

在限定的秒数流逝之后，操作系统向该进程发送一个警告信号（alarm signal）。此信号引起该进程暂时挂起，无论该进程正在做什么，系统将其寄存器的值保存到堆栈，并开始运行一个特别的信号处理过程，比如重新发送可能丢失的消息。这些信号是软件模拟的硬件中断，除了定时器到期之外，该信号可以由各种原因产生。许多由硬件检测出来的陷阱，诸如执行了非法指令或使用了无效地址等，也被转换成该信号并交给这个进程。

系统管理器授权每个进程使用一个给定的UID标识（User Identification）。每个被启动的进程都有一个启动该进程的用户UID。子进程拥有与父进程一样的UID。用户可以是某个组的成员，每个组也有一个GID标识（Group Identification）。

在UNIX中，有一个UID称为超级用户（superuser），具有特殊的权利，可以违背一些保护规则。在大型系统中，只有系统管理员掌握着成为超级用户的密码，但是许多普通用户（特别是学生）们做出可观的努力试图找出系统的缺陷，从而使他们不用密码就可以成为超级用户。

在第2章中，我们将讨论进程、进程间通信以及有关的内容。

1.5.2 地址空间

每台计算机都有一些主存，用来保存正在执行的程序。在非常简单的操作系统中，内存中一次只能有一个程序。如果要运行第二个程序，第一个程序就必须被移出内存，再把第二个程序装入内存。

较复杂的操作系统允许在内存中同时运行多道程序。为了避免它们彼此互相干扰（包括操作系统），需要有某种保护机制。虽然这种机制必然是硬件形式的，但是它由操作系统掌控。

上述的观点涉及对计算机主存的管理和保护。另一种不同的但是同样重要并与存储器有关的内容，是管理进程的地址空间。通常，每个进程有一些可以使用的地址集合，典型值从0开始直到某个最大值。在最简单的情形下，一个进程可拥有的最大地址空间小于主存。在这种方式下，进程可以用满其地址空间，而且内存中也有足够的空间容纳该进程。

但是，在许多32位或64位地址的计算机中，分别有 2^{32} 或 2^{64} 字节的地址空间。如果一个进程有比计算机拥有的主存还大的地址空间，而且该进程希望使用全部的内存，那怎么办呢？在早期的计算机中，这个进程只好承认坏运气了。现在，有了一种称为虚拟内存的技术，正如前面已经介绍过的，操作系统可以把部分地址空间装入主存，部分留在磁盘上，并且在需要时穿梭交换它们。在本质上，操作系统创建了一个地址空间的抽象，作为进程可以引用地址的集合。该地址空间与机器的物理内存解耦，可能大于也可能小于该物理空间。对地址空间和物理空间的管理组成了操作系统功能的一个重要部分，本书中整个第3章都与这个主题有关。

1.5.3 文件

实际上，支持操作系统的另一个关键概念是文件系统。如前所述，操作系统的一项主要功能是隐藏磁盘和其他I/O设备的细节特性，并提供给程序员一个良好、清晰的独立于设备的抽象文件模型。显然，创建文件、删除文件、读文件和写文件等都需要系统调用。在文件可以读取之前，必须先在磁盘上定位和打开文件，在文件读过之后应该关闭该文件，有关的系统调用则用于完成这类操作。

为了提供保存文件的地方，大多数操作系统支持目录（directory）的概念，从而可把文件分类成组。比如，学生可给所选的每个课程创建一个目录（用于保存该课程所需的程序），另设一个目录存放电子邮件，再有一个目录用于保存万维网主页。这就需要系统调用创建和删除目录、将已有的文件放入目录中、从目录中删除文件等。目录项可以是文件或者目录，这样就产生了层次结构——文件系统，如图1-14所示。

进程和文件层次都可以组织成树状结构，但这两种树状结构有不少不同之处。一般进程的树状结构层次不深（很少超过三层），而文件树状结构的层次常常多达四层、五层或更多层。进程树层次结构是暂时的，通常最多存在几分钟，而目录层次则可能存在数年之久。进程和文件在所有权及保护方面也是有区别的。典型地，只有父进程能控制和访问子进程，而在文件和目录中通常存在一种机制，使文件所有者之外的其他用户也可以访问该文件。

目录层结构中的每一个文件都可以通过从目录的顶部，即根目录（root directory）开始的路径名（path name）来确定。绝对路径名包含了从根目录到该文件的所有目录清单，它们之间用正斜线隔开。如在图1-14中，文件CS101路径名是/Faculty/Prof.Brown/Courses/CS101。最开始的正斜线表示这是从根