**第5章 自主访问控制和强制访问控制比较'**

**1. 理解和解释自主访问控制和强制访问控制，举例说明其含义。**

自主访问(DAC)控制是依据主体的判断力授予访问权限，通常由客体的拥有者授权。应用于 UNIX, Windows系统。

强制访问控制 (MAC) 按照系统级策略限制主体对客体的访问。用户所创建的资源，也拒绝用户的完全控制。系统的安全策略完全取决于权限，权限由管理员设置。

**2. 比较ACL和能力表的差异**

从静态角度比较ACL和capability模型是不够的，不能说明其逻辑等价性。因为安全机制是动态变化的。

**自主访问控制：**

缺点：

1. 不适合用户多、用户经常变化的情况。不能授权用户在某时间段使用。
2. 运行时的安全检查不充分。比如查询某进程应用了哪些权限是困难的。
3. 不容易查询用户对哪些文件有权限，因为以文件为单位授权。

（4）不能解决混淆责任问题

**能力表：**

优点：（1）运行时安全检查更加有效，授权更方便。

（2）用户运行程序时，权责清晰。能解决混淆责任问题。

缺点：（1）改变文件状态比较困难。

二者差别：

1. 命名空间：ACL需要客体和主体的命名空间。能力表指定资源和权限。
2. 能力表授权给主体相应的权限，是基于主体聚集的权限管理方式。ACL授权给资源访问权限，由文件拥有者指定权限。
3. 权利边界：能力表的权限清晰，没有额外权限。ACL执行程序时权限边界不清晰，额外权限多，会引发混淆责任问题。
4. 鉴别内容：ACL鉴别主体。能力表不需要鉴别主体，但需要控制权限的传播。
5. 权限审查：ACL提供单客体的权限审查。能力表提供单主体的权限审查。
6. 权限撤销：ACL基于单客体撤销权限。能力表基于单主体撤销权限。
7. 最小特权：能力表提供细粒度的最小特权控制，可动态、短时间访问。ACL提供粗粒度的权限管理，不能实现短时间访问。
8. 适用性：能力表适合于进程及共享。ACL适合用户级共享。

**3. ACL和CAPBILITY中如何撤销权限，两种方式中撤销权限的不同之处**

ACL以资源为单位进行授权。资源属于客体，文件、端口等。撤销权限时基于客体进行权限收回。当收回用户的权限时比较困难，需要遍历文件系统才能把该用对对应的文件和目录权限收回，还不能收回动态使用过程的权限。所以收回用户的权限比较困难，采用该密码，使用后不能登录的放式收回权限。

CAPBILITY：

能力表基于单主体撤销权限，授权时基于主体进行授权。可检查主体的权限，然后收回主体的权限。但要控制主体权限的分发。比如软件制造商，卖给用户软件的使用权，用户也可以把软件给他人使用，这时能力表对主体控制其权限分发就非常重要。能力表要能够检查权限是否被用户分发出去，且能控制被分发出去的权限。

**4. ACL授权时为什么引发混淆责任问题，如何解决该问题**

**ACL:**

以编译程序 SYSX/FORT为例，SYSX文件夹下的文件： STAT、BILL文件。编译程序需要在SYSX目录下写文件, 因此对目录SYSX授予写权限。一个普通用户可以运行编译程序SYSX/FORT，用户也可以自己指定输出文件。恶意用户：输出文件名指定为SYSX/BILL,导致编译程序的清单文件被替换。编译器运行时，执行两个用户的权限：编译用户和执行用户的权限，系统无法区分应为哪个用户服务。

**能力表：**

各用户的权限清晰，各用户写自己文件夹下的目录不会冲突。

编译程序compiler的权能：访问SYSX/STAT和SYSX/BILL, 权限存放在自己的能力槽(slots 1 & 2)。应用者运行compiler,有写文件权限, 其权限存放在能力槽(slot 3). 应用者没有写SYSX/BILL文件的权限，因为授权时没有赋予相应的权限。当写入billing信息时, 编译程序使用slot 2中的权限. 当写输出信息时, 使用slot 3中的权限。