# 



**课 程 实 验 报 告**

**课程名称： 操作系统原理**

**专业班级： CS1503班**

**学 号： U201514559**

**姓 名： 周铭昊**

**指导教师： 郑然**

**报告日期： 2018年1月19日**

**计算机科学与技术学院**

**目录**

[**实验一 进程控制 1**](#_Toc502051830)

[**1.1实验目的 1**](#_Toc502051831)

[**1.2 实验内容 1**](#_Toc502051832)

[**1.3 实验过程 1**](#_Toc502051833)

[**1.3.1 编程思路 1**](#_Toc502051834)

[**1.3.2 遇到的问题及解决方式 1**](#_Toc502051835)

[**1.3.3 实验测试与结果分析 2**](#_Toc502051836)

[**1.4 实验总结 2**](#_Toc502051837)

[**实验二 线程同步与通信 3**](#_Toc502051838)

[**2.1 实验目的 3**](#_Toc502051839)

[**2.2 实验内容 3**](#_Toc502051840)

[**2.3 实验过程 3**](#_Toc502051841)

[**2.3.1 编程思路 3**](#_Toc502051842)

[**2.3.2 遇到的问题及解决方式 3**](#_Toc502051843)

[**2.3.3 实验测试与结果分析 3**](#_Toc502051844)

[**2.4 实验总结 3**](#_Toc502051845)

[**实验三 共享内存与进程同步 4**](#_Toc502051846)

[**3.1 实验目的 4**](#_Toc502051847)

[**3.2 实验内容 4**](#_Toc502051848)

[**3.3 实验过程 4**](#_Toc502051849)

[**3.3.1 编程思路 4**](#_Toc502051850)

[**3.3.2 遇到的问题及解决方式 4**](#_Toc502051851)

[**3.3.3 实验测试与结果分析 4**](#_Toc502051852)

[**3.4 实验总结 4**](#_Toc502051853)

[**实验四 Linux文件目录 5**](#_Toc502051854)

[**4.1 实验目的 5**](#_Toc502051855)

[**4.2 实验内容 5**](#_Toc502051856)

[**4.3 实验过程 5**](#_Toc502051857)

[**4.3.1 编程思路 5**](#_Toc502051858)

[**4.3.2 遇到的问题及解决方式 5**](#_Toc502051859)

[**4.3.3 实验测试与结果分析 5**](#_Toc502051860)

[**4.4 实验总结 5**](#_Toc502051861)

# 实验一 进程控制

## **1.1实验目的**

1、加深对进程的理解,进一步认识并发执行的实质；

2、分析进程争用资源现象,学习解决进程互斥的方法；

3、掌握Linux进程基本控制；

4、掌握Linux系统中的软中断和管道通信。

## **1.2 实验内容**

编写程序，演示多进程并发执行和进程软中断、管道通信。

父进程使用系统调用pipe( )建立一个管道,然后使用系统调用fork()创建两个子进程，子进程1和子进程2；

子进程1每隔1秒通过管道向子进程2发送数据:

I send you x times. (x初值为1，每次发送后做加一操作）

子进程2从管道读出信息，并显示在屏幕上。

父进程用系统调用signal()捕捉来自键盘的中断信号（即按Ctrl+C键）；当捕捉到中断信号后，父进程用系统调用Kill()向两个子进程发出信号，子进程捕捉到信号后分别输出下列信息后终止：

Child Process l is Killed by Parent!

Child Process 2 is Killed by Parent!

父进程等待两个子进程终止后，释放管道并输出如下的信息后终止

Parent Process is Killed!

## **1.3 实验过程**

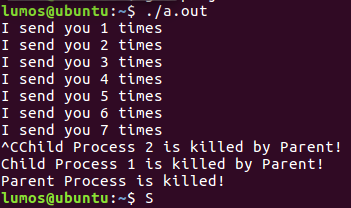
### 1.3.1 编程思路

要使用管道进行进程间的通信，首先用pipe()函数创建无名管道，pipefd[0]可以看成一个指针，表示管道的读入端，而pipefd [1]则表示管道的写入端。在写进程的时候要通过close()函数关闭读端口，读进程中要关闭写关口，防止出现读写同时进行导致输出错误信息。

然后设置软中断信号，SIGINT表示Ctrl+C的组合键信号，当信号发生后，跳转到两个信号处理函数Int1()、Int2()中，

### 1.3.2 遇到的问题及解决方式

### 1.3.3 实验测试与结果分析



## **1.4 实验总结**

# 实验二 线程同步与通信

## **2.1 实验目的**

1、掌握Linux下线程的概念；

2、了解Linux线程同步与通信的主要机制；

3、通过信号灯操作实现线程间的同步与互斥。

## **2.2 实验内容**

通过Linux多线程与信号灯机制，设计并实现计算机线程与I/O线程共享缓冲区的同步与通信。

程序要求:两个线程,共享公共变量a

线程1负责计算(1到100的累加，每次加一个数)

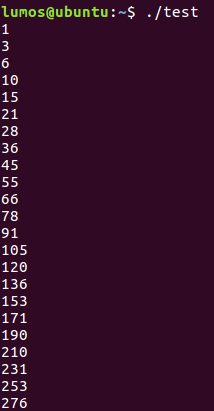
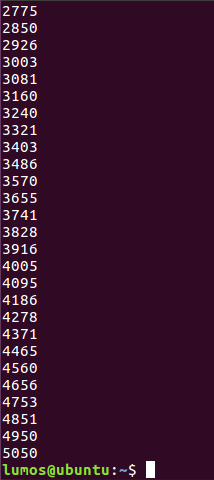
线程2负责打印（输出累加的中间结果）

## **2.3 实验过程**

### 2.3.1 编程思路

### 2.3.2 遇到的问题及解决方式

### 2.3.3 实验测试与结果分析

## **2.4 实验总结**

# 实验三 共享内存与进程同步

## **3.1 实验目的**

1、掌握Linux下共享内存的概念与使用方法；

2、掌握环形缓冲的结构与使用方法；

3、掌握Linux下进程同步与通信的主要机制。

## **3.2 实验内容**

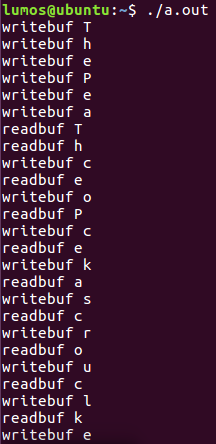
利用多个共享内存（有限空间）构成的环形缓冲，将源文件复制到目标文件，实现两个进程的誊抄。

## **3.3 实验过程**

### 3.3.1 编程思路

### 3.3.2 遇到的问题及解决方式

### 3.3.3 实验测试与结果分析





## **3.4 实验总结**

# 实验四 Linux文件目录

## **4.1 实验目的**

1、了解Linux文件系统与目录操作；

2、了解Linux文件系统目录结构；

3、掌握文件和目录的程序设计方法。

## **4.2 实验内容**

1. 功能类似ls -lR；

2. 查询指定目录下的文件及子目录信息；

3. 显示文件的类型、大小、时间等信息；

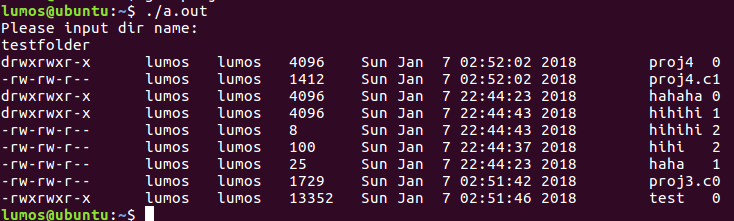
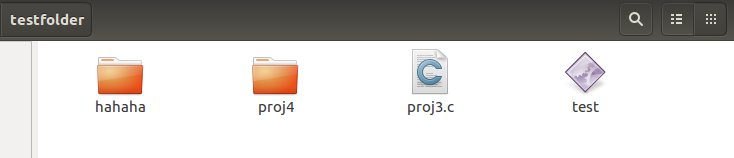
4. 递归显示子目录中的所有文件信息。

## **4.3 实验过程**

### 4.3.1 编程思路

### 4.3.2 遇到的问题及解决方式

### 4.3.3 实验测试与结果分析

## **4.4 实验总结**