Linux Shell Expansions: Braces, Brackets, Parentheses, and Extended Expansions

Your Name

August 18, 2024

1. Brace Expansion {}

Syntax: {item1,item2,...}
Use Case: Creates multiple text strings based on a pattern. Useful for generating sequences or lists.
Example:

```
mkdir folder{1,2,3}
```

Expands to: `mkdir folder1 folder2 folder3`

2. Brackets []

Syntax: `[pattern]`
Use Case: Matches any single character within the brackets. Used in filename matching (globbing).
Example:

```
ls file[abc].txt
```

Matches: `filea.txt fileb.txt filec.txt`

3. Parentheses ()

Syntax: `(command1; command2; ...)`
Use Case: Groups commands in a subshell. All commands inside parentheses are executed in a new shell process.
Example:

```
(cd dir1; ls)
```

Changes directory to `dir1` and lists its contents, without changing the current shell's working directory.

4. Double Parentheses `((...))`

Syntax: `((expression))`
Use Case: Used for arithmetic evaluations. Performs arithmetic operations and comparisons.
Example:

```
a=5
b=10
((sum = a + b))
echo $sum
```

Outputs: `15`

5. Double Brackets `[[...]]`

Syntax: `[[condition]]`
Use Case: Provides advanced conditional testing. Supports string and pattern matching with improved syntax compared to single brackets.
Example:

```
if [[ $var == a* ]]; then echo "Starts with a"; fi
```

Matches: Any string starting with `a`.

6. Extended Brace Expansion {{}}

Syntax: {{`pattern1,pattern2,...`}}
Use Case: Similar to brace expansion but allows for nested and more complex expansions.
Example:

```
echo {a,b{c,d},e}
```

Expands to: `a bc bd e`

7. Extended Globbing `[[...]]`

Syntax: `[[pattern]]`
Use Case: Used in conditional expressions in shell scripts. Provides advanced pattern matching.
Example:

```
if [[ $var == a* ]]; then echo "Starts with a"; fi
```

Matches: Any string starting with `a`.

8. Command Substitution `$()` and backticks ``

Syntax: `$(command)` or `` `command` ``
Use Case: Captures the output of a command and replaces the command with its output.

Example:

```
echo "Today is $(date)"
```

Outputs: `Today is Mon Aug 18 12:34:56 UTC 2024`

Conclusion

This document provides a quick reference for different types of expansions in the Linux shell. These tools are essential for scripting and command-line efficiency.