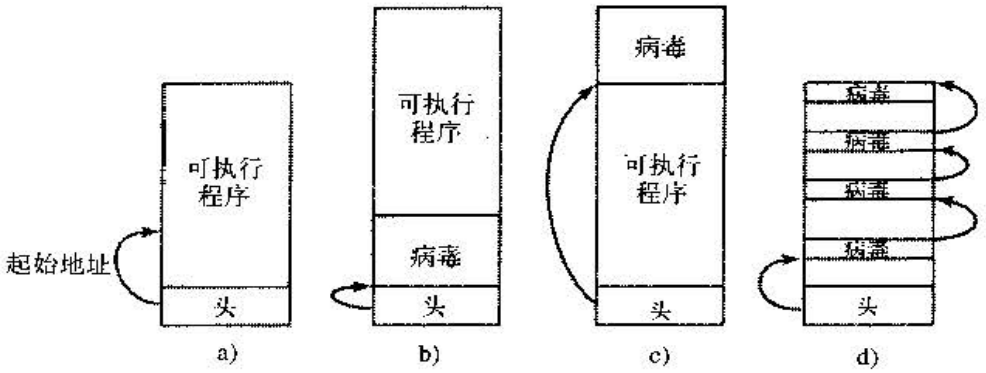

图9-28 a) 一段可执行程序；b) 病毒在前端；c) 病毒在后端；d) 病毒充斥在程序里的多余空间里

为了避免从前端装入病毒代码带来的复杂操作，大多数病毒是后端装入的，把它们自己附在可执行程序末端而不是前端，并且把文件头的起始地址指向病毒，如图9-28c所示。现在病毒要根据被感染程序的不同在不同的虚拟地址上运行，这意味着Virgil必须使用相对地址，而不是绝对地址来保证病毒是位置独立的。对资深的程序员来说，这样做并不难，并且一些编译器根据需要也可以完成这件事。

复杂的可执行程序格式，如Windows里的.exe文件和UNIX系统中几乎所有的二进制格式文件都拥有多个文本和数据段，可以用装载程序在内存中迅速把这些段组装和分配。在有些系统中（如Windows），所有的段都包含多个512字节单元。如果某个段不满，链接程序会用0填充。知道这一点的病毒会试图隐藏在这些空洞里。如果正好填满多余的空间，如图9-28d所示，整个文件大小将和未感染的文件一样保持不变，不过却有了一个附加物，所以隐含的病毒是幸运的病毒。这类病毒叫做*空腔病毒*（cavity virus）。当然如果装载程序不把多余部分装入内存，病毒也会另觅途径。

4. *内存驻留病毒*

到目前为止，我们假设当被感染的程序运行时，病毒也同时运行，然后将控制权交给真正的程序，最后退出。*内存驻留病毒*（memory-resident virus）与此相反，它们总是驻留在内存中（RAM），要么藏在内存上端，要么藏在下端的中断变量中。聪明的病毒甚至可以改变操作系统的RAM分布位图，让系统以为病毒所在的区域已经占用，从而避免了被其他程序覆盖。

典型的内存驻留病毒通过把陷阱或中断向量中的内容复制到任意变量中之后，将自身的地址放置其中，俘获陷阱或中断向量，从而将该陷阱或中断指向病毒。最好的选择是系统调用陷阱，这样病毒就可以在每一次系统调用时运行（在核心态下）。病毒运行完之后，通过跳转到所保存的陷阱地址重新激活真正的系统调用。

为什么病毒在每次系统调用时都要运行呢？这是因为病毒想感染程序。病毒可以等待直到发现一个exec系统调用，从而判断这是一个可执行二进制（而且也许是一个有价值的）代码文件，于是决定感染它。这一过程并不需要大量的磁盘活动，如图9-27所示，所以难以被发现。捕捉所有的系统调用也给了病毒潜在的能力，可以监视所有的数据并造成种种危害。

5. *引导扇区病毒*

正如我们在第5章所讨论的，当大多数计算机开机时，BIOS读引导磁盘的主引导记录放入RAM中并运行。引导程序判断出哪一个是活动分区，从该分区读取第一个扇区，即引导扇区，并运行。随后，系统要么装入操作系统要么通过装载程序导入操作系统。但是，多年以前Virgil的朋友发现可以制作一种病毒覆盖主引导记录或引导扇区，并能造成灾难性的后果。这种叫做*引导扇区病毒*（boot sector virus），它们现在已十分普遍了。

通常引导扇区病毒［包括MBR（主引导记录）病毒］，首先把真正的引导记录扇区复制到磁盘的安全区域，这样就能在完成操作后正常引导操作系统。Microsoft的磁盘格式化工具fdisk往往跳过第一个磁道，所以这是在Windows机器中隐藏引导记录的好地方。另一个办法是使用磁盘内任意空闲的扇区，然后更新坏扇区列表，把隐藏引导记录的扇区标记为坏扇区。实际上，由于病毒相当庞大，所以它也可以把自身剩余的部分伪装成坏扇区。如果根目录有足够大的固定空间，如在Windows 98中，根目录的末端