

---

Makefile 包含五种内容：显式规则、隐式规则、变量定义、指令和注释。规则、变量和指令将在后面的章节中详细描述。

---

- 明确的规则说明何时以及如何重新制作一个或多个文件，称为规则的目标。它列出了目标所依赖的其他文件，称为目标的先决条件，还可能提供用于创建或更新目标的配方。请参阅编写规则。
  - 一个隐含的规则说明了何时以及如何根据文件名重新制作一类文件。它描述了目标如何依赖于名称与目标相似的文件，并提供了创建或更新此类目标的方法。请参阅使用隐式规则。
  - 变量定义是为变量指定文本字符串值的行，该变量可以稍后替换到文本中。简单的 makefile 示例将对象的变量定义显示为所有对象文件的列表（请参阅使 Makefiles 更简单的变量）。
  - 指令是 make 在读取 makefile 时执行某些特殊操作的指令。这些包括：
    - 读取另一个 makefile（请参阅包括其他 Makefile）。
    - 决定（基于变量的值）是使用还是忽略 makefile 的一部分（参见 Makefile 的条件部分）。
    - 从包含多行的逐字字符串定义变量（请参阅定义多行变量）。
  - makefile 的一行中的 “#” 开始注释。它和该行的其余部分被忽略，除了没有被另一个反斜杠转义的尾部反斜杠将在多行中继续注释。仅包含注释的行（前面可能有空格）实际上是空白的，并且被忽略。如果您想要文字 #，请使用反斜杠（例如 #）对其进行转义。注释可能会出现在 makefile 的任何行上，尽管在某些情况下会特别处理它们。您不能在变量引用或函数调用中使用注释：任何 # 实例都将在变量引用或函数调用中按字面意思（而不是作为注释的开头）处理。
- 

配方中的注释被传递到 shell，就像任何其他配方文本一样。shell 决定如何解释它：这是否是注释取决于 shell。

---

在定义指令中，在变量定义期间不会忽略注释，而是在变量值中保持原样。扩展变量时，它们将被视为注释或配方文本，具体取决于评估变量的上下文。

---

- Splitting Lines Splitting long lines in makefiles