

历史

大部分1980年前的 Unix 竞争者都被拴到单个硬件平台上，随着这个硬件的消亡而消亡。为什么 VMS 可以坚持这么久？值得我们作为案例研究一个原因是：VMS 成功地从最初的 VAX 硬件移植到了 Alpha 处理器（2003年正从 Alpha 移植到 Itanium 上）。MacOS 也在1980年代后期成功完成了从摩托罗拉68000到 PowerPC 芯片的迁跃。微软的 Windows 处在计算机商品化将通用计算机市场扁平化到单一 PC 文化的时期，真是生逢其时。

自1980年起，对于那些要么被 Unix 压倒要么已经先 Unix 而去的其它系统，不断重现的另一个特有弱点是：不具备良好的网络支持能力。

在一个网络无处不在的世界，即使为单个用户设计的系统也需要多用户能力（多种权限组）——因为如果不具备这一点，任何可能欺骗用户运行恶意代码的网络事务都将颠覆整个系统（Windows 宏病毒只是冰山一角）。如果不具备强大的多任务处理能力，操作系统同时处理网络传输和运行用户程序的能力将被削弱。操作系统还需要高效的 IPC，这样网络程序彼此能够通信，并且能够与用户的前台应用程序通信。

Windows 在这些领域具有严重缺陷却逃脱了惩罚，这仅仅因为它们连网变得真正重要以前就形成了垄断地位，并拥有一群已经对机器经常崩溃和无数安全漏洞习以为常的用户。微软的这种地位并不稳定，Linux 阵营正是利用这一点成功地（于2003年）在服务器操作系统市场取得了重大突破。

在个人机刚刚进入全盛时期的1980年左右，操作系统设计者认为 Unix 和其它传统的分时系统笨重、麻烦、不适合单用户个人机这个美丽新世界，而弃之不理——根本不顾 GUI 接口往往要求改造多任务处理能力，来适应不同窗口及其部件的绑定执行线程的事实。青睐客户端操作系统的趋势非常强烈，服务器操作系统就像已经逝去的蒸汽机时代的遗物一样遭到冷落。

但是，正如 BeOS 设计者们所注意到的那样，如果不实现某些近似通用分时系统的东西，就无法满足普遍联网的要求。单用户客户端操作系统在互联网世界里不可能繁荣。

这个问题促使客户端操作系统和服务器操作系统重新汇到了一起。首先，互联网时代之前的1980年代晚期，人们首次尝试通过局域网进行点对点联网，这种尝试暴露了客户端操作系统设计模式的不足：网络中的数据必须放到集合点上才能实现共享，因此如果没有服务器就做不到这一点。同时，人们对 Macintosh 和 Windows 客户端操作系统的体验也抬高了客户所能容忍的最低用户体验质量的门槛。

到了1990年，随着非 Unix 分时系统模型的实际消亡，客户端操作系统设计者还是拿不出来多少可能解决这一挑战的方案。他们可以吸收 Unix（如 MacOS X 所做的），或通过一次一个补丁重复发明一些大致等价的功能（如 Windows），或试图重新发明整个世界（如 BeOS，但失败了）。但与此同时，各种开源 Unix 的类客户端能力不断增强，开始能够使用 GUI 并能在廉价的个人机上运行。

然而，这些压力在两类操作系统上并未达到上面描述所意味的那种对称。将服务器操作系统特性，如多用户优先权组和完全多任务处理，改装到客户端操作系统上非常困难，很可能打破对旧版本客户端的兼容性，而且通常做出的系统既脆弱又令人不满意，不稳定也不安全。另一方面，将 GUI 应用于服务器操作系统，所出现的问题却大部分可通过灵活处理和投入更廉价硬件资源得到解决。就像造房子一样，在坚实的地基上修理上层建筑当然要比更换地基而不破坏上层建筑来得容易。