**操作系统课程实践（网安）**

# 实验3 Linux 进程管理

## 实验介绍

本实验通过编写模拟Shell，管道通信程序，消息通信程序，共享内存机制，了解Linux进程的创建，进程通信的方法。本实验为分组实验，小组协作完成，每人完成一个程序，但需要教会同组其他同学所有实验内容。

## 任务描述

在Linux系统下，实现一个模拟的 shell

在Linux系统下，实现一个管道通信程序

利用 Linux 的消息队列通信机制实现两个线程间的通信

利用 Linux 的共享内存通信机制实现两个进程间的通信

## 实验目的

通过对 Linux 进程控制的相关系统调用的编程应用，进一步加深对进程概念的理解，明确进程和程序的联系和区别，理解进程并发执行的具体含义。

通过 Linux 管道通信机制、消息队列通信机制、共享内存通信机制的应用，加深 对不同类型的进程通信方式的理解。

通过对 Linux 的 Posix 信号量及 IPC 信号量的应用，加深对信号量同步机制的理解。

请根据自身情况，进一步阅读分析相关系统调用的内核源码实现。

## 实验内容

1. 掌握Linux 进程管理的基本概念

### 网上查找资料，了解Linux Shell基本概念，了解Shell 内部命令和外部命令的概念，（https://blog.csdn.net/coding\_dong/article/details/103576071）。

### 阅读教材3.4.5 Linux同步机制解析，了解Linux的同步机制和编程方法，网上查找资料（https://www.cnblogs.com/linengier/p/9399880.html），了解Linux信号量的编程方法。

### 阅读教材3.6.3 Linux进程通信机制，了解Linux的进程通信机制和编程方法，网上查找资料（https://www.cnblogs.com/52php/tag/%E8%BF%9B%E7%A8%8B%E9%80%9A%E4%BF%A1/），了解Linux进程通信的编程方法。

1. 实现模拟的 shell

### 编写三个不同的程序 cmd1.c，cmd2.c，cmd3.c，每个程序的功能自定（不要太简单，比如可以实现文件复制、显示文件内容、计算器等），分别编译成可执行文件 cmd1，cmd2，cmd3（文件名可以根据功能做相应修改）。然后再编写一个程序，模拟 shell 程序的功能（包括内部命令（2个及以上）和外部命令（3个及以上））：能根据用户输入的字符串（表示相应的命令名），如果是内部命令，则执行对应的函数功能（例如echo、pwd、times，exit等）；如果是外部命令，则为相应的命令创建子进程并让它去执行相应的程序，而父进程则等待子进程结束，然后再等待接收下一条命令。如果接收到的命令为 exit，则父进程结束，退出模拟 shell；如果接收到的命令是无效命令，则显示“Command not found”，继续等待输入下一条命令。

1. 实现管道通信程序

### 实现管道程序，完成如下任务：

### 编写一个管道程序，要求能试验阻塞型读写过程中的各种情况，测试管道的默认大小（接收端不读管道的情况下，管道中最多能写多少数据）。

### 编写一个管道程序，由父进程创建一个管道，然后再创建 3 个子进程，并由这三个子进程利用管道与父进程之间进行通信：子进程发送随机长度的信息 内容给父进程，父进程等三个子进程全部发完消息后再接收信息，并显示消息内容。通信的具体内容可根据自己的需要随意设计，要求利用 Posix 信号量机制实现进程间对管道的互斥访问。运行程序，观 察各种情况下，进程实际读写的字节数以及进程阻塞唤醒的情况。

1. 利用 Linux 的消息队列通信机制实现两个线程间的通信

### 编写程序创建三个线程：sender1 线程、sender2 线程和 receive 线程，三个线程的功能。描述如下：

### ①sender1 线程：运行函数 sender1()，它创建一个消息队列，然后等待用户通过终端输入一串字符，并将这串字符通过消息队列发送给 receiver 线程；可循环发送多个消息，直到用户输入“exit”为止，表示它不再发送消息，最后向 receiver 线程发送消息“end1”，并且等待 receiver 的应答，等到应答消息后，将接收到的应答信息显示在终端屏幕上，结束线程的运行。

### ②sender2 线程：运行函数 sender2()，共享 sender1 创建的消息队列，等待用户通过终端输入一串字符，并将这串字符通过消息队列发送给 receiver 线程；可循环发送多个消息， 直到用户输入“exit”为止，表示它不再发送消息，最后向 receiver 线程发送消息“end2”， 并且等待 receiver 的应答，等到应答消息后，将接收到的应答信息显示在终端屏幕上，结束 线程的运行。

### ③receiver 线程：运行函数 receive()，它通过消息队列接收来自 sender1 和 sender2 两 个线程的消息，将消息显示在终端屏幕上，当收到内容为“end1”的消息时，就向 sender1发送一个应答消息“over1”； 当收到内容为“end2”的消息时，就向 sender2 发送一个应答消息“over2”；消息接收完成后删除消息队列，结束线程的运行。选择合适的信号量机制实现三个线程之间的同步与互斥。

1. 利用 Linux 的共享内存通信机制实现两个进程间的通信

### 编写程序 sender，它创建一个共享内存，然后随机产生一个100以内的计算表达式（例如12+34），并将这串表达式字符串通过共享内存发送给 receiver；最后，receiver完成表达式运算后，将计算结果(36)写到共享内存 ，sender收到应答消息后，将接收到的计算结果显示在终端屏幕上。上述计算重复10次后，sender向receiver发送”end”，等待recever发送”over”信息后，删除共享内存，结束程序的运行。

### 编写程序receiver， 它通过共享内存接收来自 sender 产生的信息，如果该信息是计算表达式，则将表达式显示在终端屏幕上，然后计算表达式的结果，再通过该共享内存向sender 发送计算结果，等待接收下一个消息；如果该信息是”end”，则向 sender 发送一个应答消息”over”，并结束程序的运行。选择合适的信号量机制实现两个进程对共享内存的互斥及同步使用。