### 4.4   Chmod Task：改变Linux/UNIX系统的文件权限

在UNIX和Linux系统下要改变文件的权限，可以使用chmod的shell命令。例 如：chmod 777 abc.txt。设置abc.txt文件的权限为“所有人有对文件操作的所有权限”。在Ant工具中可以通过Chmod任务方便地实现这些功能。在 Chmod任务中可以引用FileSet和DirSet类型指定的文件集合和目录集合。

### 4.4.1   Chmod Task属性及功能

Chmod任务包括如下属性：

（1）file：用于指定即将被改变权限的文件名称。

（2）dir：用于指定这个目录下的所有文件的权限将要被改变。在Chmod任务中dir属性和file属性两者必须选择其中一个。

（3）perm：用于指定文件的新权限。

（4）includes：用于指定一个或多个文件的匹配模式，只有符合这些模式的文件的权限才会被改变。可用逗号符或空格符进行分隔。

（5）excludes：与includes属性的作用正好相反，用于指定一个或多个文件匹配模式，只有不符合这些模式的文件的权限才会被改变。可用逗号符或空格符进行分隔。

（6）defaultexcludes：用于指定是否不包括Ant工具默认不包含的文件匹配模式。可取值为yes或no。忽略此属性时代表不包含默认的不包含的文件模式。

Ant工具默认不包含的文件模式有：\*\*#\*#，\*\*/.#\*，\*\* /%\*%，\*\*/.\_\*， \*\*/CVS，\*\*/CVS.cvsignore，\*\*/SCCS，\*\*/SCCSvssver.scc，\*\* /.svn，\*\*/.svn.DS\_Store。

（7）parallel：用于指定是否为每个包含的文件独立执行Chmod命令，默认为true。

（8）type：可取值为file、dir或both。取值file表示只改变文件的权限；若取值dir表示只改变目录的权限；若取值both则代表改变文件和目录两者的权限。

（9）maxparallel：用于指定一次执行Chmod命令的最大值。设为<=0代表不限制，默认为不限制。

（10）verbose：用于指定在执行命令后是否输出这个命令的执行信息。默认为false不输出。

### 4.4.2   通过Ant改变Linux/UNIX文件和目录权限的例子

在Linux和UNIX下通常通过Chmod命令来改变文件或目录的权限，Ant工具的Chmod任务也能实现同样的功能。例如，改变start.sh文件的权限让所有人对此文件具有读和执行的权限。

<chmod file="${dist}/start.sh" perm="ugo+rx"/>

所有用户对${dist}/start.sh文件都具有读和执行的权限。如果只有文件的所有者才具有读、写和执行权限，那么可以编写以下程序：

<chmod file="${dist}/start.sh" perm="o+rx"/>

当然也可以用数据来表示权限，上面的程序可改为：

<chmod file="${dist}/start.sh" perm="700"/> <!--   //与上面程序作用相同，

文件的所有者才具有读、写和执行的权限   -->

当然也可以通过FileSet和DirSet类型实现更复杂的功能，例如：

<chmod perm="go-rwx" type="file">

   <fileset dir="/web">

     <include name="\*\*\*.old"/>

   </fileset>

   <dirset dir="/web">

     <include name="\*\*/private\_\*"/>

   </dirset>

</chmod>

这个例子的作用是让不是文件的所有者可以访问cgi脚本。文件的权限将会改变的文件有：所有/web目录下的.cgi和.old文件，所有以private\_\*开头的文件或目录名以private\_\*开头的目录下的文件。