

深刻理解Linux系统进程间通信（IPC）PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022__E6_B7_B1_E5_88_BB_E7_90_86_E8_c103_144744.htm

Linux下的进程通信手段基本上是从Unix平台上的进程通信手段继承而来的。序Linux下的进程通信手段基本上是从Unix平台上的进程通信手段继承而来的。而对Unix发展做出重大贡献的两大主力AT&T的贝尔实验室及BSD（加州大学伯克利分校的伯克利软件发布中心）在进程间通信方面的侧重点有所不同。前者对Unix早期的进程间通信手段进行了系统的改进和扩充，形成了“system V IPC”，通信进程局限在单个计算机内；后者则跳过了该限制，形成了基于套接口（socket）的进程间通信机制。Linux则把两者继承了下来，如图示：其中，最初Unix IPC包括：管道、FIFO、信号；System V IPC包括：System V消息队列、System V信号灯、System V共享内存区；Posix IPC包括：Posix消息队列、Posix信号灯、Posix共享内存区。有两点需要简单说明一下：1）由于Unix版本的多样性，电子电气工程协会（IEEE）开发了一个独立的Unix标准，这个新的ANSI Unix标准被称为计算机环境的可移植性操作系统界面（PSOIX）。现有大部分Unix和流行版本都是遵循POSIX标准的，而Linux从一开始就遵循POSIX标准；2）BSD并不是没有涉足单机内的进程间通信（socket本身就可以用于单机内的进程间通信）。事实上，很多Unix版本的单机IPC留有BSD的痕迹，如4.4BSD支持的匿名内存映射、4.3BSD对可靠信号语义的实现等等。图一给出了Linux所支持的各种IPC手段，在本文接下来的讨论中，为了避免概念上的混

淆，在尽可能少提及Unix的各个版本的情况下，所有问题的讨论最终都会归结到Linux环境下的进程间通信上来。并且，对于Linux所支持通信手段的不同实现版本（如对于共享内存来说，有Posix共享内存区以及System V共享内存区两个实现版本），将主要介绍Posix API。Linux下进程间通信的几种主要手段简介：1.管道（Pipe）及有名管道（named pipe）：管道可用于具有亲缘关系进程间的通信，有名管道克服了管道没有名字的限制，因此，除具有管道所具有的功能外，它还允许无亲缘关系进程间的通信；2.信号（Signal）：信号是比较复杂的通信方式，用于通知接受进程有某种事件发生，除了用于进程间通信外，进程还可以发送信号给进程本身；Linux除了支持Unix早期信号语义函数signal外，还支持语义符合Posix.1标准的信号函数sigaction（实际上，该函数是基于BSD的，BSD为了实现可靠信号机制，又能够统一对外接口，用sigaction函数重新实现了signal函数）；100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com