

统调用的方式，用户程序获得操作系统的服务。例如，一个需要调用read的进程发送一个消息给某个文件服务器，告知它需要读什么内容。

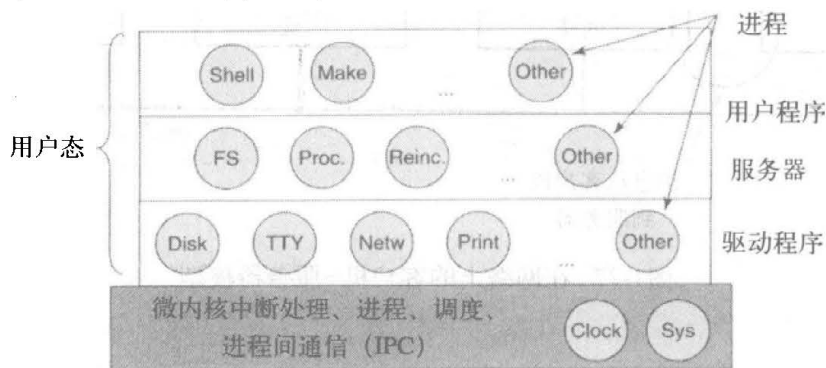


图1-26 MINIX 3 系统的结构

有一个有趣的服务器，称为再生服务器（reincarnation server），其任务是检查其他服务器和驱动器的功能是否正确。一旦检查出一个错误，它自动取代之，无须任何用户的干预。这种方式使得系统具有自修复能力，并且获得了较高的可靠性。

系统对每个进程的权限有着许多限制。正如已经提及的，设备驱动器只能与授权的I/O端口接触，对内核调用的访问也是按单个进程进行控制的，这是考虑到进程具有向其他多个进程发送消息的能力。进程也可获得有限的许可，让在内核的其他进程访问其地址空间。例如，一个文件系统可以为磁盘驱动器获得一种允许，让内核在该文件系统的地址空间内的特定地址上进行对盘块的一个新读操作。总体来说，所有这些限制是让每个驱动和服务只拥有完成其工作所需要的权限，别无其他，这样就极大地限制了故障部件可能造成的危害。

一个与小内核相关联的思想是在内核中的机制与策略分离的原则。为了更清晰地说明这一点，让我们考虑进程调度。一个比较简单的调度算法是，对每个进程赋予一个优先级，并让内核执行在具有最高优先级进程中可以运行的某个进程。这里，机制（在内核中）就是寻找最高优先级的进程并运行之。而策略（赋予进程以优先级）可以由用户态中的进程完成。在这个方式中，机制和策略是分离的，从而使系统内核变得更小。

1.7.4 客户机-服务器模式

一个微内核思想的略微变体是将进程划分为两类：服务器，每个服务器提供某种服务；客户端，使用这些服务。这个模式就是所谓的客户机-服务器模式。通常，在系统最底层是微内核，但并不是必须这样的。这个模式的本质是存在客户端进程和服务进程。

一般地，在客户端和服务之间的通信是消息传递。为了获得一个服务，客户端进程构造一段消息，说明所需要的服务，并将其发给合适的服务器。该服务完成工作，发送回应。如果客户端和服务运行在同一个机器上，则有可能进行某种优化，但是从概念上看，在这里讨论的是消息传递。

这个思想的一个显然的、普遍方式是，客户端和服务运行在不同的计算机上，它们通过局域或广域网连接，如图1-27所示。由于客户端通过发送消息与服务器通信，客户端并不需要知道这些消息是在它们的本地机器上处理，还是通过网络被送到远程机器上处理。对于客户端而言，这两种情形是一样的：都是发送请求并得到回应。所以，客户机-服务器模式是一种可以应用在单机或者网络机器上的抽象。

越来越多的系统，包括用户家里的PC机，都成为了客户端，而在某地运行的大型机器则成为服务器。事实上，许多Web就是以这个方式运行的。一台PC机向某个服务器请求一个Web页面，而后，该Web页面回送。这就是网络中客户机-服务器的典型应用方式。