

都可用,因而事实上,整个目录树通常作为一个单元导出。服务器导出的目录列表用一个文件来维护,通常是/etc/exports。因此服务器启动后这些目录可以被自动地导出。客户端通过挂载这些导出的目录来访问它们。当一个客户端挂载了一个(远程)目录,该目录就成为客户端目录层次的一部分,如图10-35所示。

在这个例子中,客户端1将服务器1的bin目录挂载到客户端1自己的bin目录。因此它现在可以用/bin/sh引用shell并获得服务器的shell。无磁盘工作站通常只有一个框架文件系统(在RAM中),它从远程服务器中得到所有的文件,就像上例中一样。类似地,客户端1将服务器2中的/projects目录挂载到自己的/usr/ast/work目录,因此它用usr/ast/work/proj1/a就可以访问文件a。最后,客户端2也挂载了projects目录,它可以用/mnt/proj1/a访问文件a。从这里可以看到,由于不同的客户端将文件挂载到各自目录树中不同的位置,同一个文件在不同的客户端有不同的名字。对客户端来说挂载点是完全局部的,服务器不会知道文件在任何一个客户端中的挂载点。

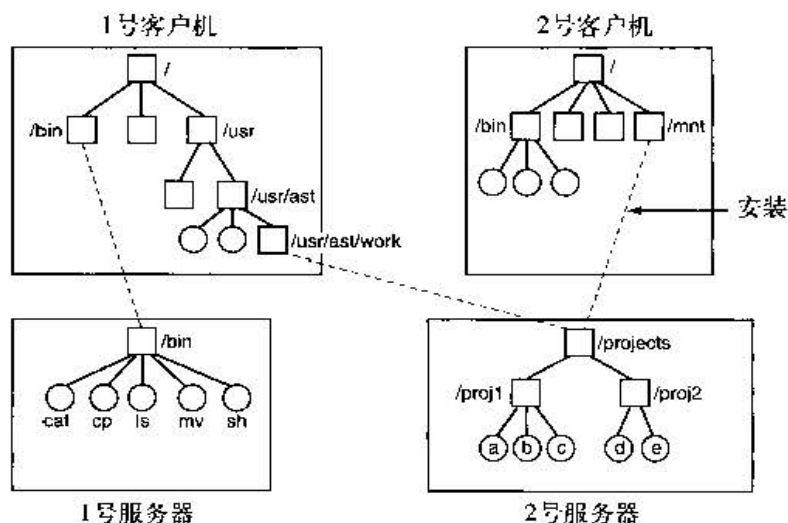


图10-35 远程挂载的文件系统的例子。图中的方框表示目录,圆形表示文件

2. NFS协议

由于NFS的目标之一是支持异构系统,客户端和服务端可能在不同硬件上运行不同操作系统,因此对客户端和服务端之间的接口给予明确定义是很关键的。只有这样,才有可能让任何一个新的客户端能够跟现有的服务器一起正确工作,反之亦然。

NFS通过定义两个客户端-服务器协议来实现这一目标。一个协议就是从客户端发送到服务器的一组请求以及从服务器返回给客户端的响应的集合。

第一个NFS协议处理挂载。客户端可以向服务器发送路径名,请求服务器许可将该目录挂载到自己的目录层次的某个地方。由于服务器并不关心目录将被挂载到何处,因此请求消息中并不包含挂载地址。如果路径名是合法的并且该目录已被导出,那么服务器向客户端返回一个文件句柄。这个文件句柄中的域唯一地标识了文件系统类型、磁盘、目录的i节点号以及安全信息等。随后对已挂载目录及其子目录中文件的读写都使用该文件句柄。

Linux启动时会在进入多用户之前运行shell脚本/etc/rc。可以将挂载远程文件系统的命令写入该脚本中,这样就可以在允许用户登录之前自动挂载必要的远程文件系统。此外,大部分Linux版本也支持自动挂载。这个特性允许一组远程目录跟一个本地目录相关联。当客户端启动时,并不挂载这些远程目录(甚至不与它们所在的服务器进行联络)。相反,在第一次打开远程文件时,操作系统向每个服务器发送一条信息。第一个响应的服务器胜出,其目录被挂载。

相对于通过/etc/rc文件进行静态挂载,自动挂载具有两个主要优势。第一,如果/etc/rc中列出的某个NFS服务器出了故障,那么客户端将无法启动,或者至少会带来一些困难、延迟以及很多出错信息。如果用户当前根本就不需要这个服务器,那么刚才的工作就白费了。第二,允许客户端并行地尝试一组服务器,可以实现一定程度的容错性(因为只要其中一个是在运行的就可以了),而且性能也可以得到提高(通过选择第一个响应的服务器——推测该服务器负载最低)。