

- [Blog](#)
- [Paste](#)
- [Ubuntu](#)
- [Wiki](#)
- [Linux](#)
- [Forum](#)

搜索

進入

搜索

- [頁面](#)
- [討論](#)
- [編輯](#)
- [歷史](#)
- [简体](#)
- [繁體](#)

- [導航](#)
 - [首頁](#)
 - [社群入口](#)
 - [現時事件](#)
 - [最近更改](#)
 - [隨機頁面](#)
 - [幫助](#)
- [工具箱](#)
 - [鏈入頁面](#)
 - [鏈出更改](#)
 - [所有特殊頁面](#)
- [個人工具](#)
 - [登入](#)

跟我一起寫**Makefile**:使用條件判斷

出自**Ubuntu**中文

*導 | 概述 + **MakeFile**介紹 + 書寫規則 + 書寫命令 + 使用變量 + 使
*航 | 用條件判斷 |
* | 使用函數 + **make**運行 + 隱含規則 + 使用**make**更新函數庫文件
* | 後序 |

目錄

- 1 使用條件判斷
 - 1.1 示例
 - 1.2 語法

使用條件判斷

使用條件判斷，可以讓**make**根據運行時的不同情況選擇不同的執行分支。條件表達式可以是比較變量的值，或是比較變量和常量的值。

示例

下面的例子，判斷\$(CC)變量是否“gcc”，如果是的話，則使用GNU函數編譯目標。

```
libs_for_gcc = -lgnu
normal_libs =

foo: $(objects)
ifeq ($(CC),gcc)
    $(CC) -o foo $(objects) $(libs_for_gcc)
else
    $(CC) -o foo $(objects) $(normal_libs)
endif
```

可見，在上面示例的這個規則中，目標“foo”可以根據變量“\$(CC)”值來選取不同的函數庫來編譯程序。

我們可以從上面的示例中看到三個關鍵字：ifeq、else和endif。ifeq的意思表示條件語句的開始，並指定一個條件表達式，表達式包含兩個參數，以逗號分隔，表達式以圓括號括起。else表示條件表達式為假的情況。endif表示一個條件語句的結束，任何一個條件表達式都應該以endif結束。

當我們的變量\$(CC)值是“gcc”時，目標foo的規則是：

```
foo: $(objects)
    $(CC) -o foo $(objects) $(libs_for_gcc)
```

而當我們的變量\$(CC)值不是“gcc”時（比如“cc”），目標foo的規則是：

```
foo: $(objects)
    $(CC) -o foo $(objects) $(normal_libs)
```

當然，我們還可以把上面的那個例子寫得更簡潔一些：

```
libs_for_gcc = -lgnu
normal_libs =

ifeq ($(CC),gcc)
    libs=$(libs_for_gcc)
else
    libs=$(normal_libs)
endif

foo: $(objects)
    $(CC) -o foo $(objects) $(libs)
```

語法

條件表達式的語法為：

```
<conditional-directive>
<text-if-true>
endif
```

以及：

```
<conditional-directive>
<text-if-true>
else
<text-if-false>
endif
```

其中<conditional-directive>;表示條件關鍵字，如“ifeq”。這個關鍵字有四個。

第一個是我們前面所見過的“ifeq”

```
ifeq (<arg1>, <arg2>)
ifeq '<arg1>' '<arg2>'
ifeq "<arg1>" "<arg2>"
ifeq "<arg1>" '<arg2>'
ifeq '<arg1>' "<arg2>"
```

比較參數“arg1”和“arg2”的值是否相同。當然，參數中我們還可以使用make的函數。如：

```
ifeq ($(strip $(foo)),)
<text-if-empty>
endif
```

這個示例中使用了“strip”函數，如果這個函數的返回值是空（Empty），那麼<text-if-empty>;就生效。

第二個條件關鍵字是“ifneq”。語法是：

```
ifneq (<arg1>, <arg2>)
ifneq '<arg1>' '<arg2>'
ifneq "<arg1>" "<arg2>"
ifneq "<arg1>" '<arg2>'
ifneq '<arg1>' "<arg2>"
```

其比較參數“arg1”和“arg2”的值是否相同，如果不同，則為真。和“ifeq”類似。

第三個條件關鍵字是“ifdef”。語法是：

```
ifdef <variable-name>
```

如果變量<variable-name>的值非空，那到表達式為真。否則，表達式為假。當然，<variable-name>同樣可以是一個函數的返回值。注意，ifdef只是測試一個變量是否有值，其並不會把變量擴展到當前位置。還是來看兩個例子：

示例一：

```
bar =
foo = $(bar)
ifdef foo
frobozz = yes
else
frobozz = no
endif
```

示例二：

```
foo =
ifdef foo
frobozz = yes
else
frobozz = no
endif
```

第一個例子中，“\$(frobozz)”值是“yes”，第二個則是“no”。

第四個條件關鍵字是“ifndef”。其語法是：

```
ifndef <variable-name>
```

這個我就不多說了，和“ifdef”是相反的意思。

在<conditional-directive>這一行上，多餘的空格是被允許的，但是不能以[Tab]鍵做為開始（不然就被認為是命令）。而註釋符“#”同樣也是安全的。“else”和“endif”也一樣，只要不是以[Tab]鍵開始就行了。

特別注意的是，make是在讀取Makefile時就計算條件表達式的值，並根據條件表達式的值來選擇語句，所以，你最好不要把自動化變量（如“\$@”等）放入條件表達式中，因為自動化變量是在運行時才有的。

而且，為了避免混亂，make不允許把整個條件語句分成兩部分放在不同的文件中。

取自"<http://wiki.ubuntu.org.cn/index.php?title=%E8%B7%9F%E6%88%91%E4%B8%80%E8%B5%B7%E5%86%99Makefile:%E4%BD%BF%E7%94%A8%E6%9D%A1%E4%BB%B6%E5%88%A4%E6%96%AD&variant=zh-hant>"

本頁面已經被瀏覽3,913次。

- 此頁由Ubuntu中文的匿名用戶於2009年12月8日（星期二）19:39的最後更改。 在 Dbzhang800的工作基礎上。
 - 關於Ubuntu中文
 - 免責聲明