|  |
| --- |
| Họ và tên: Nguyễn Duy Khang  Mã số sinh viên:22520619  Lớp: IT007.O11.2 |

HỆ ĐIỀU HÀNH  
BÁO CÁO LAB 1

**CHECKLIST**

**2.5. BÀI TẬP THỰC HÀNH**

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

**2.6. BÀI TẬP ÔN TẬP**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **BT 1.a** | **BT 1.b** | **BT 1.c** | **BT 2.a** | **BT 2.b** |
| **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>\_LAB2.pdf***

**2.5. BÀI TẬP THỰC HÀNH**

# 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.

# 2.4.1.1 Điều khiển shell từ dòng lệnh

A screenshot of a computer

Description automatically generated

* 2.4.1.2 Điều khiển shell bằng tập tin kịch bản (script file)

A screenshot of a computer program

Description automatically generated

* 2.4.1.3 Thực thi script

Cách 1:

A screenshot of a computer

Description automatically generated

Cách 2:

A screenshot of a computer

Description automatically generated

* 2.4.2.1 Sử dụng biến

A screenshot of a computer program

Description automatically generated

2.4.2.2 Các ký tự đặc biệt (metacharacters của shell)

a. Chuyển hướng vào/ra

A computer screen shot of a computer code

Description automatically generated

- Khi thực thi dòng lệnh date > login.time sẽ xuất hiện một file ghi kết quả của date

Tạo file1 chứa Họ tên và MSSV và thực thi ví dụ

A screenshot of a computer program

Description automatically generated

b. Các ký tự đặc biệt kiểm soát tiến trình

- & (Ampersand):

A screenshot of a computer program

Description automatically generated

* Ngoặc đơn ( ; )

A computer screen shot of a computer program

Description automatically generated

* Dấu nháy ` ` (backquotes)

A computer screen shot of a program

Description automatically generated

* Ống dẫn (Pipelines)
* A screenshot of a computer

  Description automatically generated

c. Dấu bọc chuỗi (quoting)

- Backslash ( \ )

A computer program with text

Description automatically generated with medium confidence- Các ví dụ khác về “ ” hay ‘ ’:

Ví dụ 2-1: variables.sh

A screenshot of a computer program

Description automatically generated

2.4.2.3 Biến môi trường (environment variable)

A screenshot of a computer

Description automatically generated

2.4.2.4 Biến tham số (parameter variable)

A screenshot of a computer program

Description automatically generated

Ví dụ 2-2: try\_variables.sh

A screenshot of a computer program

Description automatically generated

2.4.3 Cấu trúc điều kiện

2.4.3.1 Lệnh test hoặc [ ]

A screenshot of a computer program

Description automatically generated

2.4.3.2 Các kiểu điều kiện kiểm tra

A screenshot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated2.4.4 Cấu trúc điều khiển

2.4.4.1 Lệnh if

Ví dụ 2-3: if\_control.sh

A screenshot of a computer program

Description automatically generated

2.4.4.2 Lệnh elif

Ví dụ 2-4: elif\_control.sh

A screenshot of a computer program

Description automatically generated

2.4.4.3 Vấn đề phát sinh với các biến

Trước và sau khi dùng dấu “” bao quanh $timeofday

A screenshot of a computer program

Description automatically generated

# 2.4.4.4 Lệnh for

Ví dụ 2-6: for\_loop.sh

A screenshot of a computer

Description automatically generated

Ví dụ 2-7: for\_loop2.sh

A screen shot of a computer

Description automatically generated

2.4.4.5 Lệnh while

A screenshot of a computer

Description automatically generated

Ví dụ 2-8: password.sh

A screenshot of a computer program

Description automatically generated

Ví dụ 2-9: while\_for.sh

A screenshot of a computer program

Description automatically generated

2.4.4.6 Lệnh until

Ví dụ 2-10: until\_user.sh

A screen shot of a computer

Description automatically generated

2.4.4.7 Lệnh case

Ví dụ 2-11: case1.sh

A screen shot of a computer program

Description automatically generated

Ví dụ 2-12: case2.sh

A screenshot of a computer program

Description automatically generated

Ví dụ 2-13 case3.sh

A screen shot of a computer

Description automatically generated

# 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.

Đoạn code:

#!/bin/sh

my\_mssv="22520619"

echo "Nhap ten"

read name

echo "Nhap mssv:"

read mssv

until [ "$my\_mssv" == "$mssv" ]

do

echo "MSSV vua nhap khong trung voi MSSV cua ban, xin moi nhap lai"

echo "Nhap ten"

read name

echo "Nhap mssv:"

read mssv

done

echo "Chào mừng $name, $my\_mssv đã đăng nhập"

* Kết quả:
* A screenshot of a computer

  Description automatically generated

**Giải thích cách làm:**

* Tạo file bt2.sh để thực hiện đoạn code
* Dòng 2: khởi tạo mssv cụ thể
* Dòng 3 -> 6: yêu cầu người dùng nhập vào họ tên và mssv
* Sử dụng vòng lặp với until, vòng lặp sẽ lặp cho đến khi

[ "$my\_mssv" == "$mssv"] trả về true, nghĩa là khi người dùng nhập đúng mssv thì dừng việc nhập

* Sau đó, vòng lặp sẽ kiểm tra liệu mssv được nhập ở dòng 6 đã nhập đúng hay chưa, nếu chưa sẽ yêu cầu nhập lại ở dòng 10 -> 14 , nếu ở dòng 6 đã nhập đúng thì sẽ tự động kết thúc vòng lặp
* Nếu dòng 10->14, người dùng vẫn tiếp tục nhập sai mssv thì sẽ yêu cầu nhập đến khi đúng thì mới ngừng vòng lặp
* Dòng 16: chào mừng người dùng sau khi hoàn thành nhập

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

* Đoạn code được viết như sau:

#!/bin/sh

echo "Nhap so n: "

read n

while [ "$n" -lt "10" ]

do

    echo "Số vừa nhập nhỏ hơn 10, nhập lại!"

    echo "Nhap so n: "

    read n

done

sum=0

for i in $(seq 1 $n); do

  sum=$((sum + i))

done

 echo "Tổng các số từ 1 đến $n là: $sum"

# - Kết quả: A screenshot of a computer Description automatically generated

Giải thích:

* Dòng 2->3: yêu cầu người dùng nhập số n
* Dòng 5->10: thực hiện vòng lặp while
* Dòng 5: vòng lặp while sẽ thực hiện khi số n này vẫn nhỏ hơn 10, điều kiện ở đây là [ “$n” -lt “10” ], nếu điều kiện vẫn đúng, nghĩa là số n được nhập vào vẫn nhỏ hơn 10 thì sẽ yêu cầu nhập lại cho đến khi số n này lớn hơn hoặc bằng 10
* Dòng 12 -> 14: sử dụng vòng lặp for, thực hiện lặp với index chạy từ 1 tới n, sau đó cộng các index vào biến sum vừa khởi tạo ở dòng 11, ta sẽ thu được sum là tổng các số từ 1 đến n
* Dòng 16: xuất kết quả vừa tìm được

# 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?

Trả lời...

* Đầu tiên tạo 1 file text.txt để chứa chuỗi cụ thể

A screenshot of a computer

Description automatically generated

* Tạo file bt4.sh để thực hiện đoạn code
* #!/bin/sh
* echo "Nhập chuỗi cần kiểm tra:"
* read input\_string
* file\_name="text.txt"
* if [ -f "$file\_name" ]; then
* if grep -q "$input\_string" "$file\_name"; then
* echo "Chuỗi '$input\_string' tồn tại trong file $file\_name."
* else
* echo "Chuỗi '$input\_string' không tồn tại trong file $file\_name."
* fi
* else
* echo "Tệp $file\_name không tồn tại trong thư mục hiện tại."
* Fi
* Chạy chương trình:

A screenshot of a computer program

Description automatically generated

Giải thích:

- Yêu cầu người dùng nhập chuỗi cần kiểm tra và được lưu vào biến “input\_string”.

- Tên của tệp văn bản cần kiểm tra là “text.txt” và được lưu trong biến “file\_name”.

- Sử dụng câu lệnh điều kiện “if [ -f "$file\_name" ]” để kiểm tra xem tệp “test.txt” tồn tại trong thư mục hiện tại hay không.

- Nếu tệp tồn tại, sử dụng lệnh “grep -q "$input\_string" "$file\_name"” để kiểm tra

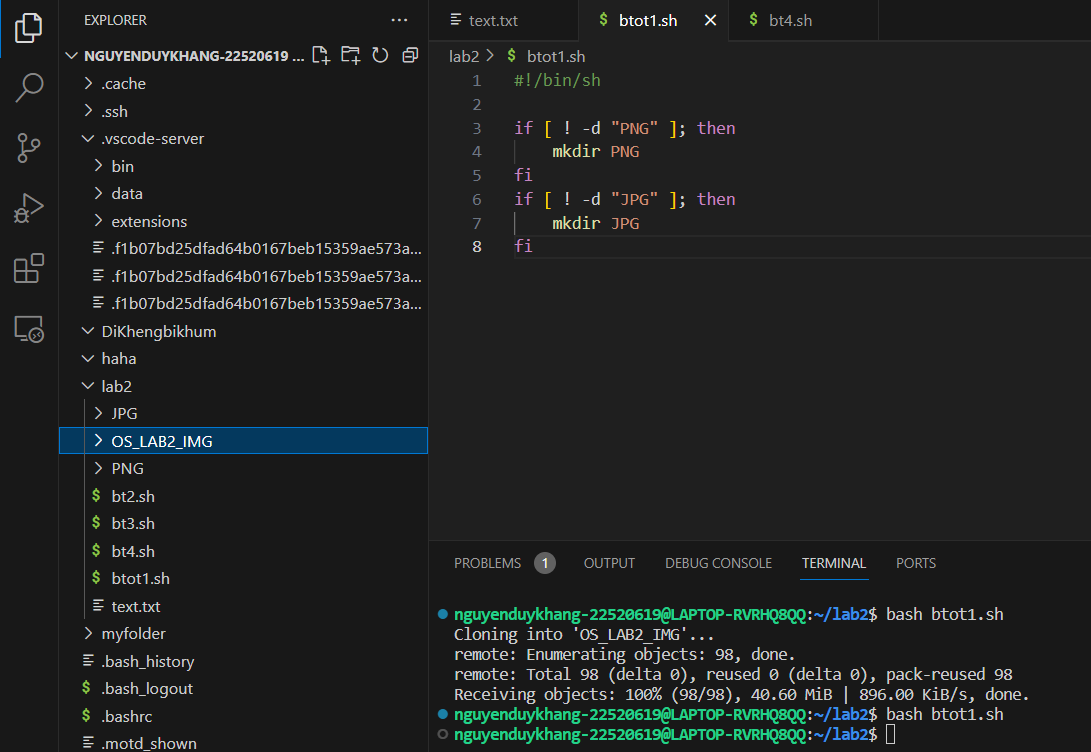
Giải thích: Nếu “grep” tìm thấy chuỗi, nó sẽ trả về 0 và in ra thông điệp “Chuỗi '$input\_string' tồn tại trong tệp $file\_name.” Nếu không tìm thấy, nó sẽ trả về 1 và in ra thông điệp “Chuỗi '$input\_string' không tồn tại trong tệp $file\_name.”

**2.6. BÀI TẬP ÔN TẬP**

# Tìm hiểu trên Google về việc cài đặt lệnh git, sử dụng git để tải thư mục ảnh tại đây: [*https://github.com/locth/OS\_LAB2\_IMG.git*](https://github.com/locth/OS_LAB2_IMG.git)

# A screenshot of a computer Description automatically generated

# Để tải thư mục từ git, dùng lệch “git clone https://github.com/locth/OS\_LAB2\_IMG.git”. Lệnh này sẽ tải thư mục từ đường dẫn về máy. Viết một file kịch bản để làm những công việc sau: a. Kiểm tra trong thư mục người dùng, nếu thấy thư mục PNG và JPG chưa tồn tại thì tạo 02 thư mục này.

* Sử dụng câu lệnh điều kiện if để kiểm tra tồn tại thư mục PNG và JPG chưa, nếu chưa thì sử dụng lệnh mkdir để tạo thư mục.
* 

# b. Di chuyển tất cả file PNG trong thư mục ảnh ở trên vào thư mục PNG. Xuất ra màn hình số lượng ảnh PNG.

* Đoạn code:
* user\_dir="$HOME/lab2"
* count\_png=0
* for picture in $user\_dir/OS\_LAB2\_IMG/\*.png; do
* mv "$picture" "$user\_dir/PNG/"
* count\_png=$((count\_png+1))
* done
* echo "Số lượng ảnh PNG: $count\_png";
* Kết quả chạy chương trình:

A screenshot of a computer screen

Description automatically generated

# Giải thích

* Sử dụng biến “user\_dir” để chứa đường dẫn đến folder di chuyển ảnh PNG.
* Khởi tạo biến “count\_png” với giá trị ban đầu là 0 để đếm số lượng ảnh PNG. o
* “for file in OS\_LAB2\_IMG/\*.png; do”: vòng lặp `for` được sử dụng để lặp qua tất cả các tệp có đuôi “.png” trong thư mục “OS\_LAB2\_IMG”.

# Di chuyển tệp và đếm:

# “mv "$file" "$user\_dir/PNG/"”: Lệnh “mv” được sử dụng để di chuyển tệp từ vị trí ban đầu (được lưu trong biến “file”) đến thư mục “PNG” trong thư mục người dùng

# “count\_png=$((count\_png + 1))”: Sau mỗi lần di chuyển tệp, biến “count\_png” được tăng lên 1 để đếm số lượng tệp PNG đã di chuyển.

# “echo "Số lượng ảnh PNG: $count\_png"”: In ra số lượng tệp PNG đã di chuyển vào thư mục “PNG” trong thư mục người dùng.

# c. Di chuyển tất cả file JPG trong thư mục ảnh ở trên vào thư mục JPG. Xuất ra màn hình số lượng ảnh JPG.

- Đoạn code:

user\_dir="$HOME/lab2"

count\_jpg=0

for picture in $user\_dir/OS\_LAB2\_IMG/\*.jpg; do

    mv "$picture" "$user\_dir/PNG/"

    count\_jpg=$((count\_jpg+1))

done

echo "Số lượng ảnh JPG: $count\_jpg";

* Kết quả
* A screenshot of a computer program

  Description automatically generated
* Do mình gộp chung câu b và c, nên hiện thị thêm 1 phần kết quả khác của câu b

Giải thích: hoàn toàn tương tự với câu b

# 2. Tạo ra một file text tên monhoc.txt chứa danh sách mã môn học của sinh viên trong học kỳ này, mỗi mã nằm trên một dòng. Viết một file kịch bản thực hiện các việc sau: a. Yêu cầu người dùng nhập vào họ và tên (không dấu), tạo ra thư mục có tên tương ứng với thông tin người dùng vừa nhập

# A screen shot of a computer program Description automatically generated

* Dùng lệnh read -r để ngăn chặn các ký tự đặc biệt như / trong phần nhập, và -p để chương trình in ra dòng “Nhap ho va ten: “ và đợi người dùng nhập họ tên vào
* Sau đó gán biến user\_directory là họ tên người dùng và dùng lệnh mkdir để tạo thư mục mới ứng với tên người dùng vừa nhập lúc đầu

# b. Đọc file text monhoc.txt ở trên, ở trong thư mục vừa tạo ở câu a, với mỗi môn học, tạo ra một thư mục có tên tương ứng với mã môn đó.

Trả lời  
A screen shot of a computer

Description automatically generated

Kết quả

A screenshot of a computer

Description automatically generated

Giải thích:

* Dùng vòng lặp while đẻ đọc file monhoc.txt
* read -r ma\_mon\_hoc được dùng để đọc dòng và lưu vào biến ma\_mon\_hoc
* IFS để ngăn cách không sử dụng ký tự trắng hoặc dấu tab khi đọc dòng
* Khi nhập tên có dấu cách thì chương trình sẽ tự động thêm dấu \_ ở những khoảng cách
* Nếu thành công, khi kiểm tra thư mục con, chương trình sẽ xuất ra kết quả như ảnh trên