|  |
| --- |
| Họ và tên: Đồng Quốc Thắng  Mã số sinh viên: 23521421  Lớp: IT007.P11.CTTN |

HỆ ĐIỀU HÀNH  
BÁO CÁO LAB 2

**CHECKLIST (Đánh dấu x khi hoàn thành)**

**Lưu ý mỗi câu phải làm đủ 3 yêu cầu**

1. **CLASSWORK**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **BT 1** | **BT 2** | **BT 3** | **BT 4** | **BT 5** |
| **Trình bày cách làm** |  |  |  |  |  |
| **Chụp hình minh chứng** |  |  |  |  |  |
| **Giải thích kết quả** |  |  |  |  |  |

1. **HOMEWORK**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **BT1** | **BT2** | **BT3** | **BT4** | **BT5** | **BT6** | **BT7** |
| **Trình bày cách làm** |  |  |  |  |  |  |  |
| **Chụp hình minh chứng** |  |  |  |  |  |  |  |
| **Giải thích kết quả** |  |  |  |  |  |  |  |

1. **BONUS**

|  |  |  |
| --- | --- | --- |
|  | **BT1** | **BT2** |
| **Trình bày cách làm** |  |  |
| **Chụp hình minh chứng** |  |  |
| **Giải thích kết quả** |  |  |

**Tự chấm điểm:**  10

*\*Lưu ý: Xuất báo cáo theo định dạng PDF, đặt tên theo cú pháp:* ***<MSSV>\_LABx.pdf***

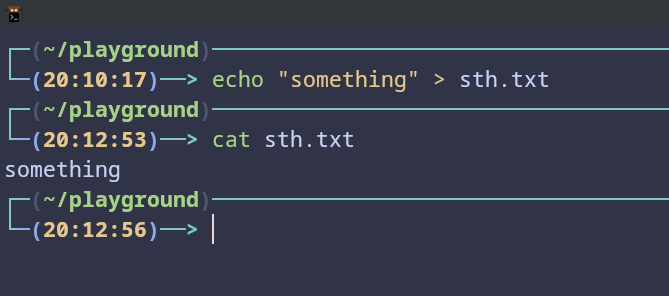
1. **CLASSWORK**

# Classwork 01

Chuyển hướng đầu ra: >

Cách làm: Gõ dấu “>” sau 1 câu lệnh và gõ tên file sau dấu > để chuyển hướng đầu ra của câu lệnh vô file đó

Ví dụ: echo “something” > sth.txt

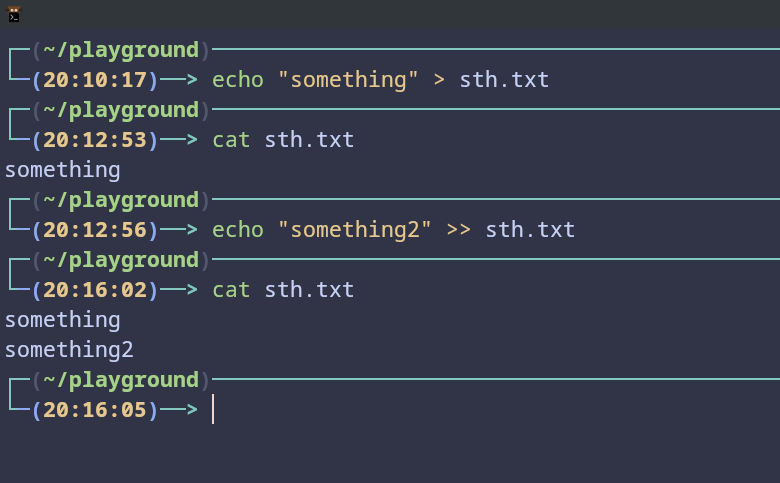


Giải thích: dấu “>” sẽ chuyển hướng từ stdout đến file mà mình chọn, ở đây chọn là file sth.txt. Câu lệnh echo bình thường sẽ in ra stdout mà ở đây ta dùng “>” nên file sth.txt giờ đây có chữ something.

Nối thêm đầu ra: >>

Cách làm: Gõ dấu “>>” sau 1 câu lệnh và gõ tên file sau dấu >> để nối thêm đầu ra của câu lệnh vô file đó

Ví dụ: echo “something2” >> sth.txt

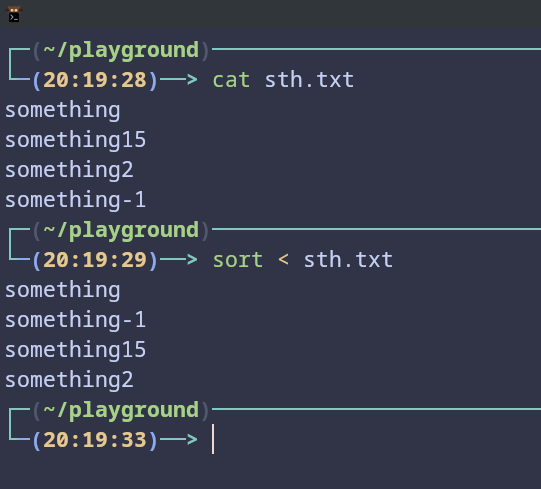


Giải thích: dấu “>>” sẽ append từ stdout đến file mà mình chọn, ở đây chọn là file sth.txt. Câu lệnh echo bình thường sẽ in ra stdout mà ở đây ta dùng “>>” nên file sth.txt giờ đây có thêm dòng something2.

Chuyển hướng đầu vào: <

Cách làm: Gõ dấu “<” sau 1 câu lệnh và gõ tên file sau dấu < để chuyển hướng đầu vào file đó vào câu lệnh( câu lệnh lấy input là content của file đó)

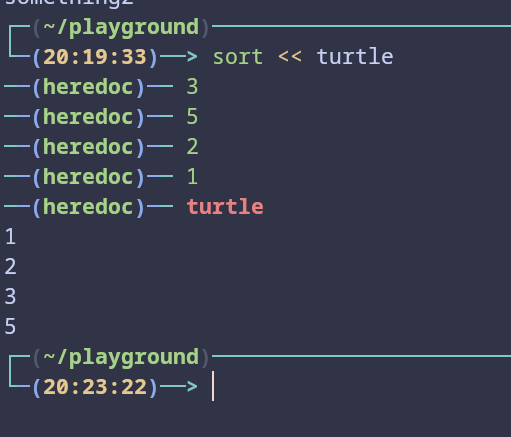
Ví dụ: sort < sth.txt



Giải thích: dấu “<” sẽ chuyển hướng từ stdin của 1 file vào câu lệnh. Nên sau khi chuyển hướng đầu vào của file vào câu lệnh sort. Câu lệnh sort sẽ in ra content của file sau khi được sort

Here document: <<

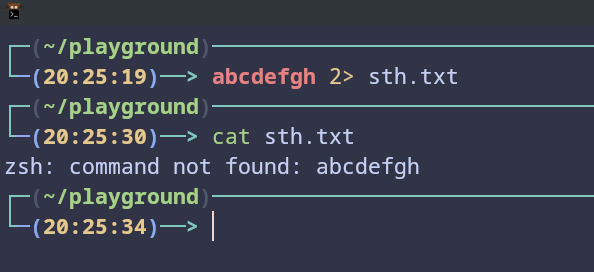
Cách làm: Gõ dấu “<<” sau 1 delimeter và sau đó gõ các dòng cho câu lệnh đó. Việc gõ vào sẽ dừng khi gõ tới delimeter



Giải thích: ở đây chọn delimeter là turtle, hàm sort sẽ sort các số được nhập vào.

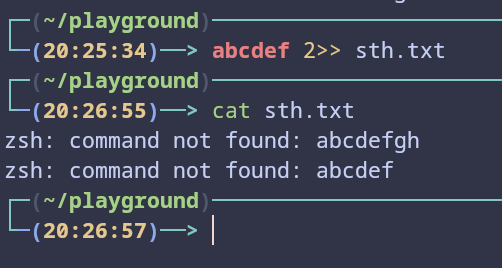
Chuyển hướng lỗi: 2>

Tương tự như > nhưng mà ở đây thay vì là chuyển stdout vào file thì ta sẽ chuyển stderr vào file



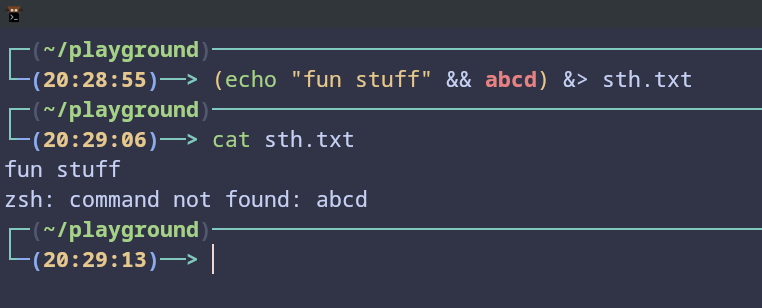
Giải thích: ở đây gõ 1 lệnh không tồn tại trong máy là abcdefgh, và chuyển hướng lỗi vào file sth.txt. Ta sẽ không thấy báo lỗi ngoài terminal, thay vì đó nó sẽ báo lỗi vào trong file sth.txt

Nối thêm lỗi: 2>>



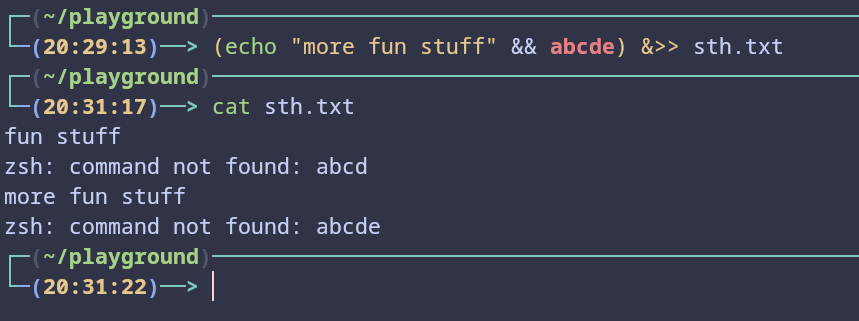
Giải thích: tương tự như >>, nhưng 2>> sẽ nối thêm lỗi vào file chứ không xóa hết content của file rồi mới viết vào như 2>

Chuyển hướng cả đầu ra và lỗi: &>



Giải thích: ở đây sẽ chuyển hướng cả đầu ra( từ command echo) và chuyển hướng lỗi( từ command không tồn tại abcd) vào file sth.txt. Cần phải có dấu ngoặc, nếu không nó chỉ hiểu là thực hiện command echo, sau đó thực hiện command abcd và chuyển hướng đầu ra và lỗi vào trong file sth.txt

Nối thêm cả đầu ra và lỗi: &>>



Giải thích: &>> thực hiện giống &> nhưng sẽ nối vào file thay vì xóa file và viết từ đầu.

# Classwork 02

Giải thích command

ps aux | grep apache | awk '{print $2}' | xargs kill -9

ps aux là để hiện ra tất cả các process, trong đó:

a = show tất cả các process của các user  
u = show user/owner  
x = show process không thuộc terminal đó

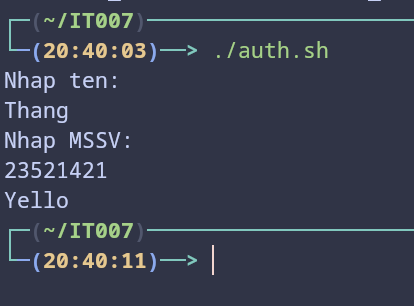
grep apache để lấy dòng có chữ apache.

awk ‘{print $2}’ là để in ra các dòng có chữ apache sao cho chỉ lấy cột thứ 2( lấy pid)

xargs kill -9 là để force kill pid vừa lấy được ở trên.

# Classwork 03

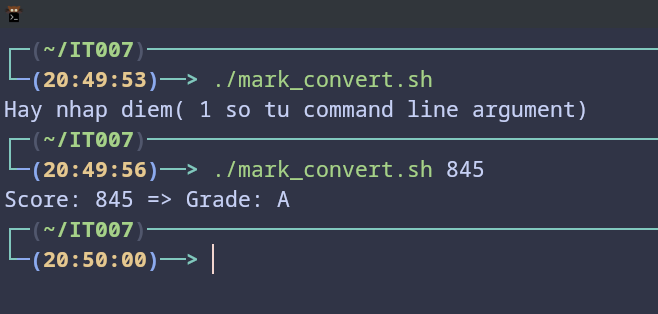
Cách làm: dùng read để nhập vào thông tin, so sánh và rẽ nhánh đơn giản bằng if để so thông tin nhập vào với thông tin đã có sẵn để kiểm tra người dùng.

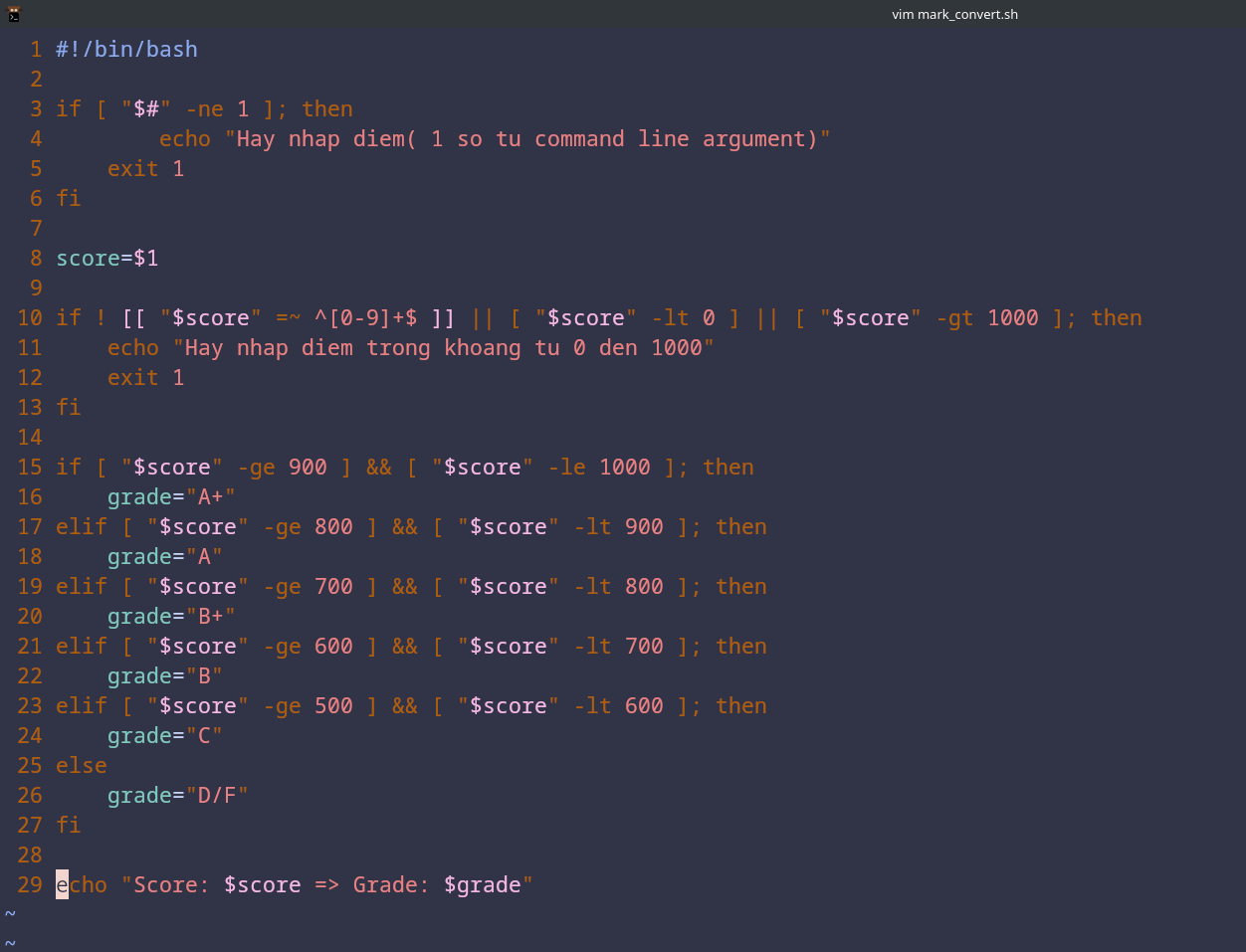


Giải thích: script sẽ dùng #! hay shebang để chỉ đến /bin/bash để interpret script. Sau đó dùng chmod +x auth.sh để cho quyền execute và dùng ./auth.sh để chạy script. Script này chỉ có dùng if và so sánh string cơ bản.

# Classwork 04

Cách làm: Kiểm tra chỉ có 1 argument là điểm, kiểm tra điểm chỉ là số từ khoảng 0 đến 1000, và sử dụng if else để chuyển điểm số sang điểm chữ

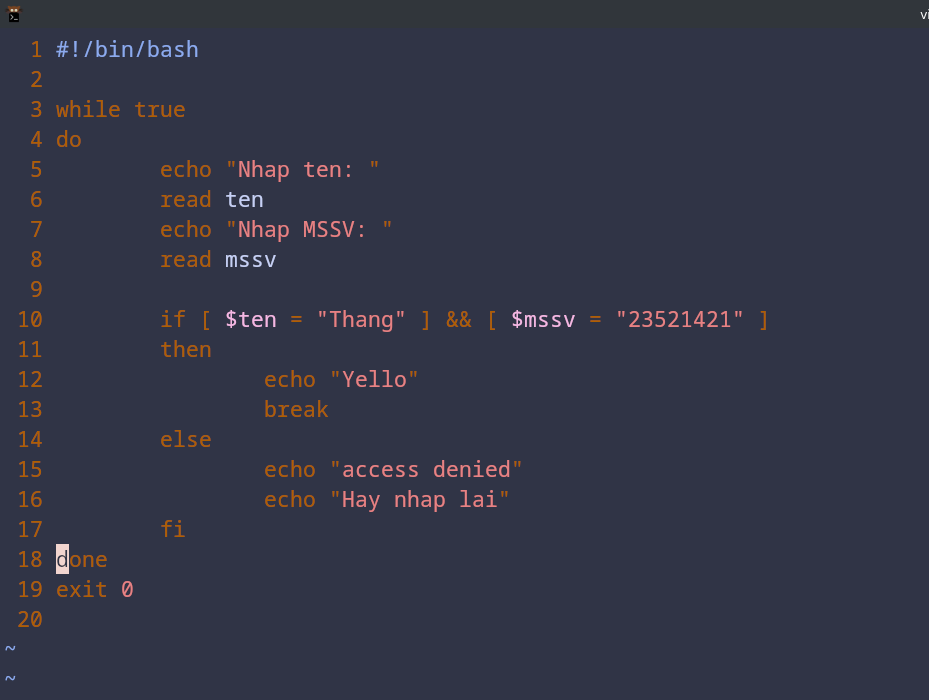


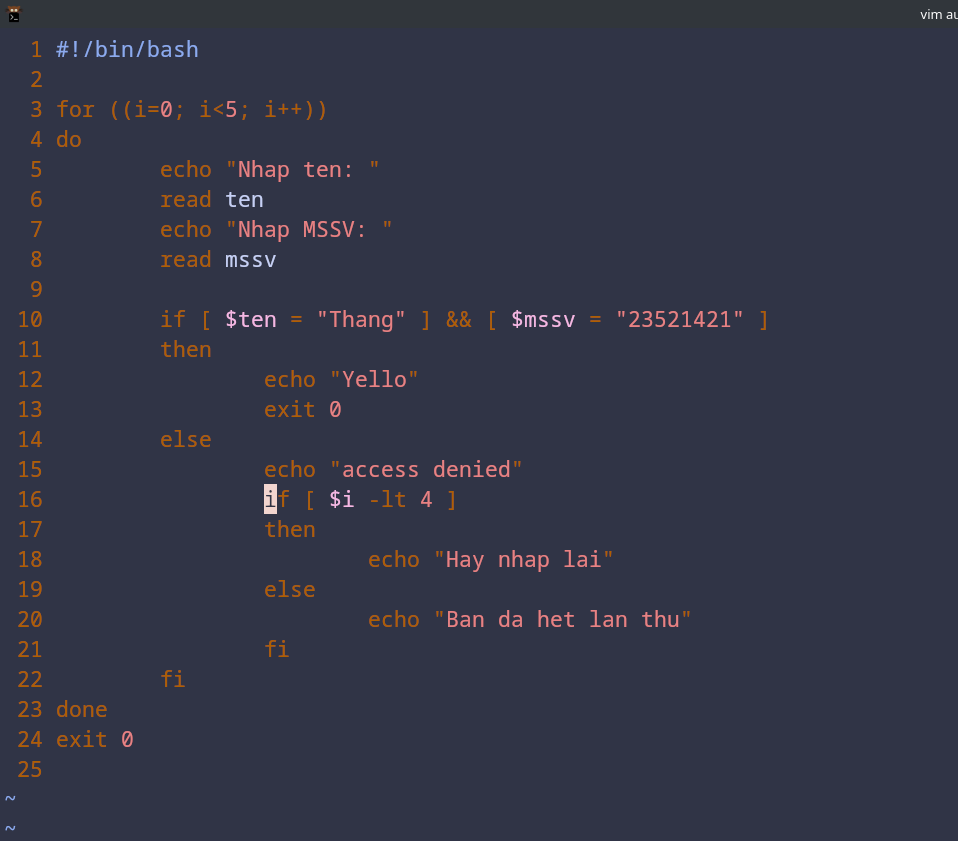


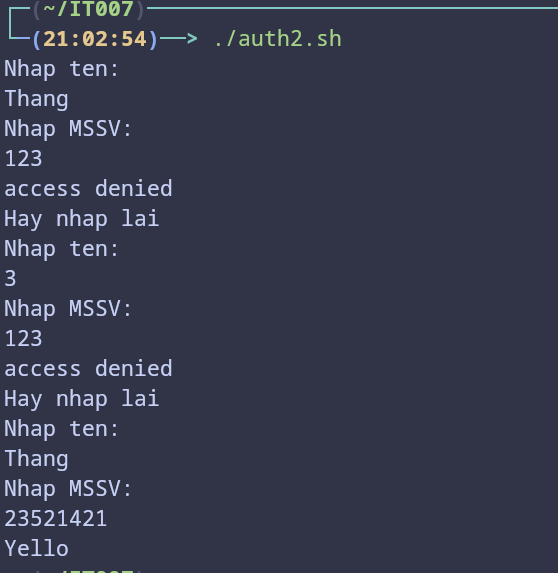
Giải thích: Lần đầu tiên chỉ chạy script mà không viết gì thêm nên không có command line argument => script không chạy được. Lần thứ 2 chạy với argument 845 nên đã đưa ra được kết quả là điểm A

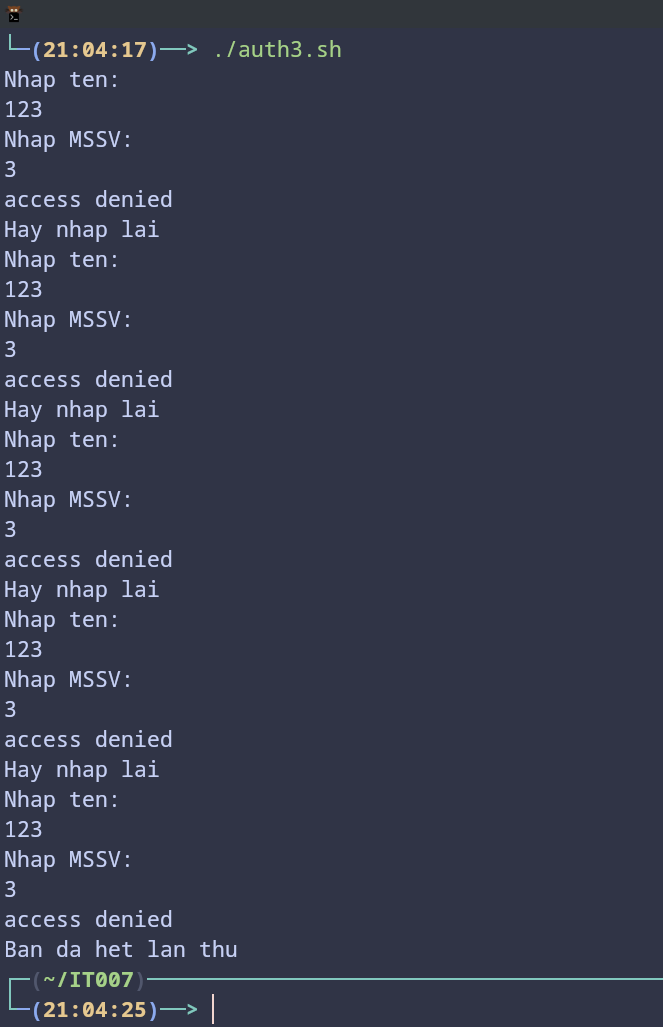
# Classwork 05

Cách làm: dùng for loop theo kiểu C cho yêu cầu thứ 2 và while loop( dùng while true), break khi người dùng nhập đúng( cho yêu cầu thứ nhất)







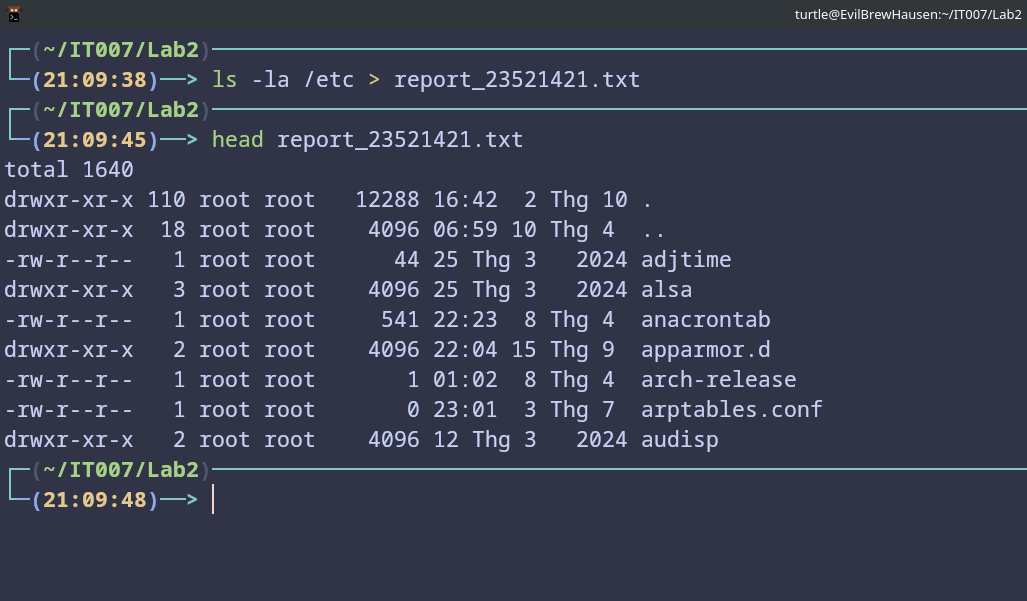


Giải thích: 2 ảnh đầu là code cho 2 script được yêu cầu. Ảnh thứ 3 là khi dùng vòng lặp while loop, chương trình sẽ hỏi đến khi nhập đúng. Ảnh thứ 4 là khi dùng vòng lặp for loop và chỉ có 5 lần thử.

1. **HOMEWORK**

# Homework 01

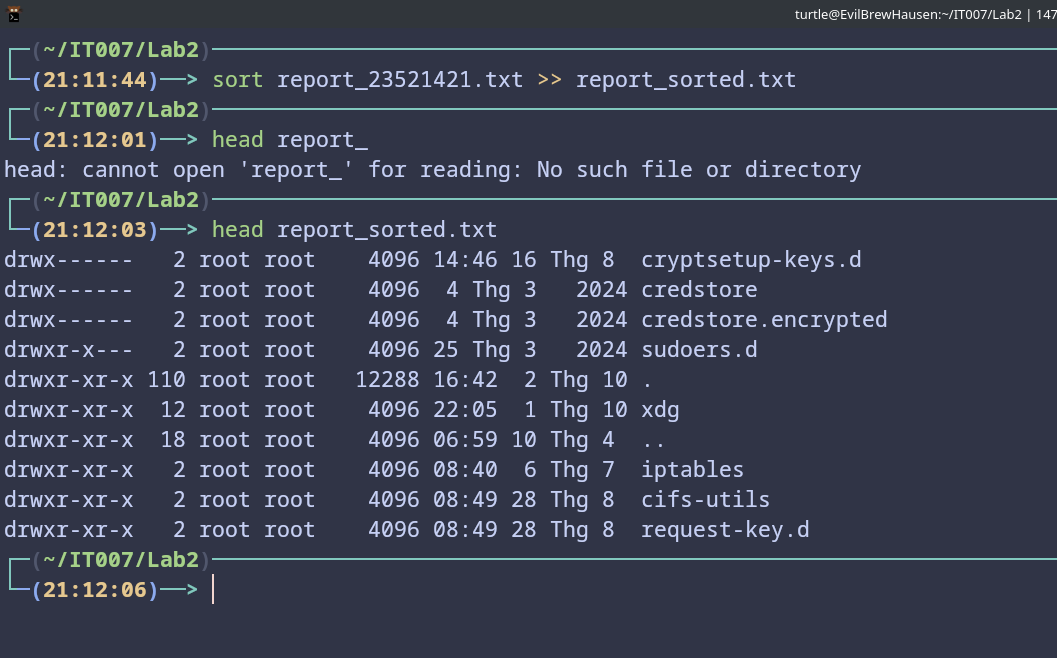
Cách làm: dùng chuyển hướng đầu vào >



Giải thích: dùng ls -la để in ra chi tiết chế độ của tất cả các file, sau đó redirect về file report\_23521421.txt và dùng command head để lấy ra các dòng đầu tiên trong file.

# Homework 02

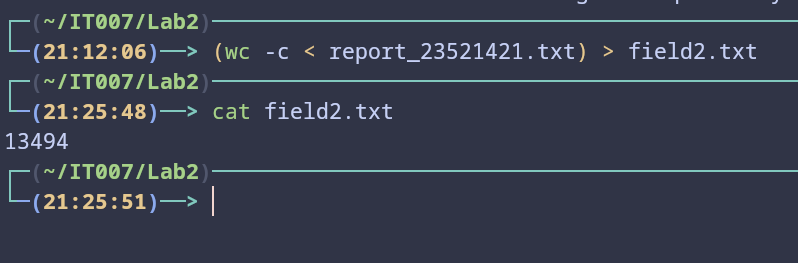
Cách làm: dùng hàm sort và chuyển hướng nối >>



Giải thích: dùng sort report\_23521421.txt để sort file đó, và sử dụng >> report\_sorted.txt để chuyển hướng nối vào file report\_sorted.txt

# Homework 03

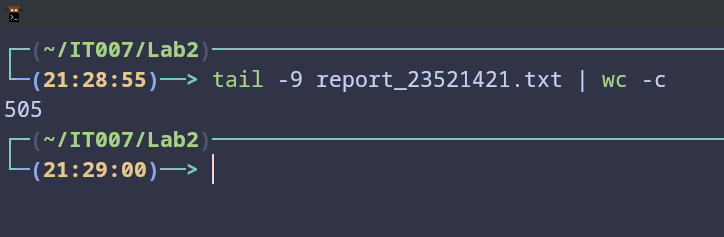
Cách làm: dùng command wc và chuyển hướng đầu vào, đầu ra( <, >)



Giải thích: dùng command wc -c để tìm số từ trong file report, ta chuyển hướng đầu vào của file report vào command wc -c để lấy số từ trong file. Sau đó chuyển hướng đầu ra vào file field2.txt. Kết quả là có 13494 từ.

# Homework 04

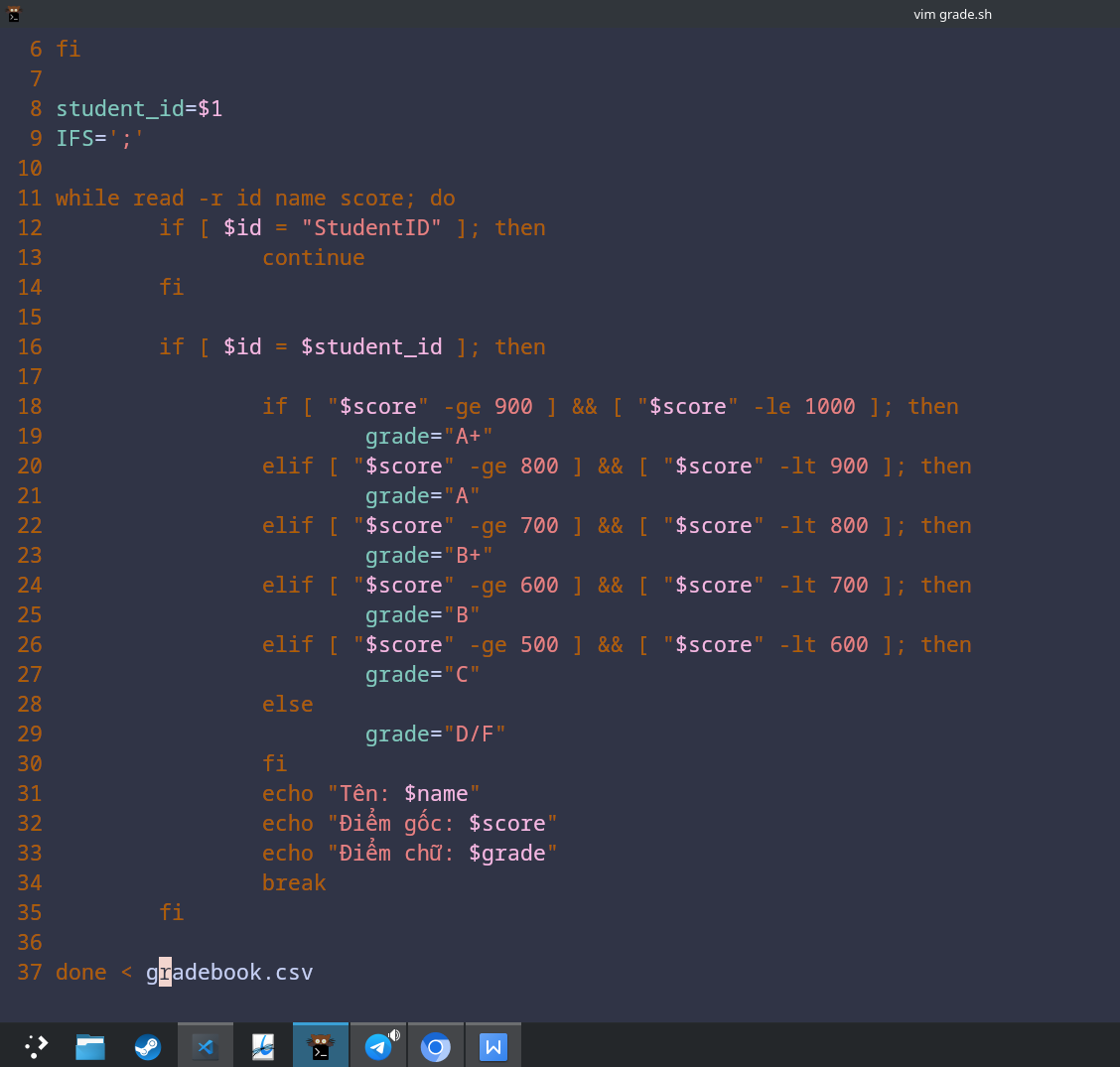
Cách làm: dùng command tail, wc và pipe.

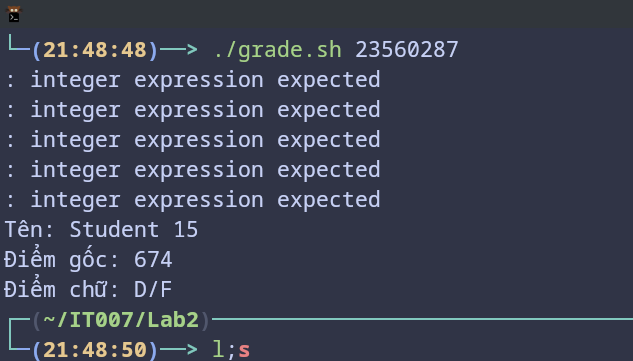


Giải thích: dùng command tail -9 để lấy 9 dòng cuối cùng của file report\_23521421.txt. Sau đó pipe qua command wc -c để tính số từ có trong 9 dòng đó, kết quả là được 505 từ.

# Homework 05

Cách làm: chọn IFS là “;” đọc từng dòng của file, khi nào student\_id = id thì in ra kết quả và break.

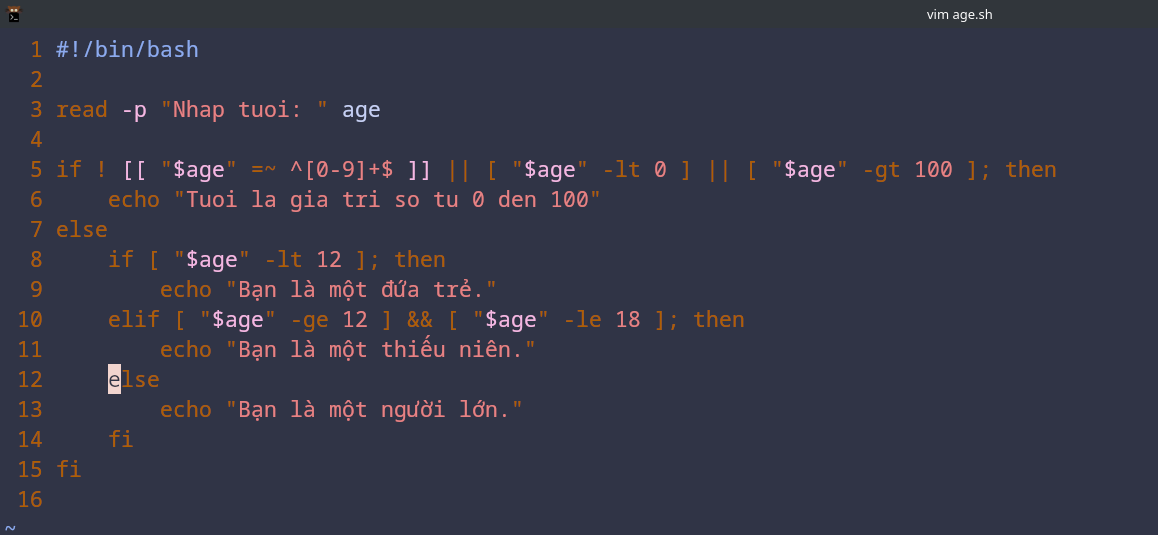


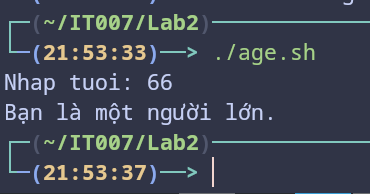


Giải thích: sử dụng lệnh read để đọc 1 dòng 1 lần vào 3 tham số là id, name và score. Để đọc file này thì ta file redirect input từ file vào vòng lặp while, với mỗi dòng thì ta so ID, nếu ID đúng thì ta sẽ in thông tin sinh viên ra và break.

# Homework 06

Cách làm: dùng regex để match tuổi và dùng if else để xác định độ tuổi





Giải thích: regex được dùng trong bài có nghĩa là:

^ là bắt đầu của string

^[0-9] tức là tuổi phải bắt đầu bằng 1 số

^[0-9]+$ tức là tuổi có ít nhất 1 số và string kết thúc chỉ có tuổi

2 mệnh đề ở phía sau đảm bảo tuổi phải không nhỏ hơn 0 và không lớn hơn 100.

# Homework 07

Cách làm: sử dụng command line argument để lấy search string và directory. Sau đó lặp qua tất cả các file trong directory đó, tìm string đó bằng command grep

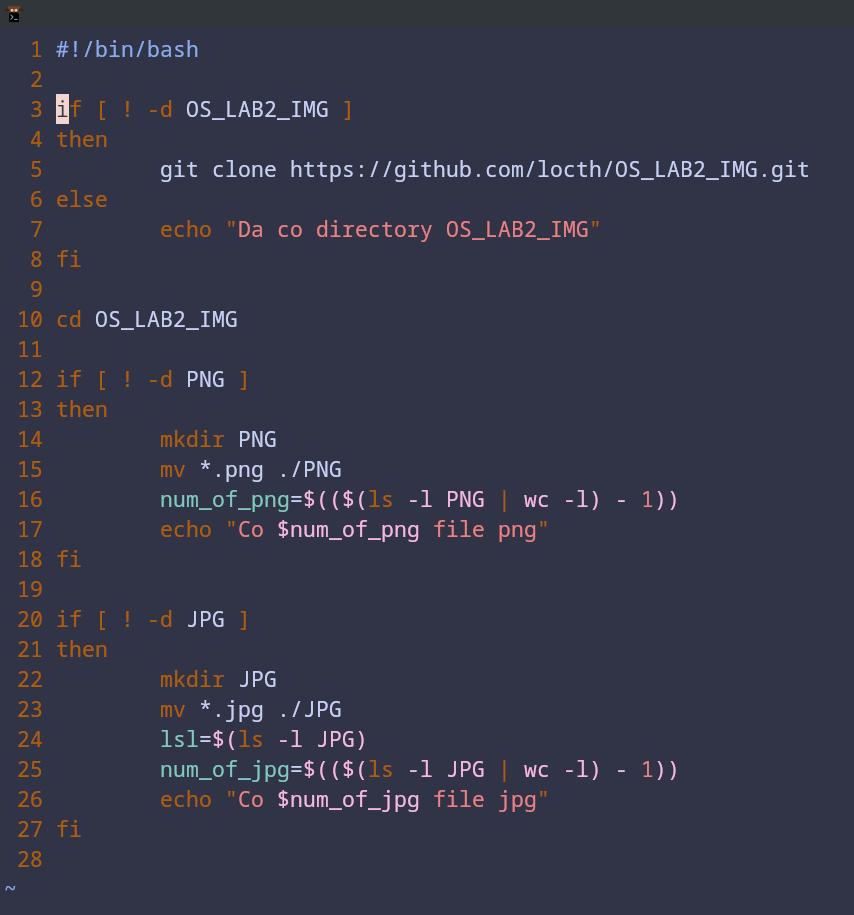


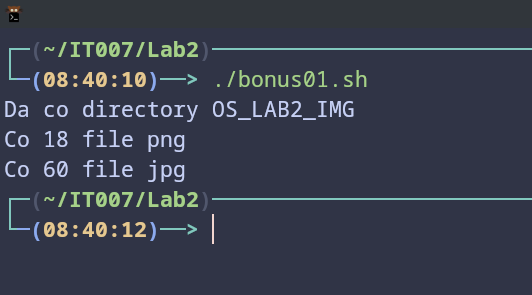
Giải thích: các -ne -d -f -n trong if lần lượt có ý nghĩa là not equal, check là một directory, check là 1 file, và check là 1 string không rỗng.

1. **BONUS**

# Bonus 01

Cách làm: dùng git clone {url} để tải từ github, sau đó dùng hàm ls -l và lấy số hàng trong output(là số file + 1). Rồi in ra kết quả.



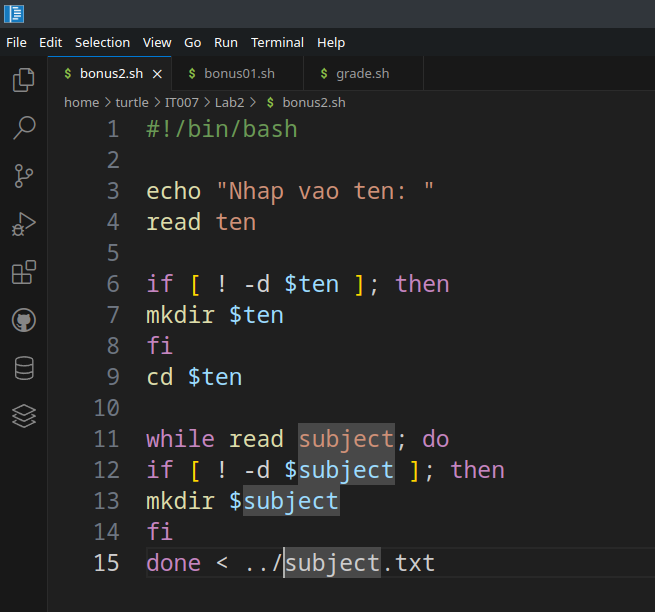


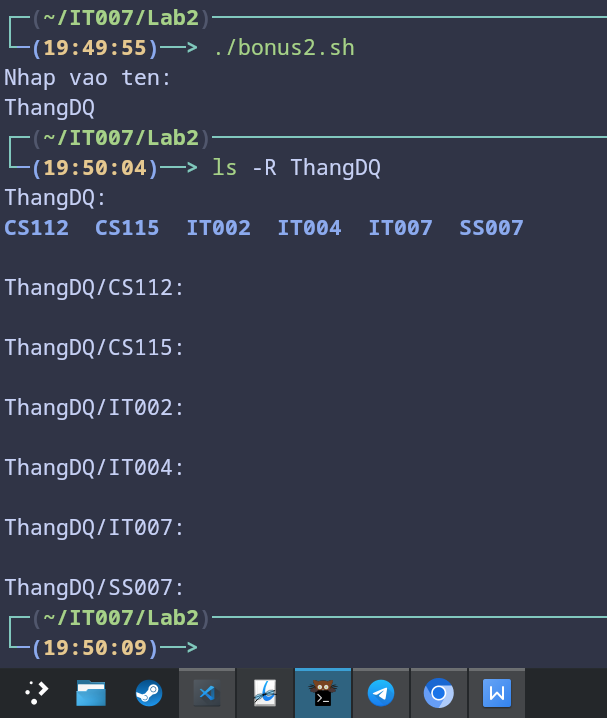
Giải thích: trước đó em đã clone directory về rồi nên đã có dòng if detect là đã có local repo => ko clone nữa. Sau đó lần lượt kiểm tra directory có tồn tại hay không bằng if [ -d directory ] và dùng \*.png, \*.jpg để match tất cả các file có đuôi là png, jpg. Câu lệnh ls sẽ ra 1 hàng metadata nữa nên phải trừ bớt đi 1 để ra đúng số file.

# Bonus 02

Cách làm: dùng lệnh while read var; do {code} done < file.txt

để đọc lần lượt các môn học có trong file subject.txt. Sau đó lần lượt kiểm tra nếu directory đó chưa tồn tại thì ta sẽ tạo với command mkdir.





Giải thích: trong file subject.txt có 6 môn mà em học kì này: CS112, CS115, IT002, IT004, IT007 và SS007. Sau khi chạy script ở trên(với tên e nhập là ThangDQ) thì em dùng lệnh ls -R ThangDQ để hiển thị toàn bộ cây directory của ThangDQ, và nó hiện ra 6 subdir đã được tạo.