文件F2在ACL中有3个登录项：A、B和C。它们都可以读文件，而且B还可以写文件。除此之外，不允许其他的访问。文件F3很明显是个可执行文件，因为B和C都可以读并执行它。B也可以执行写操作。

这个例子展示了使用ACL进行保护的最基本形式。在实际中运用的形式要复杂得多。为了简便起见，我们目前只介绍了3种权限：读、写和执行。当然还有其他的权限。有些是一般的权限，可以运用于所有的对象，有些是对象特定的。一般的权限有destory object和copy object。这些可以运用于任何的对象，而不论对象的类型是什么。与对象有关的特定的权限会包括为邮箱对象的append message和对目录对象的sort alphabetically（按字母排序）等。

到目前为止，我们的ACL登录项是针对个人用户的。许多系统也支持用户组（group）的概念。组可以有自己的名字并包含在ACL中。语义学上组的变化也是可能的。在某些系统中，每个进程除了有用户ID（UID）外，还有组ID（GID）。在这类系统中，一个ACL登录项包括了下列格式的登录项：

UID1，GID1；rights1；UID2，GID2；rights2；...

在这样的条件下，当出现要求访问对象的请求时，必须使用调用者的UID和GID来进行检查。如果它们出现在ACL中，所列出的权限就是可行的。如果（UID，GID）的组合不在列表中，访问就被拒绝。

使用组的方法就引入了角色（role）的概念。如在某次系统安装后，Tana是系统管理员，在组里是sysadm。但是假设公司里也有很多为员工组织的俱乐部，而Tana是养鸽爱好者的一员。俱乐部成员属于pigfan组并可访问公司的计算机来管理鸽子的数据。那么ACL中的一部分会如图9-8所示。

文件	访问控制列表
Password	tana, sysadm: RW
Pigeon_data	bill, pigfan: RW; tana, pigfan: RW; ...

图9-8 两个访问控制列表

如果Tana想要访问这些文件，那么访问的成功与否将取决于她当前所登录的组。当她登录的时候，系统会让她选择想使用的组，或者提供不同的登录名和密码来区分不同的组。这一措施的目的在于阻止Tana在使用养鸽爱好者组的时候获得密码文件。只有当她登录为系统管理员时才可以这么做。

在有些情况下，用户可以访问特定的文件而与当前登录的组无关。这样的情况将引入通配符（wildcard）的概念，即“任何组”的意思。如，密码文件的登录项

tana，*：RW

会给Tana访问的权限而不管她的当前组是什么。

但是另一种可能是如果用户属于任何一个享有特定权限的组，访问就被允许。这种方法的优点是，属于多个组的用户不必在登录时指定组的名称。所有的组都被计算在内。同时它的缺点是几乎没有提供什么封装性：Tana可以在召开养鸽俱乐部会议时编辑密码文件。

组和通配符的使用使得系统有可能有选择地阻止用户访问某个文件。如，登录项

virgil，*：(none)；*，*：RW

给Virgil之外的登录项以读写文件的权限。上述方法是可行的，因为登录项是按顺序扫描的，只要第一个被采用，后续的登录项就不需要再检查。在第一个登录项中为Virgil找到了匹配，然后找到并应用这个存取权限，在本例中为（none）。整个查找在这时就中断了。实际上，再也不去检查剩下的访问权限了。

还有一种处理组用户的方法，无须使用包含（UID，GID）对的ACL登录项，而是让每个登录项成为UID或GID。如，一个进入文件pigeon_data的登录项是：

debbie：RW；phil：RW；pigfan：RW

表示debbie、phil以及其他所有pigfan组里的成员都可以读写该文件。

有时候也会发生这样的情况，即一个用户或组对特定文件有特定的许可权，但文件的所有者稍后又会收回。通过访问控制列表，收回过去赋予的访问权相对比较简单。这只要编辑ACL就可以修改了。但是如果ACL仅仅在打开某个文件时才会检查，那么改变它以后的结果就只有在将来调用open命令时才能奏效。对于已经打开的文件，就会仍然持有原来打开时拥有的权限，即使用户已经不再具有这样的权限。

9.3.3 权能

另一种切分图9-6矩阵的方法是按行存储。在使用这种方法的时候，与每个进程关联的是可访问的