【Linux Programming】Day5

Tags	
Date	@May 19, 2022
Summary	Conditions: test and [

【Ch2】Shell Programming

2.4.5 Conditions

A shell script can test the exit code of any command that can be invoked from the command line, including the scripts that we write.

That's why it's important to always include an `exit` command with a value at the end of any scripts that we write.

The test or [Command

Most scripts make extensive use of the `[` or test command, the shell's Boolean check.

Note: Because the test command is infrequently used outside shell scripts, many Linux users call their scripts `test`. If such a program doesn't work, it's probably conflicting with the shell's test command.

To find out whether our system has an external command of a given name, try type `which test`, to check which test command is being executed, or use `./test` to ensure that we execute the script in the current directory.

We'll introduce the test command using one of the simplest conditions: checking to see whether a file exists. The command for this is test -f <filename>, so within a script we can write:

```
if test -f fred.c
then
```

```
...
fi
```

We can also write it like this

```
if [ -f fred.c ]
then
...
fi
```

The `test` command's exit code determines whether the conditional code is run.

Note that we must put spaces between the `[` braces and the condition being checked.

The condition types that we can use with the test command fall into three types: string comparison, arithmetic comparison, and file conditionals. The following table describes these condition types:

String Comparison	Result
`string1 = string2`	True if the strings are equal
`string1 != string2`	True if the strings are not equal
`-n string`	True if the string is not null
`-z string`	True if the string is null (an empty string)
Arithmetic Comparison	**Result**
`expression1 -eq expression2`	True if the expressions are equal
`expression1 -ne expression2`	True if the expressions are not equal
`expression1 -gt expression2`	True if `expression1` is greater than `expression2`
`expression1 -ge expression2`	True if `expression1` is greater than or equal to `expression2`
`expression1 -lt expression2`	True if `expression1` is less than `expression2`
`expression1 -le expression2`	True if `expression1` is less than or equal to `expression2`
`! expression`	True if the expression is false, and vice versa
File Conditional	**Result**
`-d file`	True if the file is a directory
`-e file`	True if the file exists. Note that historically the `-e` option has not been portable, so `-f` is usually used.
`-f file`	True if the file is a regular file
`-g file`	True if `set-group-id` is set on `file`
`-r file`	True if the file is readable
`-s file`	True if the file has nonzero size
`-u file`	True if `set-user-id` is set on `file`
`-w file`	True if the file is writable
`-x file`	True if the file is executable