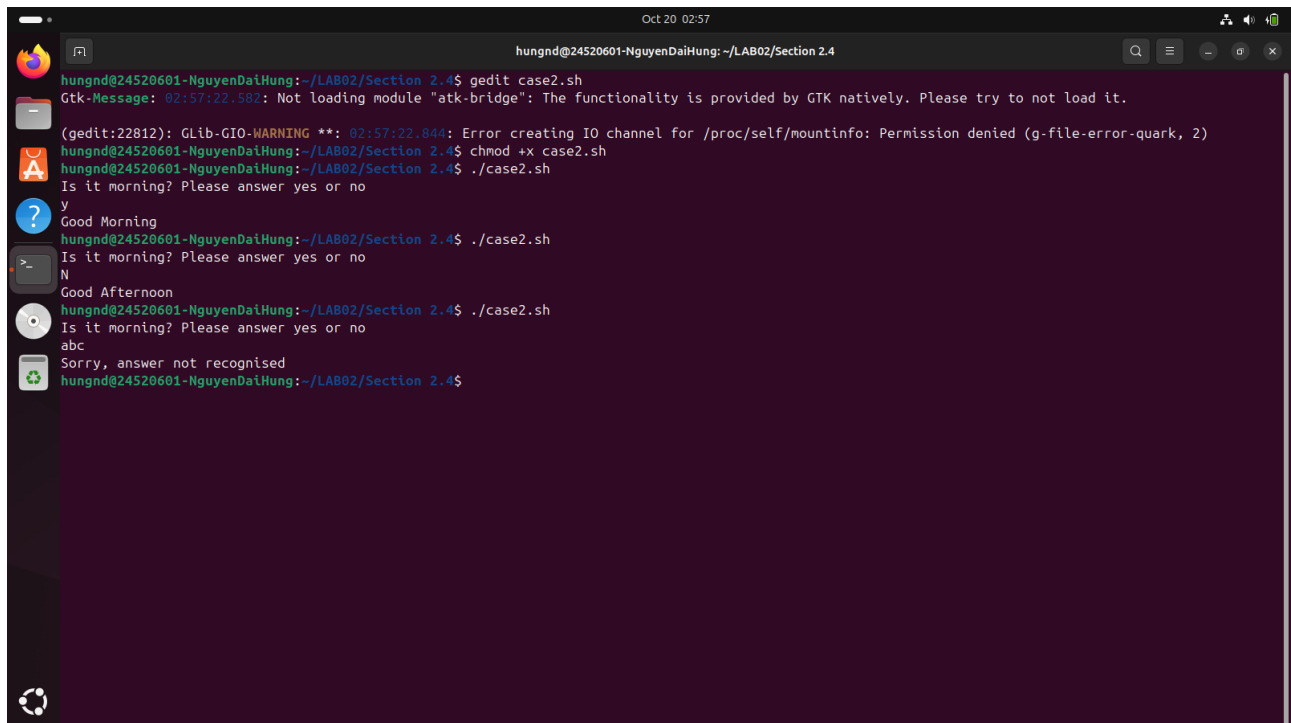


Hình 11.2: Kết quả khi thực thi script case1.sh

- Ví dụ 2-12: case2.sh

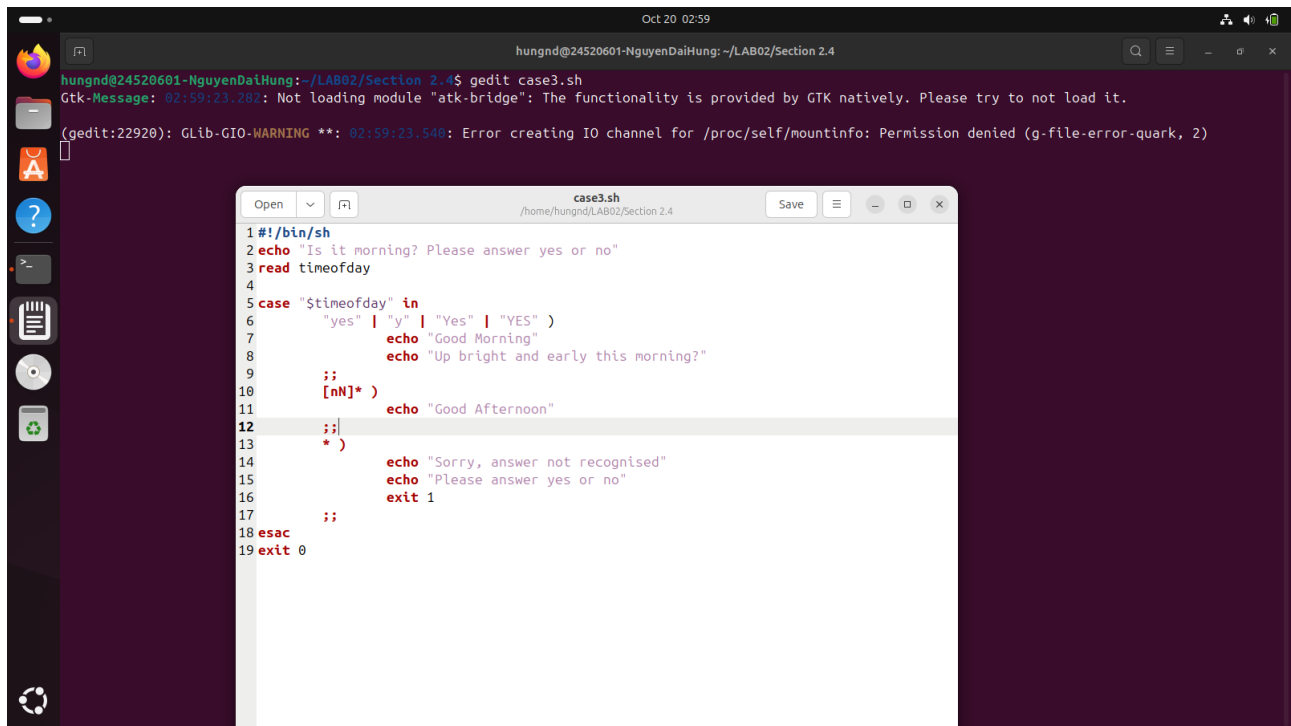


Hình 12.1: Nội dung script của case2.sh

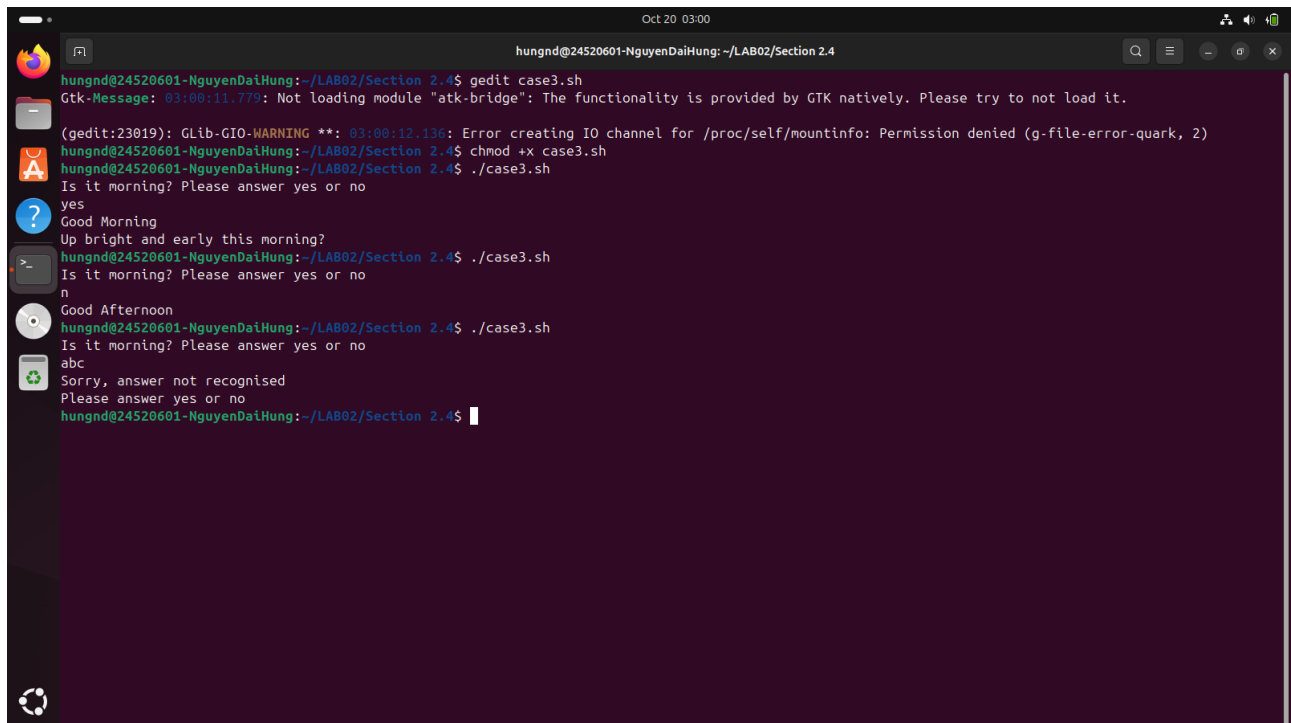


Hình 12.2: Kết quả khi thực thi script case2.sh

- Ví dụ 2-13: case3.sh



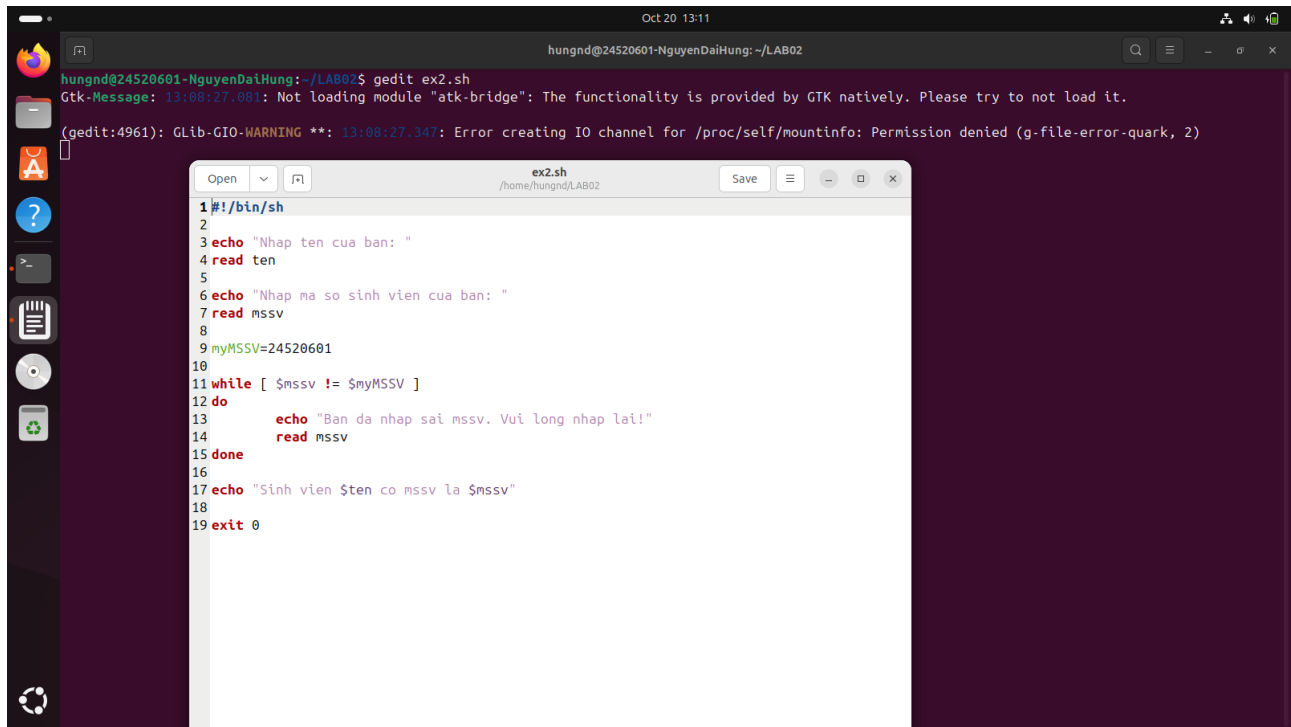
Hình 13.1: Nội dung script của case3.sh



Hình 13.2: Kết quả khi thực thi script case3.sh

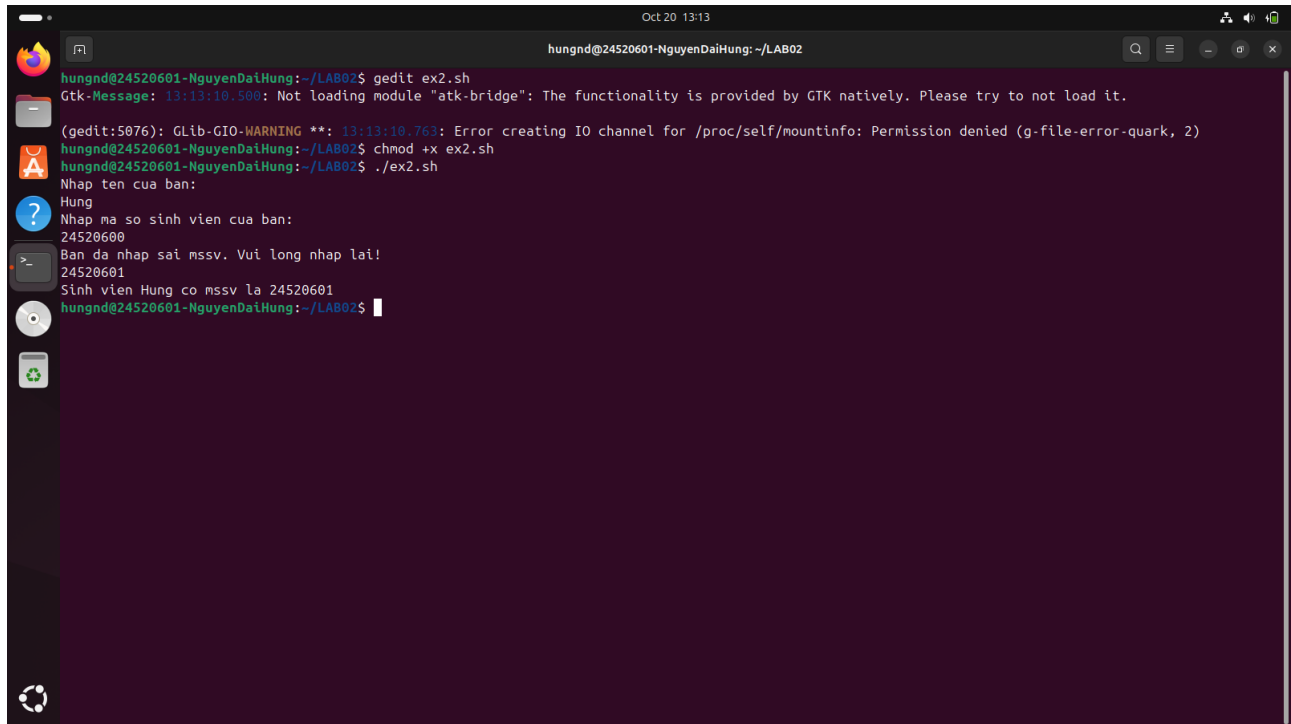
- Viết chương trình cho phép nhập vào tên và mssv. Kiểm tra nếu mssv đó không trùng với mình thì bắt nhập lại. In ra màn hình kết quả.

- Viết file script để thực thi đặt tên là ex.sh nằm trong thư mục LAB02.
- Để viết chương trình theo như yêu cầu bài 2 ta tạo file script có tên là ex2.sh bằng lệnh: `gedit ex2.sh`.



*Hình 14.1: Nội dung script của ex2.sh*

- Sau đó, cấp quyền thực thi cho script bằng lệnh: `chmod +x ex2.sh`. Cuối cùng, ta gọi thực thi script bằng lệnh: `./ex.sh`



```
Oct 20 13:13
hungnd@24520601-NguyenDaiHung: ~/LAB02
hungnd@24520601-NguyenDaiHung:~/LAB02$ gedit ex2.sh
Gtk-Message: 13:13:10.500: Not loading module "atk-bridge": The functionality is provided by GTK natively. Please try to not load it.
(hungnd@24520601-NguyenDaiHung:~/LAB02$ chmod +x ex2.sh
hungnd@24520601-NguyenDaiHung:~/LAB02$ ./ex2.sh
Nhap ten cua ban:
Hung
Nhap ma so sinh vien cua ban:
24520600
Ban da nhap sai mssv. Vui long nhap lai!
24520601
Sinh vien Hung co mssv la 24520601
hungnd@24520601-NguyenDaiHung:~/LAB02$
```

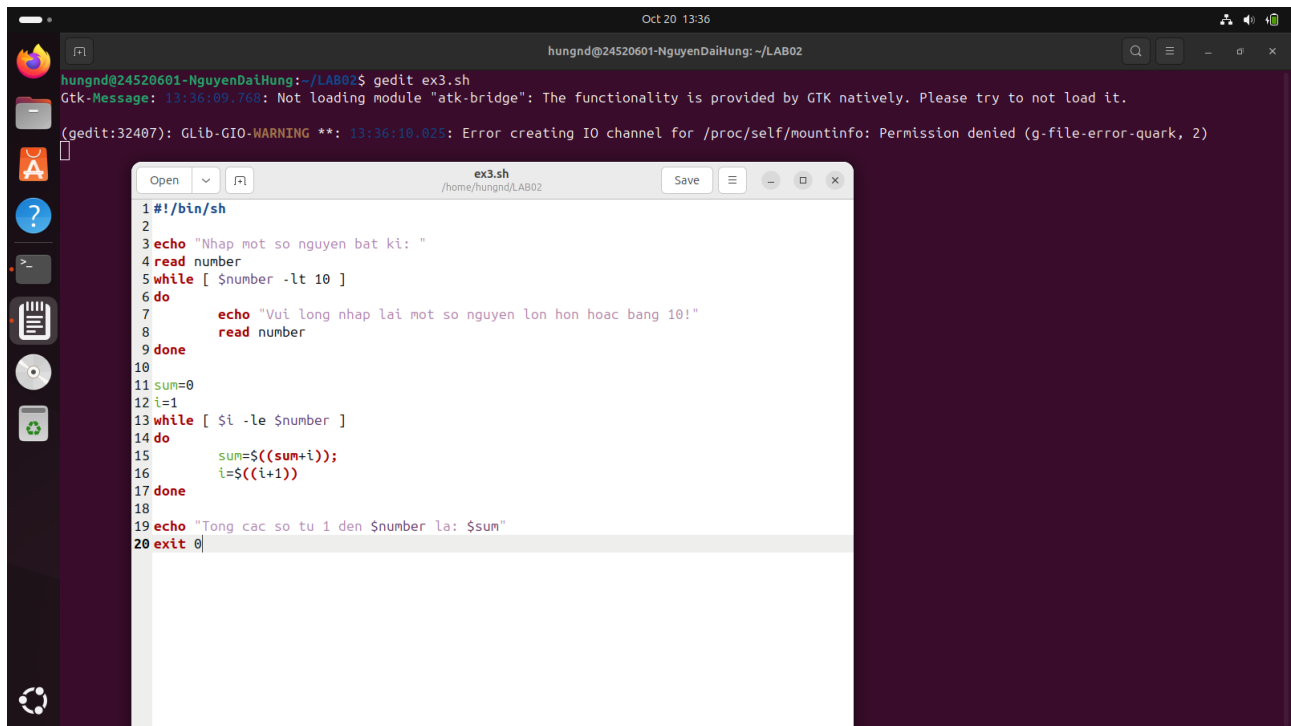
Hình 14.2: Kết quả khi thực thi script ex2.sh

- Để đảm bảo file script mà ta đã viết là đúng, ta có thể nhập một vài test-case để kiểm tra chương trình như sau: Dòng đầu tiên chương trình yêu cầu nhập tên, ta nhập **Hung** (có thể nhập tên khác). Dòng tiếp theo chương trình yêu cầu nhập mssv, ta nhập **24520600**, chương trình sẽ tiến hành kiểm tra và so sánh với “**24520601**” lúc này vì hai mssv này không giống nhau nên chương trình sẽ yêu cầu ta nhập lại.
- Ta nhập lại **24520601**, chương trình sẽ tiến hành kiểm tra và so sánh với “**24520601**” lúc này vì hai mssv này là giống nhau nên chương trình sẽ in ra màn hình dòng chữ “**Sinh vien Hung co MSSV la 24520601**” và kết thúc chương trình. Có nghĩa là chúng ta đã viết thành công chương trình theo như yêu cầu bài 2.

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

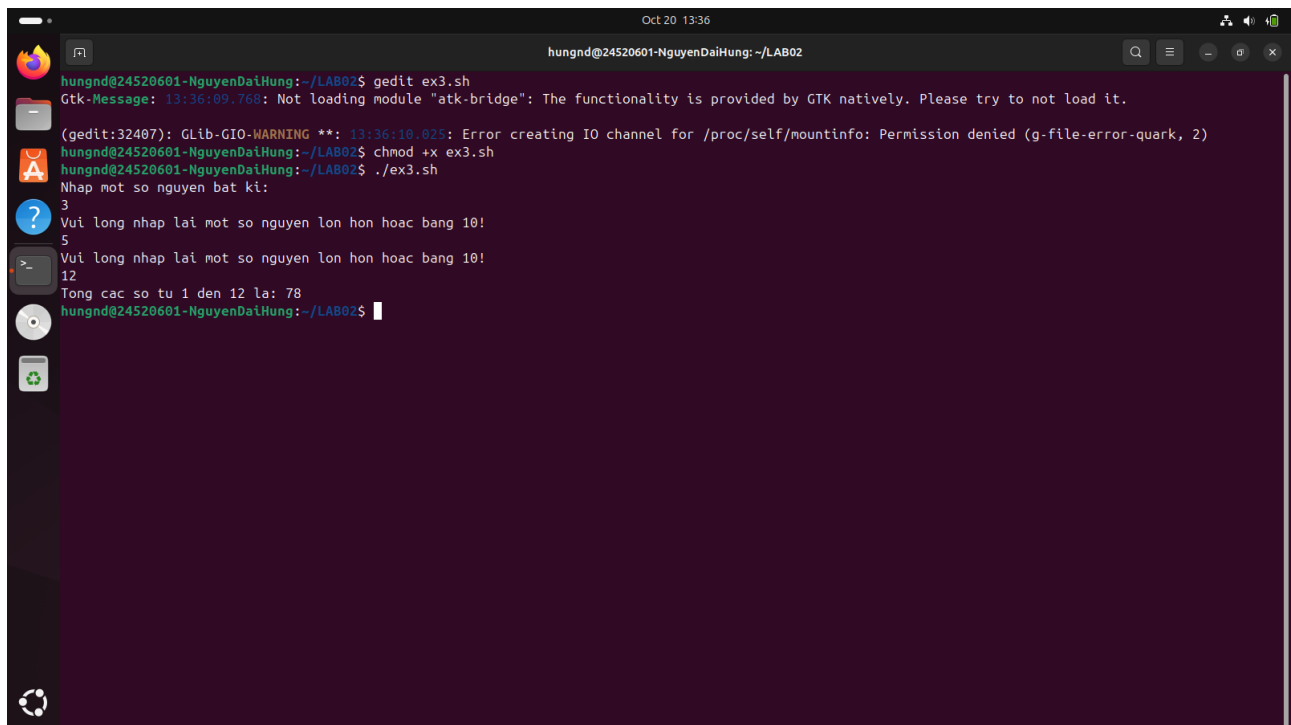
- Viết file script để thực thi đặt tên là ex3.sh nằm trong thư mục LAB02.
- Để viết chương trình theo như yêu cầu bài 3 ta tạo file script có tên là ex3.sh bằng lệnh: gedit ex3.sh





Hình 15.1: Nội dung script của ex3.sh

- Sau đó, cấp quyền thực thi cho script bằng lệnh: `chmod +x ex3.sh`. Cuối cùng, ta gọi thực thi script bằng lệnh: `./ex3.sh`

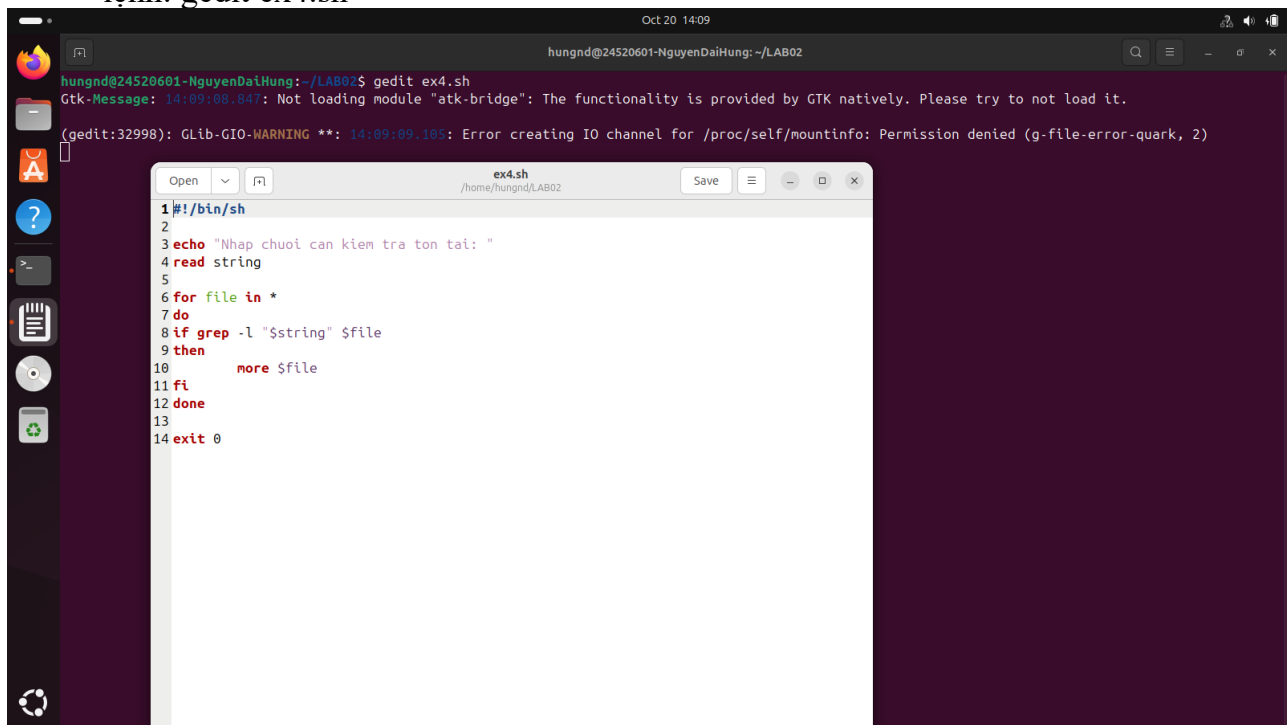


Hình 15.2: Kết quả khi thực thi script ex3.sh

- Để đảm bảo file script mà ta đã viết là đúng, ta có thể nhập một vài test-case để kiểm tra chương trình như sau: Dòng đầu tiên chương trình yêu cầu nhập một số nguyên bất kì, ta nhập **3**, chương trình sẽ tiến hành kiểm tra và so sánh với **10** lúc này vì  $3 < 10$  nên chương trình sẽ yêu cầu ta nhập lại. Ta nhập lại **5**, chương trình sẽ tiến hành kiểm tra và so sánh với **10** lúc này vì  $5 < 10$  nên chương trình sẽ yêu cầu ta nhập lại.
- Ta nhập lại **12**, chương trình sẽ tiến hành kiểm tra và so sánh với **10** lúc này vì  $12 = 10$  nên chương trình sẽ tính tổng các số từ 1 đến 12 và thu được kết quả là 78 nên chương trình sẽ in ra màn hình dòng chữ **“Tong cac so tu 1 den 12 la: 78”** và kết thúc chương trình. Có nghĩa là chúng ta đã viết thành công chương trình theo như yêu cầu bài 3.

#### 4. Viết trình cho phép nhập vào một chuỗi. Kiểm tra chuỗi đó có tồn tại trong một file text (ví dụ test.txt) cùng thư mục hay không.

- Viết file script để thực thi đặt tên là ex4.sh nằm trong thư mục LAB02.
- Để viết chương trình theo như yêu cầu bài 4 ta tạo file script có tên là ex4.sh bằng lệnh: `gedit ex4.sh`



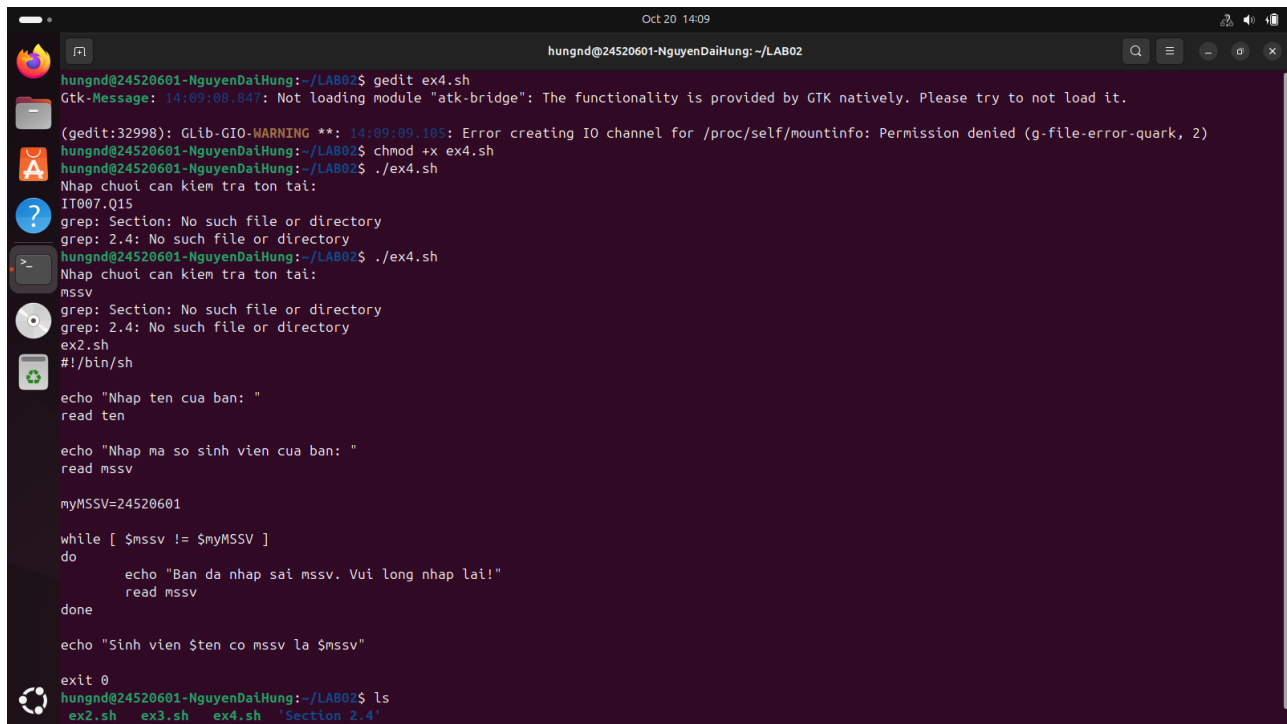
```

Oct 20 14:09
hungnd@24520601-NguyenDaiHung:~/LAB02
hungnd@24520601-NguyenDaiHung:~/LAB02$ gedit ex4.sh
Gtk-Message: 14:09:08.847: Not loading module "atk-bridge": The functionality is provided by GTK natively. Please try to not load it.
(gedit:32998): GLib-GIO-WARNING **: 14:09:09.105: Error creating IO channel for /proc/self/mountinfo: Permission denied (g-file-error-quark, 2)

ex4.sh
#!/bin/sh
2
3 echo "Nhap chuỗi cần kiểm tra tồn tại: "
4 read string
5
6 for file in *
7 do
8 if grep -l "$string" $file
9 then
10     more $file
11 fi
12 done
13
14 exit 0
  
```

Hình 16.1: Nội dung script của ex4.sh

- Sau đó, cấp quyền thực thi cho script bằng lệnh: `chmod +x ex4.sh`. Cuối cùng, ta gọi thực thi script bằng lệnh: `./ex4.sh`



```
Oct 20 14:09
hungnd@24520601-NguyenDaiHung: ~/LAB02
hungnd@24520601-NguyenDaiHung:~/LAB02$ gedit ex4.sh
Gtk-Message: 14:09:08.847: Not loading module "atk-bridge": The functionality is provided by GTK natively. Please try to not load it.
(hungnd@24520601-NguyenDaiHung:~/LAB02$ chmod +x ex4.sh
hungnd@24520601-NguyenDaiHung:~/LAB02$ ./ex4.sh
Nhap chuoi can kiem tra ton tai:
IT007.Q15
grep: Section: No such file or directory
grep: 2.4: No such file or directory
hungnd@24520601-NguyenDaiHung:~/LAB02$ ./ex4.sh
Nhap chuoi can kiem tra ton tai:
mssv
grep: Section: No such file or directory
grep: 2.4: No such file or directory
ex2.sh
#!/bin/sh
echo "Nhap ten cua ban: "
read ten
echo "Nhap ma so sinh vien cua ban: "
read mssv
myMSSV=24520601
while [ $mssv != $myMSSV ]
do
    echo "Ban da nhap sai mssv. Vui long nhap lai!"
    read mssv
done
echo "Sinh vien $ten co mssv la $mssv"
exit 0
hungnd@24520601-NguyenDaiHung:~/LAB02$ ls
ex2.sh  ex3.sh  ex4.sh  'Section 2.4'
```

Hình 16.2: Kết quả khi thực thi script *ex4.sh*

- Để đảm bảo file script mà ta đã viết là đúng, ta có thể nhập một vài test-case để kiểm tra chương trình như sau: Khi thực thi chương trình ở lần đầu tiên, chương trình yêu cầu ta nhập chuỗi, ta nhập **IT007.Q15**, chương trình sẽ tiến hành tìm và kiểm tra nội dung của tất cả các file trong cùng thư mục và không tìm thấy file nào có nội dung chứa chuỗi **IT007.Q15** nên chương trình không xuất ra màn hình kết quả gì mà kết thúc chương trình luôn.
- Khi thực thi chương trình ở lần thứ hai, chương trình yêu cầu nhập chuỗi, ta nhập **mssv**, chương trình sẽ tiến hành tìm và kiểm tra nội dung của tất cả các file trong cùng thư mục và tìm thấy file **ex2.sh** có nội dung chứa chuỗi **mssv** nên đã xuất kết quả là nội dung của file **ex2.sh** ra màn hình.
- Dùng lệnh **ls** để kiểm tra xem có tồn tại file **ex2.sh** trong cùng thư mục (thư mục hiện hành) hay không, lúc này ta thấy có tồn tại file **ex2.sh** trong thư mục hiện hành. Có nghĩa là chúng ta đã viết thành công chương trình theo như yêu cầu bài 4.