

第21章 管道

一个“管道”(pipe)是一个FIFO(先进先出)字符列表，UNIX将管道作为一种特定的文件进行管理。

一组进程可将字符串“写”入管道，同时另一组进程则可从同一管道“读”信息。因此“管道”可以是，也主要用作作为进程间通信机构。

UNIX经常使用的一个概念是过滤程序(filter)，这是一种读输入文件，然后将之处理为输出文件的程序。使用“管道”可将两个或多个这种类型的程序连接起来，这样，UNIX向用户提供了一组非常综合和复杂的设施。

21.1 pipe(7723)

用户态程序通过系统调用“pipe”要求在核内执行“pipe”过程，以创建一个“管道”文件。

7728：在根设备上分配一“inode”。

7731：分配一“file”表项。

7736：由“r”记忆该“file”表项，分配第2个“file”表项。

7744：在R0和R1中返回用户文件标识。

7746：填充“file”数组项和“inode”项。

21.2 readp(7758)

“管道”与其他文件的一个重要区别是：一个“管道”文件有两个不同的位移量，一个用于“读”操作，另一个则用于“写”操作。而“写”位移量实际与文件长度相同。

7763：传送给“readp”的参数是一指向“file”数组项的指针，从该“file”项中可取得一“inode”指针。

7768：“plock”(7862)保证在同一时间只能有一个操作：“read”或“write”。

7776：由于“管道”已满(或者由于该文件的有效部分已达到文件限制)，一希望对“管道”进行写操作的进程已被阻塞，其标志是在“ip->i_mode”中设置了“IWRITE”。

7786：在进入睡眠之前释放锁。

7787：“i_count”是指向该“i_node”的“file”表项数。如果此值小于2，则“写进程”组中已无进程。

7789：正等待从管道输入信息的进程将设置“IREAD”标志。因为一个管道不可能同时为满和空，所以不可能同时设置“IWRITE”和“IREAD”标志。

7799：“prele”解锁该文件，并唤醒所有等待该管道的进程。

21.3 writep(7805)

此过程的结构在很多方面与“readp”相同，且互为补充。

7828：注意，若一“写进程”发现已无“读进程”，则接到一“信号”。（若一“读”进程发现已无“写”进程，则得到的返回结果为0，这是标准的文件尾端指示。）

7835：“管道”文件的长度不能超过“PIPSIZ”字符。只要“PIPSIE”（7715）不大于4096，该文件就不会变成“大”文件。从存取效率角度考虑，这是我们所希望的。（注意，PIPSIZ限制了“写”位移量的值。如果“读”位移量指针不是远超前于“写”位移量指针，那么该“管道”文件的真正内容会非常少。）

21.4 plock(7862)

在需要的等待之后，对该“inode”上锁。此过程由“readp”（7768）和“writep”（7815）调用。

21.5 prele(7882)

对该“inode”解锁，然后唤醒任一等待进程。此过程由“readp”、“writep”以及另外一些过程（特别是iput）调用。

