Error 520 原因解析

linux中case语句的使用示例

在 Linux 中，`case` 语句用于根据不同的条件执行不同的命令，它可以被用来替代多个 `if-else` 语句，尤其适用于处理多种可能的情况。`case` 语句基于模式匹配来执行不同的代码块。

语法：

```bash
case "$variable" in
    pattern1)
        # Command(s) to execute if $variable matches pattern1
        ;;
    pattern2)
        # Command(s) to execute if $variable matches pattern2
        ;;
    *)
        # Command(s) to execute if no patterns match
        ;;
esac
```

示例 1：简单的 `case` 语句

```bash
#!/bin/bash

echo "Enter a number between 1 and 3:"
read num

case "$num" in
    1)
        echo "You selected one."
        ;;
    2)
        echo "You selected two."
        ;;
    3)
        echo "You selected three."
        ;;
    *)
        echo "Invalid selection."
        ;;
esac
```

- 这个脚本会提示用户输入 1、2 或 3，然后根据用户的输入输出相应的消息。如果输入的是 1、2 或 3，将分别输出对应的消息。如果输入的是其他值，则输出 `"Invalid selection."`。

示例 2：处理文件类型

```
bash


  #!/bin/bash

  echo "Enter a filename:"
  read filename

  case "$filename" in
      *.txt)
          echo "This is a text file."
          ;;
      *.jpg|*.jpeg)
          echo "This is a JPEG image."
          ;;
      *.png)
          echo "This is a PNG image."
          ;;
      *)
          echo "Unknown file type."
          ;;
  esac
```

- 该脚本根据用户输入的文件名的扩展名输出文件类型。例如，如果用户输入一个以 `.txt` 结尾的文件名，脚本会输出 `"This is a text file."`。如果是 `.jpg` 或 `.jpeg` 文件，脚本会输出 `"This is a JPEG image."`。

示例 3：使用 `case` 判断命令行参数

```
bash


  #!/bin/bash

  echo "Enter a command (start, stop, restart):"
  read command

  case "$command" in
      start)
          echo "Starting the service..."
          # Start the service (假设的命令)
          ;;
      stop)
          echo "Stopping the service..."
          # Stop the service (假设的命令)
          ;;
      restart)
          echo "Restarting the service..."
          # Restart the service (假设的命令)
          ;;
      *)
          echo "Invalid command."
          ;;
  esac
```

- 该脚本提示用户输入 `start`、`stop` 或 `restart` 之一，根据输入执行不同的操作。如果输入的是其他值，脚本会输出 `"Invalid command."`。

示例 4：多个模式匹配

```bash
#!/bin/bash

echo "Enter a character (a, b, or c):"
read char

case "$char" in
    [aA])
        echo "You entered 'a' or 'A'."
        ;;
    [bB])
        echo "You entered 'b' or 'B'."
        ;;
    [cC])
        echo "You entered 'c' or 'C'."
        ;;
    *)
        echo "You entered something else."
        ;;
esac
```

- 这个例子展示了如何使用字符范围 `[aA]` 来匹配字母 `a` 或 `A`，同理 `[bB]` 和 `[cC]` 用来匹配字母 `b` 或 `B` 和 `c` 或 `C`。

示例 5：`case` 中的逻辑操作

```bash
#!/bin/bash

echo "Enter a number:"
read num

case "$num" in
    [0-9])   # 匹配 0 到 9 的数字
        echo "You entered a single digit."
        ;;
    [0-9][0-9])  # 匹配 2 位数字
        echo "You entered a two-digit number."
        ;;
    *)
        echo "Invalid input."
        ;;
esac
```

- 这里 `case` 语句用于检查输入的数字是单一的数字，还是两位数的数字。

总结：

- `case` 语句通常用于处理多个选项，尤其是在需要处理多个条件时，它比 `if-else` 语句更简洁。
- 可以使用正则模式来匹配不同的输入模式。
- `*` 表示默认情况，即当没有匹配到任何模式时执行的代码块。

`case` 语句结构清晰，常用于脚本中来处理多种不同的输入。