

# 8. Test

`test` 是一个测试命令，用于测试各种条件。它通常用于脚本中，以根据条件的真假来执行不同的操作。在 Unix 系统中，`test` 命令通常位于 `/usr/bin/` 目录下。它的语法如下：

```
$ type [  
[ is a shell builtin  
$ which [  
/usr/bin/[  
$ ls -l /usr/bin/[  
lrwxrwxrwx 1 root root 4 Mar 27 2000 /usr/bin/[ -> test
```

例如，我们可以使用 `test` 命令来检查一个变量是否等于某个值：

```
if [ $foo = "bar" ]
```

这里，`test` 命令被用来检查变量 `$foo` 是否等于字符串 `"bar"`。如果条件为真，那么后面的命令就会被执行。需要注意的是，在 `test` 命令中，字符串需要用单引号或双引号括起来，以避免空格等问题。

```
if SPACE [ SPACE "$foo" SPACE = SPACE "bar" SPACE ]
```

这里，我们使用 `SPACE` 来代表空格。在 `test` 命令中，空格是一个特殊的字符，需要特殊处理。使用 `"=="` 和 `"-eq"` 来比较数值。

`test` 命令还可以用于测试文件的存在性、权限等。例如，我们可以使用 `test` 命令来检查文件 `man test` 是否存在：

`test` 命令还可以用于控制流。例如，我们可以使用 `if`、`while` 等语句来根据条件的真假来执行不同的操作。例如，我们可以使用 `if...then...else...` 语句来根据条件的真假来执行不同的操作：

```
if [ ... ] then  
    # if-code  
else  
    # else-code  
fi
```

```
fi
do something
done
!do something
done
esac
do something
done
if [ ... ]; then
do something
done
else
do something
done
fi
```

```
if [ ... ]; then
    # do something
fi
```

```
do something elif do something done
```

```
if [ something ]; then
    echo "Something"
elif [ something_else ]; then
    echo "Something else"
else
    echo "None of the above"
fi
```

```
[something] do something "Something" done , do something [ something_else ]
done
[something_else] do something "Something else" done . do something done
"None of the above" done
```

```
do something done do X do something done (-1, 0, 1, hello, bye do something
done ). do something done (do - 1 do something done do something done Dave
done ):
```

```
$ X=5
$ export X
$ ./test.sh
... output of test.sh ...
$ X=hello
$ ./test.sh
... output of test.sh ...
$ X=test.sh
$ ./test.sh
... output of test.sh ...
```

```
do something $X do something do (do :/etc/hosts) do something done
```

```
#!/bin/sh
if [ "$X" -lt "0" ]
```



```
[ -f $X ] && echo "X is a file" || echo "X is not a file"
[ -n $X ] && echo "X is of non-zero length" || \
    echo "X is of zero length"
```

```

if (test) {
    // if test is true, then do this
} else {
    // if test is false, then do this
}

// if test is true, then do this, otherwise do this
if (test) {
    // if test is true, then do this
} else {
    // if test is false, then do this
}

// if test is true, then do this, otherwise do this
if (test) {
    // if test is true, then do this
} else {
    // if test is false, then do this
}

// if test is true, then do this, otherwise do this
if (test) {
    // if test is true, then do this
} else {
    // if test is false, then do this
}

```

X            :

```
test.sh: [: integer expression expected before -lt
test.sh: [: integer expression expected before -gt
test.sh: [: integer expression expected before -le
test.sh: [: integer expression expected before -ge
```

[illegible]

```
echo -en "Please guess the magic number: "
read X
echo $X | grep "[^0-9]" > /dev/null 2>&1
if [ "$?" -eq "0" ]; then
    # If the grep found something other than 0-9
    # then it's not an integer.
    echo "Sorry, wanted a number"
else
    # The grep found only 0-9, so it's an integer.
    # We can safely do a test on it.
    if [ "$X" = "7" ]; then
        echo "You entered the magic number!"
    fi
fi
```

[illegible]

while test [ -n "\$X" ] :

```
#!/bin/sh
X=0
while [ -n "$X" ]
do
    echo "Enter some text (RETURN to quit)"
    read X
    echo "You said: $X"
done
```

RETURN [ -n "\$X" ] (X 0 ).  
Justin Heath . [ -n "\$X" ] \$X  
 . \$X [ -n "\$X" ]  
 :  
 :

```
$ ./test2.sh
Enter some text (RETURN to quit)
fred
You said: fred
Enter some text (RETURN to quit)
wilma
You said: wilma
Enter some text (RETURN to quit)
```

:  
 :

\$

:  
 :

```
#!/bin/sh
X=0
while [ -n "$X" ]
do
    echo "Enter some text (RETURN to quit)"
    read X
    if [ -n "$X" ]; then
        echo "You said: $X"
    fi
done
```

