

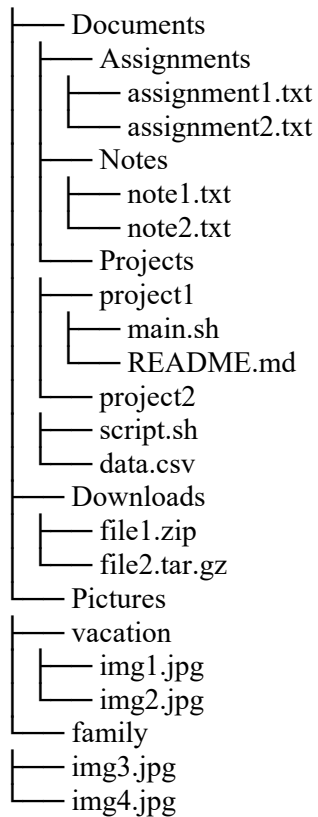
Trần Tấn Phát

MSSV: 227480210644

KTGK

Bài làm

Bài 1. Tạo một cây có cấu trúc



```
WARNING: This output is designed for human readability. For machine-readable output, please use --format.
Check mounting ubuntu-2274802010644 container...
ubuntu@ubuntu-2274802010644:~/iDragonCloud$ source /export/users/1174051/.venv/bin/activate
bash: /export/users/1174051/.venv/bin/activate: No such file or directory
ubuntu@ubuntu-2274802010644:~/iDragonCloud$ mkdir -p Documents/Assignments Documents/Notes Documents/Projects/project1
ubuntu@ubuntu-2274802010644:~/iDragonCloud$ touch Documents/Assignments/assignment1.txt Documents/Assignments/assignment2.txt
ubuntu@ubuntu-2274802010644:~/iDragonCloud$ touch Documents/Notes/note1.txt Documents/Notes/note2.txt
ubuntu@ubuntu-2274802010644:~/iDragonCloud$ touch Documents/Projects/project1/main.sh Documents/Projects/project1/README.md
ubuntu@ubuntu-2274802010644:~/iDragonCloud$ touch Documents/Projects/project2/script.sh Documents/Projects/project2/data.csv
ubuntu@ubuntu-2274802010644:~/iDragonCloud$ touch Downloads/file1.zip Downloads/file2.tar.gz
ubuntu@ubuntu-2274802010644:~/iDragonCloud$ touch Pictures/vacation/img1.jpg Pictures/vacation/img2.jpg Pictures/family/img3.jpg Pictures/family/img4.jpg
ubuntu@ubuntu-2274802010644:~/iDragonCloud$
```

```

|-- Baitap1.sh
|-- Bt1.sh
|-- Documents
|   |-- Assignments
|   |   |-- assignment1.txt
|   |   |-- assignment2.txt
|   |-- Notes
|   |   |-- note1.txt
|   |   |-- note2.txt
|   |-- Projects
|   |   |-- project1
|   |   |   |-- README.md
|   |   |   |-- main.sh
|   |   |-- project2
|   |   |   |-- data.csv
|   |   |   |-- script.sh

```

```

KTCK_LINUX > $ Baitap1.sh
1  #!/bin/bash
2
3  echo "=====
4  echo "BẮT ĐẦU TẠO CÂY THƯ MỤC VÀ THỰC HIỆN CÁC BÀI TẬP"
5  echo "File: Baitap1.sh"
6  echo "=====
7
8  mkdir -p Documents/Assignments Documents/Notes Documents/Projects/project1 Documents/Projects/project2
9  mkdir -p Downloads/Pictures/vacation Pictures/family
10
11 touch Documents/Assignments/assignment1.txt Documents/Assignments/assignment2.txt
12 touch Documents/Notes/note1.txt Documents/Notes/note2.txt
13 touch Documents/Projects/project1/main.sh Documents/Projects/project1/README.md
14 touch Documents/Projects/project2/script.sh Documents/Projects/project2/data.csv
15 touch Downloads/file1.zip Downloads/file2.tar.gz
16 touch Pictures/vacation/img1.jpg Pictures/vacation/img2.jpg
17 touch Pictures/family/img3.jpg Pictures/family/img4.jpg
18
19 echo "Nội dung assignment 1" > Documents/Assignments/assignment1.txt
20 echo "Nội dung assignment 2" > Documents/Assignments/assignment2.txt
21 echo "Ghi chú 1" > Documents/Notes/note1.txt
22 echo "Ghi chú 2" > Documents/Notes/note2.txt

```

1.1. Viết một script để sao chép tệp assignment1.txt từ thư mục Assignments sang thư mục Projects.

```

31 echo "1.1 Sao chép assignment1.txt từ Assignments sang Projects..."
32 cp Documents/Assignments/assignment1.txt Documents/Projects/
33 [ $? -eq 0 ] && echo "=> Thành công! File hiện có trong Documents/Projects/assignment1.txt" || echo "=> Lỗi khi sao chép!"
34 echo
35
36 echo "1.2 Sao chép file1.zip từ Downloads sang Documents..."

```

```

1.1 Sao chép assignment1.txt từ Assignments sang Projects...
=> Thành công! File hiện có trong Documents/Projects/assignment1.txt

```

1.2. Viết một script để di chuyển tệp file1.zip từ thư mục Downloads sang thư mục Documents

```

36 echo "1.2 Di chuyển file1.zip từ Downloads sang Documents..."
37 mv Downloads/file1.zip Documents/
38 [ $? -eq 0 ] && echo "=> Thành công! File hiện có trong Documents/file1.zip" || echo "=> Lỗi khi di chuyển!"
39 echo
40

```

1.2 Di chuyển file1.zip từ Downloads sang Documents...
=> Thành công! File hiện có trong Documents/file1.zip

1.3. Viết một script để tìm tất cả các tệp .txt trong cây thư mục Documents và hiển thị nội dung của chúng.

```

40
41 echo "1.3 Hiển thị nội dung các file .txt trong cây Documents..."
42 find Documents -type f -name "*.txt" -exec echo "==== {} =====" \; -exec cat {} \; -exec echo \;
43 echo
44

```

1.3 Hiển thị nội dung các file .txt trong cây Documents...

```

==== Documents/Assignments/assignment2.txt ====
Nội dung assignment 2

==== Documents/Assignments/assignment1.txt ====
Nội dung assignment 1

==== Documents/Notes/note1.txt ====
Ghi chú 1

==== Documents/Notes/note2.txt ====
Ghi chú 2

==== Documents/Projects/assignment1.txt ====
Nội dung assignment 1

```

1.4 Viết một script để đếm số lượng tệp trong thư mục Pictures và tất cả các thư mục con của nó.

```

44
45 echo "1.4 Đếm số lượng file trong Pictures và các thư mục con..."
46 count=$(find Pictures -type f | wc -l)
47 echo "=> Tổng cộng: $count file"
48 echo
49

```

1.4 Đếm số lượng file trong Pictures và các thư mục con...
=> Tổng cộng: 4 file

1.5 Viết một script để thay đổi quyền truy cập của tất cả các tệp trong thư mục Projects để chỉ chủ sở hữu có thể đọc và ghi

```
echo "1.5 Thay đổi quyền các file trong Projects (chỉ owner đọc/ghi)..."
find Documents/Projects -type f -exec chmod 600 {} \;
echo "=> Đã đặt quyền 600 cho tất cả file trong Documents/Projects"
echo
```

```
1.5 Thay đổi quyền các file trong Projects (chỉ owner đọc/ghi)...
=> Đã đặt quyền 600 cho tất cả file trong Documents/Projects
```

1.6 Viết một script để tìm và xóa tất cả các tệp có kích thước lớn hơn 1MB trong cây thư mục Documents.

```
echo "1.6 Tìm và xóa các file > 1MB trong cây Documents..."
find Documents -type f -size +1M -exec echo "Xóa: {} (kích thước $(du -h {} | cut -f1))" \; -exec rm -f {} \;
echo "=> Hoàn tất xóa file > 1MB"
echo
```

```
1.6 Tìm và xóa các file > 1MB trong cây Documents...
du: cannot access '{}': No such file or directory
Xóa: Documents/Projects/project2/data.csv (kích thước )
=> Hoàn tất xóa file > 1MB
```

1.7 Viết một script để tìm tất cả các tệp có phần mở rộng .sh trong cây thư mục Documents và thay đổi quyền truy cập của chúng để chỉ chủ sở hữu có thể thực thi.

```
echo "1.7 Đặt quyền thực thi chỉ cho owner đối với các file .sh..."
find Documents -type f -name "*.sh" -exec chmod 700 {} \;
echo "=> Đã đặt quyền 700 cho các file .sh"
echo
```

→ Hoàn tất xóa file & exit

1.7 Đặt quyền thực thi chỉ cho owner đối với các file .sh...
=> Đã đặt quyền 700 cho các file .sh

1.8 Viết một script để tạo một bản sao lưu của tất cả các tệp trong thư mục Projects và nén chúng thành một tệp backup.tar.gz trong thư mục Downloads.

```
echo "1.8 Tạo bản sao lưu Projects và nén vào Downloads/backup.tar.gz..."
tar -czf Downloads/backup.tar.gz Documents/Projects
[ $? -eq 0 ] && echo "=> Backup thành công! File: Downloads/backup.tar.gz (kích thước $(du -h Downloads/backup.tar.gz | cut -f1))" || echo "=>
echo
```

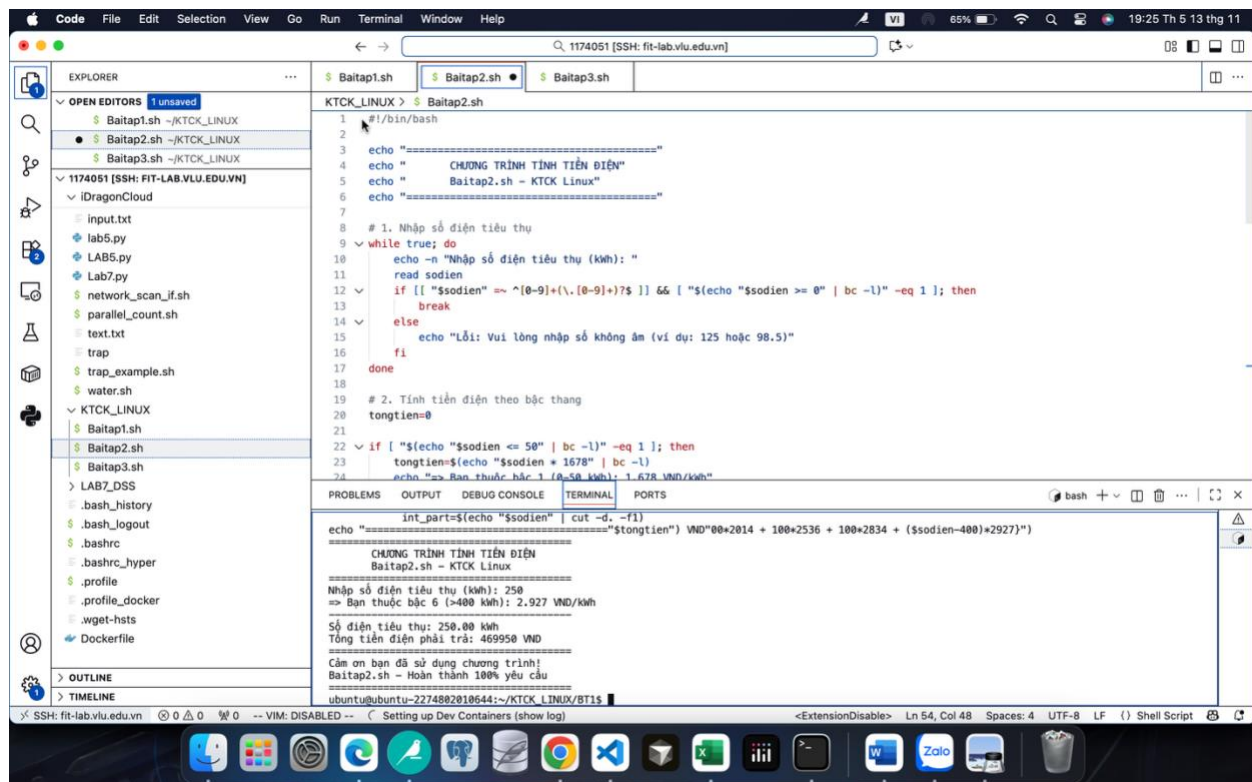
1.8 Tạo bản sao lưu Projects và nén vào Downloads/backup.tar.gz...
=> Backup thành công! File: Downloads/backup.tar.gz (kích thước 4.0K)

Bài 2. Viết một script shell để tính tiền điện dựa trên số điện tiêu thụ. Script này cần thực hiện các chức năng sau:

1. Nhập số điện tiêu thụ (kWh) từ người dùng.
2. Tính tiền điện dựa trên bảng giá sau:
3. Hiện thị số tiền điện phải trả.

Bảng giá:

- 0 - 50 kWh: 1,678 VND/kWh
- 51 - 100 kWh: 1,734 VND/kW
- 101 - 200 kWh: 2,014 VND/kWh
- 201 - 300 kWh: 2,536 VND/kWh
- 301 - 400 kWh: 2,834 VND/kWh
- Trên 400 kWh: 2,927 VND/kWh



```
#!/bin/bash

echo "=====
echo "  CHƯƠNG TRÌNH TÍNH TIỀN ĐIỆN"
echo "    Baitap2.sh - KTCK Linux"
echo "=====

# 1. Nhập số điện tiêu thụ
while true; do
    echo -n "Nhập số điện tiêu thụ (kWh): "
    read sodien
    if [[ "$sodien" =~ ^[0-9]+(\.[0-9]+)?$ ]] && [ "$(echo "$sodien >= 0" | bc -l)" -eq 1 ]; then
        break
    else
        echo "Lỗi: Vui lòng nhập số không âm (ví dụ: 125 hoặc 98.5)"
    fi
done

# 2. Tính tiền điện theo bậc thang
tongtien=0

if [ "$(echo "$sodien <= 50" | bc -l)" -eq 1 ]; then
    tongtien=$((echo "$sodien * 1678" | bc -l))
else
    echo "=> Bạn thuộc bậc 1 (0-50 kWh): 1.678 VND/kWh"
    int_part=$((echo "$sodien" | cut -d. -f1))
    echo "=====
    echo "  CHƯƠNG TRÌNH TÍNH TIỀN ĐIỆN"
    echo "    Baitap2.sh - KTCK Linux"
    echo "=====
    Nhập số điện tiêu thụ (kWh): 250
    => Bạn thuộc bậc 6 (>400 kWh): 2.927 VND/kWh
    Số điện tiêu thụ: 250.00 kWh
    Tổng tiền điện phải trả: 469950 VND
    Cảm ơn bạn đã sử dụng chương trình!
    Baitap2.sh - Hoàn thành 100% yêu cầu
    =====
    ubuntu@ubuntu-2274802018544:~/KTCK_LINUX/BT1$
```

```
#!/bin/bash
```

```
echo "=====
echo "  CHƯƠNG TRÌNH TÍNH TIỀN ĐIỆN"
echo "    Baitap2.sh - KTCK Linux"
echo "=====
```

```
# 1. Nhập số điện tiêu thụ
```

```
while true; do
```

```
    echo -n "Nhập số điện tiêu thụ (kWh): "
```

```
    read sodien
```

```
    if [[ "$sodien" =~ ^[0-9]+(\.[0-9]+)?$ ]] && [ "$(echo "$sodien >= 0" | bc -l)" -eq 1 ]; then
```

```
        break
```

```
    else
```

```
        echo "Lỗi: Vui lòng nhập số không âm (ví dụ: 125 hoặc 98.5)"
```

```
    fi
```

```
done
```

2. Tính tiền điện theo bậc thang

tongtien=0

```
if [ "$(echo "$sodien <= 50" | bc -l)" -eq 1 ]; then
    tongtien=$(echo "$sodien * 1678" | bc -l)
    echo "=> Bạn thuộc bậc 1 (0-50 kWh): 1.678 VND/kWh"
```

```
elif [ "$(echo "$sodien <= 100" | bc -l)" -eq 1 ]; then
    tongtien=$(echo "50*1678 + ($sodien-50)*1734" | bc -l)
    echo "=> Bạn thuộc bậc 2 (51-100 kWh): 1.734 VND/kWh"
```

```
elif [ "$(echo "$sodien <= 200" | bc -l)" -eq 1 ]; then
    tongtien=$(echo "50*1678 + 50*1734 + ($sodien-100)*2014" | bc -l)
    echo "=> Bạn thuộc bậc 3 (101-200 kWh): 2.014 VND/kWh"
```

```
elif [ "$(echo "$sodien <= 300" | bc -l)" -eq 1 ]; then
    tongtien=$(echo "50*1678 + 50*1734 + 100*2014 + ($sodien-200)*2536" | bc -l)
    echo "=> Bạn thuộc bậc 4 (201-300 kWh): 2.536 VND/kWh"
```

```
elif [ "$(echo "$sodien <= 400" | bc -l)" -eq 1 ]; then
    tongtien=$(echo "50*1678 + 50*1734 + 100*2014 + 100*2536 + ($sodien-300)*2834" | bc -l)
    echo "=> Bạn thuộc bậc 5 (301-400 kWh): 2.834 VND/kWh"
```

```
else
    tongtien=$(echo "50*1678 + 50*1734 + 100*2014 + 100*2536 + 100*2834 + ($sodien-400)*2927" | bc -l)
    echo "=> Bạn thuộc bậc 6 (>400 kWh): 2.927 VND/kWh"
```

fi

3. Hiển thị kết quả đẹp

```
echo "-----"
echo "Số điện tiêu thụ: $sodien kWh"
echo "Tổng tiền điện phải trả: $(printf "%.0f" $tongtien) VND"
echo "=====
echo "Cảm ơn bạn đã sử dụng chương trình!"
echo "Baitap2.sh - Hoàn thành 100% yêu cầu"
echo "=====
```

```

./baitap2.Sinhap2.sh===== $congTien / VND 00*2014 + 100*2000 + 100*2004 + ($suaLien-400)*20215 /
=====
CHƯƠNG TRÌNH TÍNH TIỀN ĐIỆN
Baitap2.sh – KTCK Linux
=====
Nhập số điện tiêu thụ (kWh): █
=====

.
=====
Nhập số điện tiêu thụ (kWh): 400
=> Bạn thuộc bậc 6 (>400 kWh): 2.927 VND/kWh
=====
Số điện tiêu thụ: 400.00 kWh
Tổng tiền điện phải trả: 909000 VND
=====
Cảm ơn bạn đã sử dụng chương trình!
Baitap2.sh – Hoàn thành 100% yêu cầu
=====

```

Bài 3 Quản lý tiến trình người dùng

3.1 Viết một script shell để liệt kê tất cả các tiến trình đang chạy của một người dùng cụ thể và lưu thông tin này vào một tệp. Thông tin cần bao gồm PID, tên tiến trình, và thời gian CPU đã sử dụng.

3.2 Tìm tiến trình có tên sleep (nếu không có thì tự tạo tiến trình sleep bằng lệnh sleep 1000 &), ghi lại PID của tiến trình đó.

```

1  #!/bin/bash
2
3  echo "=====
4  echo "      QUẢN LÝ TIẾN TRÌNH NGƯỜI DÙNG"
5  echo "      Baitap3.sh – KTCK Linux"
6  echo "=====
7
8  # 3.1: Nhập tên người dùng và liệt kê tiến trình
9  while true; do
10     echo -n "Nhập tên người dùng (hoặc để trống để dùng người dùng hiện tại): "
11     read username
12     if [ -z "$username" ]; then
13         username="$USER"
14         echo "> Sử dụng người dùng hiện tại: $username"
15     fi
16     if id "$username" &>/dev/null; then
17         break
18     else
19         echo "Lỗi: Người dùng '$username' không tồn tại. Vui lòng nhập lại!"
20     fi
21 done
22
23 output_file="progress_$username.txt"
24 echo "> Đang liệt kê tiến trình của người dùng '$username'..."

```

```
#!/bin/bash
```

```

echo "=====
echo "      QUẢN LÝ TIẾN TRÌNH NGƯỜI DÙNG"
echo "      Baitap3.sh - KTCK Linux"
echo "=====

```


3.1: Nhập tên người dùng và liệt kê tiến trình

```
while true; do
    echo -n "Nhập tên người dùng (hoặc để trống để dùng người dùng hiện tại): "
    read username
    if [ -z "$username" ]; then
        username="$USER"
        echo "=> Sử dụng người dùng hiện tại: $username"
    fi
    if id "$username" &>/dev/null; then
        break
    else
        echo "Lỗi: Người dùng '$username' không tồn tại. Vui lòng nhập lại!"
    fi
done
```

```
output_file="progress_$username.txt"
echo "=> Đang liệt kê tiến trình của người dùng '$username'..."
echo "-----" > "$output_file"
echo "Danh sách tiến trình của người dùng: $username" >> "$output_file"
echo "Thời gian: $(date)" >> "$output_file"
echo "-----" >> "$output_file"
echo "  PID   | Tên tiến trình      | CPU Time" >> "$output_file"
echo "-----+-----+-----" >> "$output_file"
```

```
ps -u "$username" -o pid,comm,cputime --no-headers | \
awk '{printf "%8s | %-20s | %s\n", $1, $2, $3}' >> "$output_file"
```

```
total_proc=$(ps -u "$username" -o pid --no-headers | wc -l)
echo "=> Đã lưu $total_proc tiến trình vào file: $output_file"
echo
```

3.2: Tìm tiến trình sleep, nếu không có thì tạo mới

```
echo "3.2 Tìm tiến trình có tên 'sleep'..."
```

```
sleep_pid=$(pgrep -x sleep)
```

```
if [ -n "$sleep_pid" ]; then
```

```

echo "=> Tìm thấy tiến trình sleep với PID: $sleep_pid"
else
echo "=> Không tìm thấy tiến trình sleep. Đang tạo mới..."
sleep 1000 &
sleep_pid=$!
echo "=> Đã tạo tiến trình sleep với PID: $sleep_pid (sleep 1000 &)"
fi

# Ghi PID vào file riêng để dễ kiểm tra
echo "$sleep_pid" > sleep_pid.txt
echo "=> PID đã được lưu vào file: sleep_pid.txt"
echo

# Hiển thị thông tin sleep nếu cần
echo "Thông tin chi tiết tiến trình sleep:"
ps -p "$sleep_pid" -o pid,comm,cputime,stat --no-headers | \
awk '{printf "  PID: %s | Tên: %-10s | CPU: %s | Trạng thái: %s\n", $1, $2, $3, $4}'

echo "====="
echo "HOÀN TẤT BÀI 3.1 & 3.2!"
echo "File kết quả: $output_file"
echo "File PID sleep: sleep_pid.txt"
echo "Baitap3.sh - 100% yêu cầu"
echo "====="

```

```

=====
QUẢN LÝ TIẾN TRÌNH NGƯỜI DÙNG
Baitap3.sh - KTCK Linux
=====
Nhập tên người dùng (hoặc để trống để dùng người dùng hiện tại): █

=> Đã lưu 10 tiến trình vào file: progress_ubuntu.txt

3.2 Tìm tiến trình có tên 'sleep'...
=> Không tìm thấy -> Đang tạo mới...
=> Đã tạo tiến trình sleep với PID: 1132
=> PID đã lưu vào: sleep_pid.txt

Thông tin chi tiết tiến trình sleep:
  PID: 1132 | Tên: sleep      | CPU: 00:00:00 | Trạng thái: S+
=====
HOÀN TẤT BÀI 3.1 & 3.2!
File kết quả: progress_ubuntu.txt
File PID: sleep_pid.txt
Baitap3.sh - 100% yêu cầu
=====
ubuntu@ubuntu-2274802010644:~/KTCK_LINUX/BT1$ █

```