

**课程实验报告**

**课程名称： 嵌入式操作系统**

**专业班级： 物联网工程1601**

**学 号： Ｕ201614898**

**姓 名： 潘翔**

**指导教师： 石柯**

**报告日期： 2018.4.24**

**计算机科学与技术学院**

[实验1 进程控制 4](#_Toc929925930)

[1.1 实验目的与要求 4](#_Toc1703594094)

[1.2 实验内容 4](#_Toc207680048)

[1.3 实验过程与结果 4](#_Toc611280519)

[1.4实验结果分析 4](#_Toc1114217023)

[1.5心得体会与总结 5](#_Toc2022153414)

[实验2 线程同步与通信 7](#_Toc1882785084)

[2.1 实验目的与要求 7](#_Toc1782888898)

[2.2 实验内容 7](#_Toc682624282)

[2.3 实验过程与结果 7](#_Toc1134361536)

[2.4 实验结果分析 7](#_Toc514058441)

[2.5 心得体会与总结 7](#_Toc1973904164)

[实验3 共享内存与进程同步 9](#_Toc1552643782)

[3.1 实验目的与要求 9](#_Toc405689082)

[3.2 实验内容 9](#_Toc1305080438)

[3.3 实验过程与结果 9](#_Toc923752061)

[3.4 实验结果分析 9](#_Toc589151370)

[3.5 心得体会与总结 9](#_Toc1783831161)

[实验4 TinyOS实验 9](#_Toc1771726525)

[4.1 实验目的与要求 10](#_Toc1181859447)

[4.1.1 基础实验(Task1-3) 10](#_Toc184665872)

[4.1.2 附加实验(Task4) 10](#_Toc524777622)

[4.2 实验内容 10](#_Toc247386889)

[4.2.1 BlinkBinary(Task1) 10](#_Toc1572472764)

[4.2.2 BlinkCompute(Task2) 10](#_Toc1953216182)

[4.2.3 BlinkComputeSlicing(Task3) 10](#_Toc1110910701)

[4.2.4 TinyOSSensor(Task4) 10](#_Toc1985251195)

[4.3 实验过程与结果 10](#_Toc1052043243)

[4.4 实验结果分析 11](#_Toc2120544162)

[3.5 心得体会与总结 11](#_Toc385750536)

[附录 11](#_Toc847494704)

[Lab1 code 12](#_Toc902986444)

[Lab2 code 12](#_Toc2089344630)

[Lab3 code 12](#_Toc1055174752)

[Lab4 code 12](#_Toc1514266963)

# 实验1 进程控制

## 1.1 实验目的与要求

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

1. 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；

1. 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。

1. 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!

1. 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 实验过程与结果



图1-1 编译C文件

图1-2 执行C文件

## 1.4实验结果分析

图1-3 函数流程图

## 1.5心得体会与总结

重点部分。

# 实验2 线程同步与通信

## 2.1 实验目的与要求

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:

1. 两个线程,共享公共变量a
2. 线程1负责计算(1到100的累加，每次加一个数)
3. 线程2负责打印（输出累加的中间结果)

## 2.3 实验过程与结果



图1-1 编译C文件

图1-2 执行C文件

## 2.4 实验结果分析

图1-3 函数流程图

## 2.5 心得体会与总结

重点部分。

# 实验3 共享内存与进程同步

## 3.1 实验目的与要求

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 实验过程与结果



图1-1 编译C文件

图1-2 执行C文件

## 3.4 实验结果分析

图1-3 函数流程图

## 3.5 心得体会与总结

重点部分。

# 实验4 TinyOS实验

## 4.1 实验目的与要求

### 4.1.1 基础实验(Task1-3)

1. 了解典型nesC的程序结构及语法
2. 了解tinyos执行机制，实现程序异步处理的方法
3. 了解tinyos中task抽象及其使用
4. 在Blink程序中使用printf输出信息，使用task实现计算和外部设备操作的并发

### 4.1.2 附加实验(Task4)

1. 了解Telosb节点中传感器的类型与使用；
2. 了解Telosb节点的传感器数据的获取；
3. 获取的数据通过printf传输至电脑；
4. 将节点的传感器数据传输到基站，并在电脑端解析显示，了解数据的采集过程。

## 4.2 实验内容

### 4.2.1 BlinkBinary(Task1)

1. Blink程序的编译和下载
2. 给Blink程序加入printf，在每次定时器事件触发点亮LED的同时通过串口显示信息
3. 修改BLink程序，只使用一个Timer，三个LED灯作为3位的二进制数表示（亮灯为1，不亮为0），按照0-7的顺序循环显示，同时将数值显示在终端上。

### 4.2.2 BlinkCompute(Task2)

修改Blink程序，在timer0的触发事件处理中加入计算

### 4.2.3 BlinkComputeSlicing(Task3)

修改computetask的内容，将400001次计算分割成为若干小的部分，从而使得LED1和LED2的fire事件可以被正常调用，并通过printf输出。

### 4.2.4 TinyOSSensor(Task4)

## 4.3 实验过程与结果



图1-1 编译C文件

图1-2 执行C文件

## 4.4 实验结果分析

图1-3 函数流程图

## 3.5 心得体会与总结

重点部分。

# 附录

## Lab1 code

## Lab2 code

## Lab3 code

## Lab4 code