

# 实验三 Linux 进程通信

## 一、实验目的

初步了解 Linux 系统中，创建进程和进程间通信的方法。

## 二、实验内容

1. 实验 Linux 下创建子进程及资源共享的方法
2. 编写一个程序，用 Linux 中的 IPC 机制，完成两个进程“石头、剪子、布”的游戏。

## 三、背景介绍

进程间通信（Interprocess Communication，IPC）实现了进程之间同步和交换数据的功能。本实验要求完成的是一个或几个用户态的进程，依靠内核提供的进程间通信的机制，完成几个用户进程之间的通信。通常，在 Linux 中允许以下几种进程间通信的机制：

- **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
- **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 `sigal` 外，还支持语义符合 Posix.1 标准的信号函数 `sigaction`
- **消息队列（报文队列）**：消息队列是消息的链接表，包括 Posix 消息队列 system V 消息队列。有足够权限的进程可以向队列中添加消息，被赋予读权限的进程则可以读走队列中的消息。消息队列克服了信号承载信息量少，管道只能承载无格式字节流以及缓冲区大小受限等缺点
- **共享内存**：使得多个进程可以访问同一块内存空间，是最快的可用 IPC 形式。是针对其他通信机制运行效率较低而设计的。往往与其它通信机制，如信号量结合使用，来达到进程间的同步及互斥
- **信号量（semaphore）**：主要作为进程间以及同一进程不同线程之间的同步手段
- **套接字（Socket）**：更为一般的进程间通信机制，可用于不同机器之间的进程间通信。起初是由 Unix 系统的 BSD 分支开发出来的，但现在一般可以移植到其它类 Unix 系统上：Linux 和 System V 的变种都支持套接字。

## 四、实验步骤

本实验可以创建三个进程，其中，一个进程为裁判进程，另外两个进程为选手进程。可以将“石头、剪子、布”这三招定义为三个整型值。胜负关系：石头 > 剪子 > 布 > 石头。

选手进程按照某种策略（例如，随机产生）出招，交给裁判进程判断大小。

裁判进程将对手的出招和胜负结果通知选手。比赛可以采取多盘（> 100 盘）定胜负，由裁判宣布最后结果。每次出招由裁判限定时间，超时判负。

每盘结果可以存放在文件或其他数据结构中。比赛结束，可以打印每盘的胜负情况和总的结果。

1. 设计表示“石头、剪子、布”的数据结构，以及它们之间的大小规则。
2. 设计比赛结果的存放方式。
3. 选择 IPC 的方法。
4. 根据你所选择的 IPC 方法，创建对应的 IPC 资源。
5. 完成选手进程。
6. 完成裁判进程。

——以下要求选作：

7. 决出班级的前三甲，与另外班级的前三甲比赛，决出年级冠军。
8. 如果有兴趣，再把这个实验改造成网络版。即在设计时就要考虑 IPC 层的封装。

## 五、实验结果

1. 实验数据结构：
2. 实验大小规则&存放方式：
3. 实验所选择的 IPC 方法和理由：
4. 如果选择消息队列机制，描述消息缓冲区结构：
5. 如何创建 IPC 资源？
6. 程序主要流程或关键算法：

## 六、实验总结