1973)。这一模型最初为管理军方安全系统而设计，现在被广泛运用于其他机构。在军方领域，文档（对象）有一定的安全等级，如内部级、秘密级、机密级和绝密级。每个人根据他可阅读文档的不同也被指定为不同的密级。如将军可能有权阅取所有的文档，而中尉可能只被限制在秘密级或更低的文档。代表用户运行的进程具有该用户的安全密级。由于该系统拥有多个安全等级，所以被称为多级安全系统。

Bell-La Padula模型对信息流做出了一些规定：

1) 简易安全规则：在密级k上面运行的进程只能读同一密级或更低密级的对象。例如，将军可以阅取中尉的文档，但中尉却不可以阅取将军的文档。

2) *规则：在密级k上面运行的进程只能写同一密级或更高密级的对象。例如，中尉只能在将军的信箱添加信息告知自己所知的全部，但是将军不能在中尉的信箱里添加信息告知自己所知的全部，因为将军拥有绝密的文档，这些文档不能泄露给中尉。

简而言之，进程既可下读也可上写，但不能颠倒。如果系统严格地执行上述两条规则，那么就不会有信息从高一级的安全层泄露到低一级的安全层。之所以用 * 代表这种规则是因为在最初的论文里，作者没有想出更好的名字所以只能用 * 作为临时的替代。但是最终作者没有想出更好的名字，所以在打印论文时用了 *。在这一模型中，进程可以读写对象，但不能直接相互通信。Bell-La Padula模型的图解如图9-13所示。

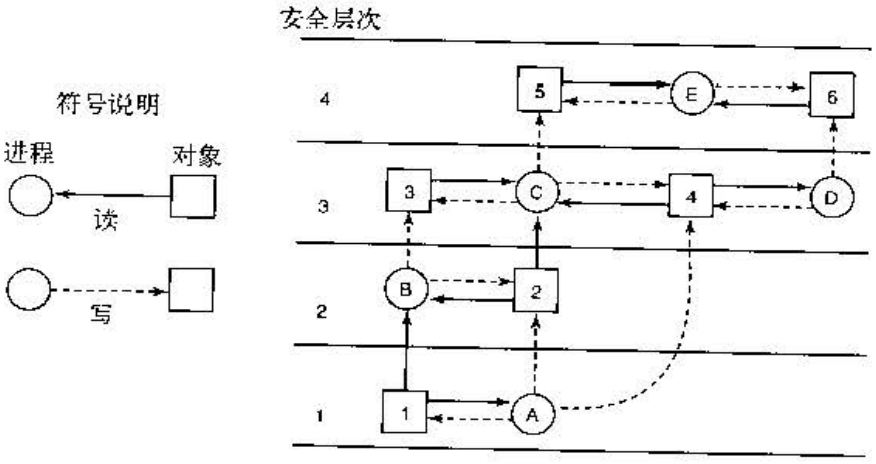

图9-13 Bell-La Padula多层安全模型

在图中，从对象到进程的（实线）箭头代该进程正在读取对象，也就是说，信息从对象流向进程。同样，从进程到对象的（虚线）箭头代表进程正在写对象，也就是说，信息从进程流向对象。这样所有的信息流都沿着箭头方向流动。例如，进程B可以从对象1读取信息但却不可以从对象3读取。

简单安全模型显示，所有的实线（读）箭头横向运动或向上；* 规则显示所有的虚线箭头（写）也横向运行或向上。既然信息流要么水平，要么垂直，那么任何从k层开始的信息都不可能出现在更低的级别。也就是说，没有路径可以让信息往下运行，这样就保证了模型的安全性。

Bell-La Padula模型涉及组织结构，但最终还是需要操作系统来强制执行。实现上述模型的一种方式是为每个用户分配一个安全级别，该安全级别与用户的认证信息（如UID和GID）一起存储。在用户登陆的时候，shell获取用户的安全级别，且该安全级别会被shell创建的所有子进程继承下去。如果一个运行在安全级别k之下的进程试图访问一个安全级别比k高的文件或对象，操作系统将会拒绝这个请求。相似地，任何试图对安全级别低于k的对象执行写操作的请求也一定会失败。

2. Biba模型

为了总结用军方术语表示的Bell-La Padula模型，一个中尉可以让一个士兵把自己所知道的所有信息复制到将军的文件里而不妨碍安全。现在让我们把同样的模型放在民用领域。设想一家公司的看门人拥有等级为1的安全性，程序员拥有等级为3的安全性，总裁拥有等级为5的安全性。使用Bell-La Padula模型，程序员可以向看门人询问公司的发展规划，然后覆写总裁的有关企业策略的文件。但并不是所有的公司都热衷于这样的模型。

Bell-La Padula模型的问题在于它可以用来保守机密，但不能保证数据的完整性。要保证数据的完