**Linux进程管理实验**

实验目的：

熟练掌握Linux操作系统的使用，掌握Linux的系统的进程管理相关内容，掌握进程之间的通信方式。

进程是操作系统中最重要的概念，贯穿始终，也是学习现代操作系统的关键。通过本次实验，要求理解进程的实质和进程管理的机制。在Linux系统下实现进程从创建到终止的全过程，从中体会进程的创建过程、父进程和子进程的关系、进程状态的变化、进程之间的同步机制、进程调度的原理和以信号和管道为代表的进程间通信方式的实现。

实验要求：

完成实验内容并写出实验报告，报告应具有以下内容：

1. 实验目的；
2. 实验内容；
3. 题目分析及基本设计过程分析；
4. 配置文件关键修改处的说明及运行情况，应有必要的效果截图；
5. 实验过程中出现的问题及解决方法；
6. 实验体会。

实验内容：

1. 在命令行新建多个普通用户，如tux，bob，Alice，lily等，给每个用户创建密码，并将这几个用户分到同一个组xjtuse中。再新建两个组coding和testing，使得某些用户也分别为其组用户。在root用户和新建用户之间切换，验证用户创建成功与否。（给出相关命令运行结果）
2. 实现*sudo*委托管理任务，给上述某一指定的普通用户赋予创建用户的权限。（给出相关配置文件和命令运行结果）
3. 备份数据是系统应该定期执行的任务，请利用cron计划作业在每周五下午6：10对某用户（如tux）主目录下的文件进行备份（可使用tar 命令）。给出相关运行结果和邮件记录。
4. **编制实现软中断通信的程序**

使用系统调用fork()创建两个子进程，再用系统调用signal()让父进程捕捉键盘上发出的中断信号（即按delete键），当父进程接收到这两个软中断的某一个后，父进程用系统调用kill()向两个子进程分别发出整数值为16和17软中断信号，子进程获得对应软中断信号，然后分别输出下列信息后终止：

**Child process 1 is killed by parent !!**

**Child process 2 is killed by parent !!**

父进程调用wait()函数等待两个子进程终止后，输入以下信息，结束进程执行：

**Parent process is killed!!**

多运行几次编写的程序，简略分析出现不同结果的原因。

1. **编制实现进程的管道通信的程序**

使用系统调用pipe()建立一条管道线，两个子进程分别向管道写一句话：

**Child process 1 is sending a message!**

**Child process 2 is sending a message!**

而父进程则从管道中读出来自于两个子进程的信息，显示在屏幕上。

要求：父进程先接收子进程P1发来的消息，然后再接收子进程P2发来的消息。