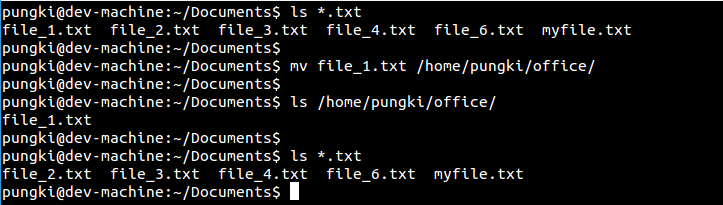
#### 1.移动文件

移动文件时需要注意的是文件的源地址和目标地址**必须不同**。这里有个例子，想要将**file\_1.txt**文件从当前目录移动到其它目录，以**/home/pungki/**为例，语法应该如下：

$ mv file\_1.txt /home/pungki/office



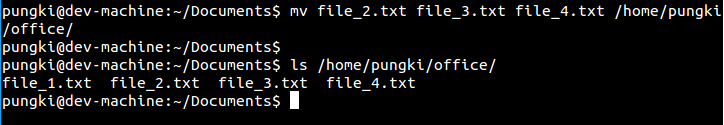
mv command

如我们所见，当我们移动 file\_1.txt 文件时，先前目录的 file\_1.txt 就**被删除了**。

#### 2.移动多个文件

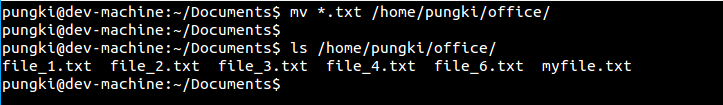
如果想一次移动多个文件，我们可以将他们放在一行并用空格分开。

**$ mv file\_2.txt file\_3.txt file\_4.txt /home/pungki/office**



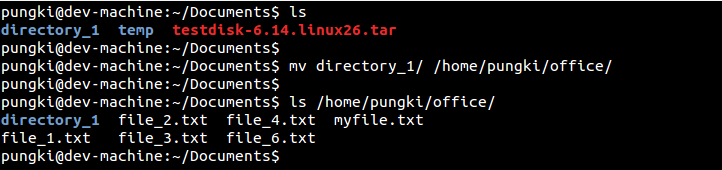
如果你的文件有规律可循的话那么你就可以使用通配符。比如，为了移除所有以**.txt**为扩展名的文件，我们可以用下面的命令：

**$ mv \*.txt /home/pungki/office**



#### 3.移动目录

不同于复制命令，用 mv 命令移动目录相当直接。移动目录你可以使用不带选项的 mv 命令。看下面的截图就一目了然了。



#### 4.重命名文件或目录

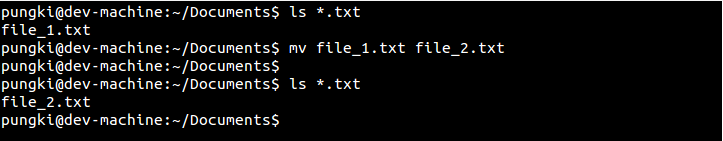
我们也用 mv 命令来重命名文件或目录。不过目标位置和源位置必须相同才可以。然后文件名必须不同。

假定我们当前所在目录为**/home/pungki/Documents**，而我们想将**file\_1.txt重命名为file\_2.txt**。那么命令应该如下：

**$ mv file\_1.txt file\_2.txt**

如果是绝对路径，它应该像下面这样：

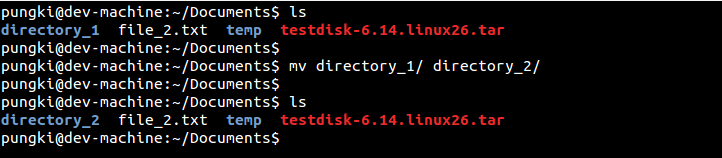
**$ mv /home/pungki/Documents/file\_1.txt /home/pungki/Documents/file\_2.txt**



#### 5. 重命名目录

上一段的规则同样适用于目录。请看这个例子：

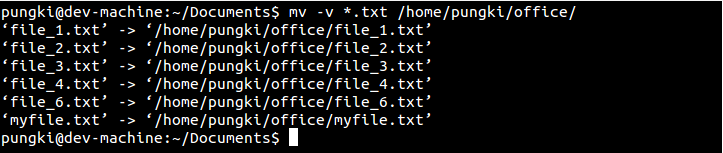
**$ mv directory\_1/ directory\_2/**



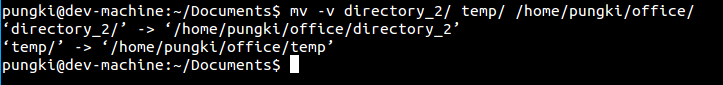
#### 6. 打印移动信息

当你移动或重命名一大堆文件或目录时，你可能会想在不去目标位置去查看的情况下知道你自己的命令是否成功地执行了。这就要用到**-v**选项了。

**$ mv -v \*.txt /home/pungki/office**



该方法同样适用于目录。

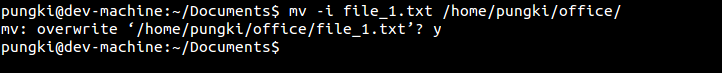


#### 7. 使用交互模式

当你将文件移动到其它位置，而那个位置恰好有同样的文件，这时 mv 命令会覆盖掉原来的文件。对于mv的这一行为一般不会有什么提示。如果想产生一个关于覆盖文件的提示，我们可以使用**-i**选项。（译注：通常发行版会通过alias命令，将-i作为默认选项，所以会有提示。）

假设我们想将 file\_1.txt 移动到 /home/pungki/office。同时，/home/pungki/office 目录下已经有file\_1.txt文件了。

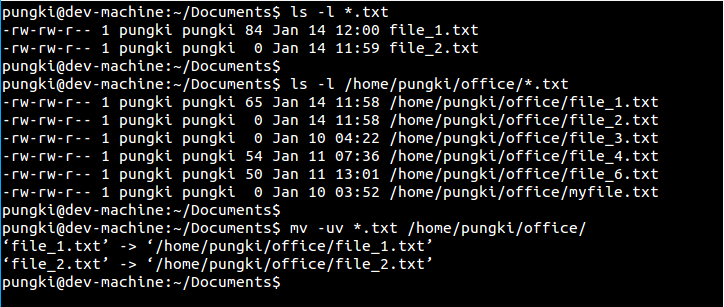
**$ mv -i file\_1.txt /home/pungki/office**



这个提示会让我们知道目标位置处file\_1.txt的存在。如果我们按**y**键，那么那个文件将会被删除，否则不会。

#### 8. 使用更新选项

-i 选项会提示我们关于覆盖文件的提示，而 -u 则只在源文件比目标文件新时才执行更新。让我们看一看下面的例子：



Update only newer

假如 file\_1.txt 和 file\_2.txt有如下特点：

1. File\_1.txt has 84 bytes file size and it last modified time is 12:00
2. File\_2.txt has 0 bytes file size and it last modified time is 11:59

我们想将它们移动到 /home/pungki/office 目录下。**但是目标地址**已经有file\_1.txt和file\_2.txt了。

我们用下面的命令将file\_1.txt 和file\_2.txt从当前目录移动到/home/pungki/office

1. $ mv -uv \*.txt /home/pungki/office

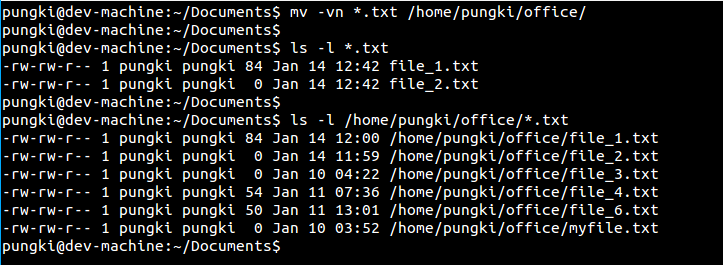
可以看到这些文件被移动了。能移动这些文件是因为它们最近的修改时间戳比 /home/pungki/office 目录中的文件新。

#### 9.不要覆盖任何已存在的文件

如果**-i**选项询问我们是否要覆盖文件，那么 -n 选项将不会允许我们覆盖任何已存在的文件。

继续使用第8点中的例子，如果我们将**-u 换成 -n**同时加上**-v**选项，那么我们会看到没有任何文件移动到了 /home/pungki/office 目录下。

1. $ mv -vn \*.txt /home/pungki/office

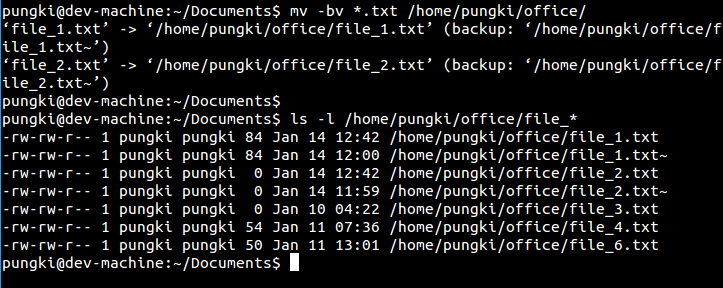


No overwrite

#### 10. 复制时创建备份

默认情况下，移动文件将会覆盖已存在的目标文件。但是如果我们移动错了文件而目标文件已经被新的文件覆盖了，这时应该怎么办才好呢？**有没有一种方法可以恢复之前的文件呢？答案是肯定的**。我们可以用**-b**选项。该选项会在新文件覆盖旧文件时将旧文件做备份。这里我们还以第8点为例。

1. $ mv -bv \*.txt /home/pungki/office



Backup option

如截图中所见，在 /home/pungki/office 目录下出现了名为**file\_1.txt~** 和 **file\_2.txt~** 的文件。那个**波浪符号（～）**意味着这些文件是备份文件。从它们的属性中我们可以看到，这些文件比file\_1.txt和file\_2.txt要旧。

#### 11. 无条件覆盖已经存在的文件

（译注：这一节是译者补充的，原文遗漏了这个重要选项）

当你希望无论如何都覆盖已经存在的文件或目录时，你可以使用 -f 选项。如果同时指定了 -f 选项和 -i 或 -n 选项，则 -f 选项会覆盖它们——即不进行任何提示而覆盖，所以，在使用此参数时，知道你在做什么。

1. $ mv -f \*.txt /home/pungki/office