PowerShell's pipeline allows for sequential command execution, where the output of one command serves as the input for the next. Filtering and operators are crucial for manipulating data within the pipeline.

Filtering with Where-Object

The Where-Object cmdlet filters objects based on specified criteria. It uses a script block to evaluate each object, passing only those that meet the condition.

```
Get-Process | Where-Object {$_.CPU -gt 1}
```

This command retrieves all processes and filters them, displaying only those with CPU usage greater than 1 second.

Comparison Operators

PowerShell offers various comparison operators for use in filtering:

- -eq: equals
- -ne: not equals
- -gt: greater than
- -ge: greater than or equals
- -lt: less than
- -le: less than or equals
- -like: wildcard matching
- -notlike: not wildcard matching
- -match: regular expression matching
- -notmatch: not regular expression matching

Logical Operators

Multiple conditions can be combined using logical operators:

- -and: Both conditions must be true.
- -or: At least one condition must be true.
- -not or !: Negates the condition.

```
Get-Process | Where-Object {$_.CPU -gt 1 -and $_.WorkingSet -gt
10MB}
```

This command retrieves processes with CPU usage greater than 1 second and working set greater than 10MB.

Pipeline Chain Operators (PowerShell 7+)

PowerShell 7 introduced pipeline chain operators for conditional execution: [1]

- &&: Executes the right-hand pipeline only if the left-hand pipeline succeeds.
- ||: Executes the right-hand pipeline only if the left-hand pipeline fails.

```
Get-Process non_existent_process && Write-Host "Process found" ||
Write-Host "Process not found"
```

Background Operator

The & operator runs a pipeline in the background as a PowerShell job.

```
Get-Process | & { Start-Sleep -Seconds 5; Write-Host "Processes
listed after 5 seconds" }
```

Order of Operations

Filtering early in the pipeline is more efficient. It reduces the amount of data processed by subsequent commands.

```
Get-ChildItem | Where-Object {$_.LastWriteTime -gt
(Get-Date).AddDays(-7)} | Select-Object Name, Length
```

This command first filters files modified in the last 7 days and then selects the name and length properties.