

第五集： Bash 中的管道与重定向

课程目标

本课程将带你深入了解 Bash 中的管道与重定向操作。通过学习本集内容，你将学会如何使用管道符号 (|) 将命令的输出作为下一个命令的输入，以及如何利用重定向符号 (> , >> , <) 来处理输入输出。我们还将结合常用命令如 `grep` 和 `sort` 进行实际操作。

1. 管道与重定向简介

- **管道 (|)**：将一个命令的输出直接作为下一个命令的输入。
- **重定向 (> , >> , <)**：
 - `>`：将命令的输出覆盖写入到文件。
 - `>>`：将命令的输出追加写入到文件。
 - `<`：将文件内容作为命令的输入。

2. 管道操作

示例：使用 `ls` 列出文件并通过 `grep` 过滤特定文件

```
ls -l | grep "filename"
```

步骤：

1. 在终端中输入 `ls -l | grep "filename"`。
2. `ls -l` 的输出会通过管道传递给 `grep`，`grep` 会从中筛选出包含 "filename" 的行。

3. 重定向操作

示例 1：将命令的输出写入文件

```
echo "Hello, World!" > output.txt
```

- 这会将 `echo` 的输出写入到 `output.txt` 文件中（如果文件已存在，会被覆盖）。

示例 2：将命令的输出追加到文件

```
echo "Hello again!" >> output.txt
```

- 这会将输出追加到 `output.txt` 文件的末尾，而不是覆盖内容。

示例 3：将文件内容作为命令输入

```
sort < unsorted.txt
```

- 这会对 `unsorted.txt` 文件中的内容进行排序并在终端显示。

4. 实操任务

任务 1：使用管道与 `grep`，筛选出当前目录下以 `.sh` 结尾的文件。

```
ls -l | grep ".sh"
```

任务 2：将一段文本写入文件，然后将另一段文本追加到同一个文件中。

```
echo "First line of text" > myfile.txt  
echo "Second line of text" >> myfile.txt
```

任务 3：从一个文件中读取内容，并对其进行排序。

```
sort < myfile.txt
```

小结

在本集课程中，我们学习了如何使用 Bash 中的管道与重定向功能，通过一系列实际操作了解了如何将多个命令的功能组合起来。管道和重定向是 Bash 中极为强大的工具，可以让我们高效地处理数据。

下节预告：在下一集课程中，我们将探讨 Bash 中的环境变量及其在自动化任务中的重要性。