

Linux为了解决这个问题,引入了可加载模块(loadable module)的概念。可加载模块是在系统运行时可以加载到内核的代码块。大部分情况下,这些模块是字符或者块设备驱动,但是它们也可以是完整的文件系统、网络协议、性能监控工具或者其他想要添加的模块。

当一个模块被加载到内核时,会发生下面几件事。第一,在加载过程中,模块会被动态地重新部署。第二,系统会检查这个驱动程序需要的资源是否可用(例如,中断请求级别)。如果有效,则把这些资源标记为正在使用。第三,设置所有需要的中断向量。第四,更新驱动转换表使其能够处理新的主设备类型。最后,运行驱动程序来完成可能需要的特定设备的初始化工作。一旦上述所有的步骤都完成了,这个驱动程序就安装完成了,也就和静态安装的驱动程序一样了。其他现代的UNIX系统也支持可加载模块。

10.6 Linux文件系统

在包括Linux在内的所有操作系统中,最可见的部分是文件系统。在本节的以下部分,我们将介绍隐藏在Linux文件系统、系统调用以及文件系统实现背后的基本思想。这些思想中有一些来源于MULTICS,虽然有很多已经被MS-DOS、Windows和其他操作系统使用过了,但是其他的都是UNIX类操作系统特有的。Linux的设计非常有意思,因为它忠实地秉承了“小的就是美好的”(Small is Beautiful)的设计原则。虽然只是使用了最简的机制和少量的系统调用,但是Linux却提供了强大的和优美的文件系统。

10.6.1 基本概念

最初的Linux文件系统是MINIX 1 文件系统。但由于它只能支持14字节的文件名(为了和UNIX Version 7兼容)和最大64MB的文件(这在只有10MB硬盘的年代是足够强大的),在Linux刚被开发出来的时候,开发者就意识到需要开发更好的文件系统(开始于MINIX 1 发布的5年后)。对MINIX 1 文件系统进行第一次改进后的文件系统是ext文件系统。ext文件系统能支持255个字符的文件名和2GB的文件大小,但是它的速度比MINIX 1 慢,所以仍然有必要对它进行改进。最终,ext2文件系统被开发出来,它能够支持长文件名和大文件,并且具有更好的性能,这使得它成为了Linux主要的文件系统。不过,Linux使用虚拟文件系统(VFS)层支持很多类型的文件系统(VFS将在下文介绍)。在Linux链接时,用户可以选择要构造到内核中的文件系统。如果需要其他文件系统,可以在运行时作为模块动态加载。

Linux中的文件是一个长度为0或多个字节的序列,可以包含任意的信息。ASCII文件、二进制文件和其他类型的文件是不加区别的。文件中各个位的含义完全由文件所有者确定,而文件系统不会关心。文件名长度限制在255个字符内,可以由除了NUL以外的所有ASCII字符构成,也就是说,一个包含了三个回车符的文件名也是合法的(但是这样命名很不方便)。

按照惯例,许多程序能识别的文件包含一个基本文件名和一个扩展名,中间用一个点连接(点也被认为是占用了文件名的一个字符)。例如一个名为prog.c的文件是一个典型的C源文件,prog.f90是一个典型的FORTRAN 90程序文件,而prog.o通常是一个object文件(编译器的输出文件)。这个惯例不是操作系统要求的,但是一些编译器和程序希望是这样,比如一个名为prog.java.gz的文件可能是一个gzip压缩的Java程序。

为了方便,文件可以被组织在一个目录里。目录存储成文件的形式并且在很大程度上可以作为文件处理。目录可以包含子目录,这样可以形成有层次的文件系统。根目录表示为“/”,它通常包含了多个子目录。字符“/”还用于分离目录名,所以/usr/ast/x实际上是说文件x位于目录ast中,而目录ast位于/usr目录中。表10-23列举了根目录下几个主要的目录及其内容。

在Linux中,不管是对shell还是一个打开文件的程序来说,都有两种方法表示一个文件的文件名。第一种方法是使用绝对路径,绝对路径告诉系统如何从根目录开始查找一个文件。例如/usr/ast/books/mos3/chap-10,这个路径名告诉系统在根目录里寻找一个叫usr的目录,然后再从usr中寻找ast目录……依照这种方式,最终找到chap-10文件。

目录	内容
bin	二进制(可执行)文件
dev	I/O设备文件
etc	各种系统文件
lib	库
usr	用户目录

图10-23 大部分Linux系统中一些重要的目录